Is Quantum Randomness Algorithmic Random? A Preliminary Attack

Cristian S. Calude, Michael J. Dinneen

Department of Computer Science
University of Auckland, New Zealand
www.cs.auckland.ac.nz/~{cristian,mjd}

CAI'05, Thessaloniki, 2005

Outline

Outline

Outline

Outline

Outline

Main problem

Randomness is an essential ingredient for computation.

The pitfalls of software-generated pseudo-randomness are well-known. So, can we do it better?

The obvious candidate is quantum randomness.

Quantum randomness has been confirmed by theoretical and experimental research. The first book containing a million of quantum random digits—generated by using radioactive decay from electronic vacuum tubes—was published by the RAND Corporation in 1955.

Is it quantum randomness provable better then pseudo-randomness? If yes, how much better?

Main problem

Randomness is an essential ingredient for computation.

The pitfalls of software-generated pseudo-randomness are well-known. So, can we do it better?

The obvious candidate is quantum randomness.

Quantum randomness has been confirmed by theoretical and experimental research. The first book containing a million of quantum random digits—generated by using radioactive decay from electronic vacuum tubes—was published by the RAND Corporation in 1955.

Is it quantum randomness provable better then pseudo-randomness? If yes, how much better?

Main problem

Randomness is an essential ingredient for computation.

The pitfalls of software-generated pseudo-randomness are well-known. So, can we do it better?

The obvious candidate is quantum randomness.

Quantum randomness has been confirmed by theoretical and experimental research. The first book containing a million of quantum random digits—generated by using radioactive decay from electronic vacuum tubes—was published by the RAND Corporation in 1955.

Is it quantum randomness provable better then pseudo-randomness? If yes, how much better?

Main problem

Randomness is an essential ingredient for computation.

The pitfalls of software-generated pseudo-randomness are well-known. So, can we do it better?

The obvious candidate is quantum randomness.

Quantum randomness has been confirmed by theoretical and experimental research. The first book containing a million of quantum random digits—generated by using radioactive decay from electronic vacuum tubes—was published by the RAND Corporation in 1955.

Is it quantum randomness provable better then pseudo-randomness? If yes, how much better?

Main problem

Randomness is an essential ingredient for computation.

The pitfalls of software-generated pseudo-randomness are well-known. So, can we do it better?

The obvious candidate is quantum randomness.

Quantum randomness has been confirmed by theoretical and experimental research. The first book containing a million of quantum random digits—generated by using radioactive decay from electronic vacuum tubes—was published by the RAND Corporation in 1955.

Is it quantum randomness provable better then pseudo-randomness? If yes, how much better?

An example

How to organise a large pot-luck party?

Two coin tosses suffice!

Assume that the host contributes with beverages and the food consists of starters, main courses, and sweets & fruits in proportions of 25%, 50%, and 25%.

The any participant can toss a coin at most twice to decide what to bring: if the result of the first toss is a head, then she/he will bring a main course; otherwise, a new toss will decide between starters and sweets & fruits.

An example

How to organise a large pot-luck party?

Two coin tosses suffice!

Assume that the host contributes with beverages and the food consists of starters, main courses, and sweets & fruits in proportions of 25%, 50%, and 25%.

The any participant can toss a coin at most twice to decide what to bring: if the result of the first toss is a head, then she/he will bring a main course; otherwise, a new toss will decide between starters and sweets & fruits.

An example

How to organise a large pot-luck party?

Two coin tosses suffice!

Assume that the host contributes with beverages and the food consists of starters, main courses, and sweets & fruits in proportions of 25%, 50%, and 25%.

The any participant can toss a coin at most twice to decide what to bring: if the result of the first toss is a head, then she/he will bring a main course; otherwise, a new toss will decide between starters and sweets & fruits.

An example

How to organise a large pot-luck party?

Two coin tosses suffice!

Assume that the host contributes with beverages and the food consists of starters, main courses, and sweets & fruits in proportions of 25%, 50%, and 25%.

The any participant can toss a coin at most twice to decide what to bring: if the result of the first toss is a head, then she/he will bring a main course; otherwise, a new toss will decide between starters and sweets & fruits.

Outline

Pure randomness?

There is no such thing as “pure randomness”.

Van der Waerden: In every binary sequence at least one of the two symbols must occur in arithmetical progressions of every length.

Pure randomness?

There is no such thing as “pure randomness”.

Van der Waerden: In every binary sequence at least one of the two symbols must occur in arithmetical progressions of every length.

Four forms of randomness

- Flipping a coin.
- John von Neumann: ‘Anyone who considers arithmetical [read: software generated] methods of producing random digits is, of course, in a state of sin’.
- Algorithmic randomness means “computational incompressibility”.
- Which slit the electron went through in the double slit experiment is “random”. Randomness is intrinsically part of the standard model of quantum mechanics.

Four forms of randomness

- Flipping a coin.
- John von Neumann: ‘Anyone who considers arithmetical [read: software generated] methods of producing random digits is, of course, in a state of sin’.
- Algorithmic randomness means “computational incompressibility”.
- Which slit the electron went through in the double slit experiment is “random”. Randomness is intrinsically part of the standard model of quantum mechanics.

Four forms of randomness

- Flipping a coin.
- John von Neumann: ‘Anyone who considers arithmetical [read: software generated] methods of producing random digits is, of course, in a state of sin’.
- Algorithmic randomness means “computational incompressibility”.
- Which slit the electron went through in the double slit experiment is “random”. Randomness is intrinsically part of the standard model of quantum mechanics.

Four forms of randomness

- Flipping a coin.
- John von Neumann: ‘Anyone who considers arithmetical [read: software generated] methods of producing random digits is, of course, in a state of sin’.
- Algorithmic randomness means “computational incompressibility”.
- Which slit the electron went through in the double slit experiment is “random”. Randomness is intrinsically part of the standard model of quantum mechanics.

Outline

P. Diaconis, Susan Holmes and R. Montgomery showed that the natural process of vigourously flipping a coin which is caught in the hand is biased to come up the same way it started with probability of about 0.51.

Outline

- It is easy and cheap to produce using software. A "random" seed is amplified.
- For many purposes it mimics well randomness.
- It is uniformly distributed.
- It is predictable, but prediction is costly.
- It is extensively used.
- It is mathematically poorly understood.
- Many pitfalls have arisen because of circularity (e.g. Netscape 1995).

- It is easy and cheap to produce using software. A "random" seed is amplified.
- For many purposes it mimics well randomness.
- It is uniformly distributed.
- It is predictable, but prediction is costly.
- It is extensively used.
- It is mathematically poorly understood.
- Many pitfalls have arisen because of circularity (e.g. Netscape 1995).

- It is easy and cheap to produce using software. A "random" seed is amplified.
- For many purposes it mimics well randomness.
- It is uniformly distributed.
- It is predictable, but prediction is costly.
- It is extensively used.
- It is mathematically poorly understood.
- Many pitfalls have arisen because of circularity (e.g. Netscape 1995).

- It is easy and cheap to produce using software. A "random" seed is amplified.
- For many purposes it mimics well randomness.
- It is uniformly distributed.
- It is predictable, but prediction is costly.
- It is extensively used.
- It is mathematically poorly understood.
- Many pitfalls have arisen because of circularity (e.g. Netscape 1995).

- It is easy and cheap to produce using software. A "random" seed is amplified.
- For many purposes it mimics well randomness.
- It is uniformly distributed.
- It is predictable, but prediction is costly.
- It is extensively used.
- It is mathematically poorly understood.
- Many pitfalls have arisen because of circularity (e.g. Netscape 1995).

- It is easy and cheap to produce using software. A "random" seed is amplified.
- For many purposes it mimics well randomness.
- It is uniformly distributed.
- It is predictable, but prediction is costly.
- It is extensively used.
- It is mathematically poorly understood.
- Many pitfalls have arisen because of circularity (e.g. Netscape 1995).

- It is easy and cheap to produce using software. A "random" seed is amplified.
- For many purposes it mimics well randomness.
- It is uniformly distributed.
- It is predictable, but prediction is costly.
- It is extensively used.
- It is mathematically poorly understood.
- Many pitfalls have arisen because of circularity (e.g. Netscape 1995).

Outline

Some characteristics of algorithmic randomness

- It satisfies all computable enumerable statistical properties of randomness.
- It is unpredictable, hence uniformly distributed.
- It is Turing uncomputable.
- Every "decent" Monte Carlo simulation algorithm (like Rabin's primality test) powered with algorithmic randomness produces the result not only true with high probability, but rigourously correct.

Some characteristics of algorithmic randomness

- It satisfies all computable enumerable statistical properties of randomness.
- It is unpredictable, hence uniformly distributed.
- It is Turing uncomputable.
- Every “decent” Monte Carlo simulation algorithm (like Rabin’s primality test) powered with algorithmic randomness produces the result not only true with high probability, but rigourously correct.

Some characteristics of algorithmic randomness

- It satisfies all computable enumerable statistical properties of randomness.
- It is unpredictable, hence uniformly distributed.
- It is Turing uncomputable.
- Every “decent” Monte Carlo simulation algorithm (like Rabin’s primality test) powered with algorithmic randomness produces the result not only true with high probability, but rigourously correct.

Some characteristics of algorithmic randomness

- It satisfies all computable enumerable statistical properties of randomness.
- It is unpredictable, hence uniformly distributed.
- It is Turing uncomputable.
- Every “decent” Monte Carlo simulation algorithm (like Rabin’s primality test) powered with algorithmic randomness produces the result not only true with high probability, but rigourously correct.

Outline

Some characteristics of quantum randomness

- It has been confirmed by theoretical and experimental research.
- It passes all reasonable statistical properties of randomness.
- Can be easily and reliably produced: A photon generated by a source beamed to a semitransparent mirror is reflected or transmitted with 50 per cent chance, and these measurements can be translated into a string of quantum random bits.

Some characteristics of quantum randomness

- It has been confirmed by theoretical and experimental research.
- It passes all reasonable statistical properties of randomness.
- Can be easily and reliably produced: A photon generated by a source beamed to a semitransparent mirror is reflected or transmitted with 50 per cent chance, and these measurements can be translated into a string of quantum random bits.

Some characteristics of quantum randomness

- It has been confirmed by theoretical and experimental research.
- It passes all reasonable statistical properties of randomness.
- Can be easily and reliably produced: A photon generated by a source beamed to a semitransparent mirror is reflected or transmitted with 50 per cent chance, and these measurements can be translated into a string of quantum random bits.

▸ JPict

Some characteristics of quantum randomness

Main result: Quantum randomness is Turing uncomputable.

Main question: Is quantum randomness algorithmic random?

In fact, for many purposes an answer to the weaker question:

Is quantum randomness “partial algorithmic” random?

will be enough.

Some characteristics of quantum randomness

Main result: Quantum randomness is Turing uncomputable.

Main question: Is quantum randomness algorithmic random?

In fact, for many purposes an answer to the weaker question:

Is quantum randomness “partial algorithmic” random?

will be enough.

Algorithmic vs. partial algorithmic randomness

The difference between algorithmic and partial algorithmic can be illustrated by the following examples:

Assume $x_1x_2\cdots x_n\cdots$ is algorithmically random. Then:

- $x_1 0 x_2 0 \cdots x_n 0 \cdots$ is algorithmically $1/2$-random,
- $x_1 0 x_2 \cdots x_{n+\lfloor \log_2 n \rfloor} 0 \cdots$ is algorithmically ε-random, for every computable $\varepsilon \in (0,1)$, but not algorithmically random.

Algorithmic vs. partial algorithmic randomness

The difference between algorithmic and partial algorithmic can be illustrated by the following examples:

Assume $x_1x_2\cdots x_n\cdots$ is algorithmically random. Then:

- $x_1 0 x_2 0 \cdots x_n 0 \cdots$ is algorithmically $1/2$-random,
- $x_1 0 x_2 \cdots x_{n+\lfloor \log_2 n \rfloor} 0 \cdots$

  is algorithmically ε-random, for every computable $\varepsilon \in (0,1)$, but not algorithmically random.

Algorithmic vs. partial algorithmic randomness

The difference between algorithmic and partial algorithmic can be illustrated by the following examples:

Assume $x_1x_2\cdots x_n\cdots$ is algorithmically random. Then:

- $x_10x_20\cdots x_n0\cdots$ is algorithmically $1/2$-random,
- $x_10x_2\cdots x_{n+\lfloor \log_2 n\rfloor}0\cdots$ is algorithmically ε-random, for every computable $\varepsilon \in (0,1)$, but not algorithmically random.

Algorithmic vs. partial algorithmic randomness

The difference between algorithmic and partial algorithmic can be illustrated by the following examples:

Assume $x_1x_2\cdots x_n\cdots$ is algorithmically random. Then:

- $x_1 0 x_2 0 \cdots x_n 0 \cdots$ is algorithmically $1/2$-random,
- $x_1 0 x_2 \cdots x_{n+\lfloor \log_2 n \rfloor} 0 \cdots$

  is algorithmically ε-random, for every computable $\varepsilon \in (0,1)$, but not algorithmically random.

Is quantum randomness “available”?

Before embarking into a difficult and uncertain theoretical program, we need to ask ourselves:

Is quantum randomness “available”?

Is quantum randomness “available”?

Before embarking into a difficult and uncertain theoretical program, we need to ask ourselves:

Is quantum randomness “available”?

Quantis

Quantis: quantum mechanical random number generator produced and sold by *id Quantique* of the University of Geneva

Why the main question is important?

- The question touches a fundamental problem in quantum mechanics: what is the nature of quantum randomness (recall that in the standard model, quantum randomness is postulated, not derived)
- It also revives the question of Turing computability of quantum mechanics as well as the question whether quantum randomness can be used to trespass the Turing's barrier
- But there are practical questions as well: the most important is to evaluate the "quality" of Monte-Carlo simulations powered with quantum randomness

Why the main question is important?

- The question touches a fundamental problem in quantum mechanics: what is the nature of quantum randomness (recall that in the standard model, quantum randomness is postulated, not derived)
- It also revives the question of Turing computability of quantum mechanics as well as the question whether quantum randomness can be used to trespass the Turing's barrier
- But there are practical questions as well: the most important is to evaluate the "quality" of Monte-Carlo simulations powered with quantum randomness

Why the main question is important?

- The question touches a fundamental problem in quantum mechanics: what is the nature of quantum randomness (recall that in the standard model, quantum randomness is postulated, not derived)
- It also revives the question of Turing computability of quantum mechanics as well as the question whether quantum randomness can be used to trespass the Turing's barrier
- But there are practical questions as well: the most important is to evaluate the "quality" of Monte-Carlo simulations powered with quantum randomness

The number 1729

The smallest number expressible as the sum of two cubes in n different ways is called $Taxicab(n)$.

$Taxicab(2) = 1729; Taxicab(5) = 4898865927696 2496.$

The value of $Taxicab(6)$ is not known.

Using a sample of 562,500 quantum random integers drawn by Quantis from the interval $[10^{18}, 24153319581254312065344]$, C.S. Calude, E. Calude and M.J. Dinneen have proved that with probability greater than 99.8%,

$$Taxicab(6) = 24153319581254312065344.$$

The number 1729

The smallest number expressible as the sum of two cubes in n different ways is called $Taxicab(n)$.

$Taxicab(2) = 1729$; $Taxicab(5) = 4898865927696496$.

The value of $Taxicab(6)$ is not known.

Using a sample of 562,500 quantum random integers drawn by Quantis from the interval $[10^{18}, 241533195812543120653 44]$, C.S. Calude, E. Calude and M.J. Dinneen have proved that with probability greater than 99.8%,

$$Taxicab(6) = 241533195812543120653 44.$$

The number 1729

The smallest number expressible as the sum of two cubes in n different ways is called $Taxicab(n)$.

$Taxicab(2) = 1729; Taxicab(5) = 4898659276962496.$

The value of $Taxicab(6)$ is not known.

Using a sample of 562,500 quantum random integers drawn by Quantis from the interval $[10^{18}, 241533195812543120653 44]$, C.S. Calude, E. Calude and M.J. Dinneen have proved that with probability greater than 99.8%,

$$Taxicab(6) = 24153319581254312065344.$$

The number 1729

The smallest number expressible as the sum of two cubes in n different ways is called $Taxicab(n)$.

$Taxicab(2) = 1729; Taxicab(5) = 4898659276962496.$

The value of $Taxicab(6)$ is not known.

Using a sample of 562,500 quantum random integers drawn by Quantis from the interval $[10^{18}, 241533195812543120653 44]$, C.S. Calude, E. Calude and M.J. Dinneen have proved that with probability greater than 99.8%,

$$Taxicab(6) = 24153319581254312065344.$$

The number 1729

The smallest number expressible as the sum of two cubes in n different ways is called $Taxicab(n)$.

$Taxicab(2) = 1729$; $Taxicab(5) = 4898659276962496$.

The value of $Taxicab(6)$ is not known.

Using a sample of 562,500 quantum random integers drawn by Quantis from the interval $[10^{18}, 24153319581254312065344]$, C.S. Calude, E. Calude and M.J. Dinneen have proved that with probability greater than 99.8%,

$$Taxicab(6) = 24153319581254312065344.$$

Outline

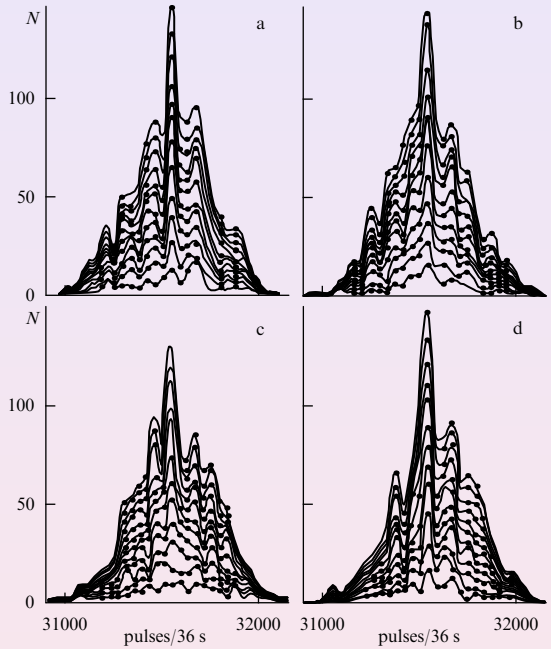


Figure 1. Illustration of the non-randomness of the fine structure of distribution of results of measurements of radioactivity. Four histograms are plotted without shifting and smoothing, each from the results of 1200 consecutive measurements of the radioactivity of a ^{55}Fe preparation. Measured with a scintillation counter and the amplitude analyzer ORTEC by counting the secondary x-ray quanta at 5.9 keV and 6.3 keV which accompany the K-capture associated with the ^{55}Fe to ^{55}Mn transformation. The mean activity is about 31500 pulses per 36 seconds. The steps along the horizontal axis are 30 pulses. Layer lines are drawn after each 100 measurements.

Shnoll's fluctuations showing the non-randomness of the fine structure of distributions of results of measurements of radioactivity, reported in 1998.

NIST statistical test

The NIST test for Quantis (1 million quantum random bits)

Test name	Mean of p-value	Variance	Conclusion
Approximate Entropy Test	0.489	0.088	SUCCESS
Frequency Test within a Block	0.506	0.081	SUCCESS
Cumulative Sums Test	0.499	0.081	SUCCESS
Discrete Fourier Transform (Spectral) Test	0.493	0.079	SUCCESS
Binary Matrix Rank Test	0.498	0.084	SUCCESS
Run Test	0.497	0.081	SUCCESS
Serial Test	0.495	0.078	SUCCESS
Maurer's Universal Statistical Test	0.493	0.081	SUCCESS
Linear Complexity Test	0.499	0.083	SUCCESS
Test for the Longest Run of Ones in a Block	0.503	0.087	SUCCESS
Non-overlapping Template Matching Test	0.499	0.082	SUCCESS
Overlapping Template Matching Test	0.490	0.081	SUCCESS
Frequency (Monobit) Test	0.505	0.084	SUCCESS
Lempel-Ziv Compression Test	0.480	0.080	SUCCESS
Random Excursions Test	0.503	0.083	SUCCESS
Random Excursions Variant Test	0.502	0.082	SUCCESS

DIEHARD test

The DIEHARD for Quantis

Test name	p-value	Conclusion
Birthday Spacing Test	0.493	SUCCESS
Overlapping 5-Permutation Test	0.679	SUCCESS
Binary Rank Test (31x31 matrices)	0.692	SUCCESS
Binary Rank Test (32x32 matrices)	0.789	SUCCESS
Binary Rank Test (6x8 matrices)	0.730	SUCCESS
Bitstream Test	0.622	SUCCESS
Overlapping-Pairs-Sparse Occupancy Test	0.524	SUCCESS
Overlapping-Quadruples-Sparse-Occupancy Test	0.562	SUCCESS
DNA Test	0.657	SUCCESS
Count-the-1's Test (on stream of bytes)	0.418	SUCCESS
Count-the-1's Test (on specific bytes)	0.557	SUCCESS
Parking Lot Test	0.409	SUCCESS
Minimum Distance Test	0.551	SUCCESS
3D Spheres Test	0.681	SUCCESS
Squeeze Test	0.415	SUCCESS
Overlapping Sums Test	0.460	SUCCESS
Runs Test	0.698	SUCCESS

Outline

Strategy

Because of difficulties related to quantum measurements, all test performed on quantum randomness involved small to relatively small samples, and test were applied only to the quantum source.

Our experiments are information-theoretic, comparative and involve larger data:

- two strings of quantum random bits of length 2^{32}, q1, q2, and
- a string of pseudo-random bits of length 2^{32}, c, generated with C,
- two strings of pseudo-random bits of length 2^{32}, m1, m2, generated with Mathematica,
- the first 2^{32} bits of π.

Strategy

Because of difficulties related to quantum measurements, all test performed on quantum randomness involved small to relatively small samples, and test were applied only to the quantum source.

Our experiments are information-theoretic, comparative and involve larger data:

- two strings of quantum random bits of length 2^{32}, q1, q2, and
- a string of pseudo-random bits of length 2^{32}, c, generated with C,
- two strings of pseudo-random bits of length 2^{32}, m1, m2, generated with Mathematica,
- the first 2^{32} bits of π.

Strategy

Because of difficulties related to quantum measurements, all test performed on quantum randomness involved small to relatively small samples, and test were applied only to the quantum source.

Our experiments are information-theoretic, comparative and involve larger data:

- two strings of quantum random bits of length 2^{32}, q1, q2, and
- a string of pseudo-random bits of length 2^{32}, c, generated with C,
- two strings of pseudo-random bits of length 2^{32}, m1, m2, generated with Mathematica,
- the first 2^{32} bits of π.

Strategy

Because of difficulties related to quantum measurements, all test performed on quantum randomness involved small to relatively small samples, and test were applied only to the quantum source.

Our experiments are information-theoretic, comparative and involve larger data:

- two strings of quantum random bits of length 2^{32}, q1, q2, and
- a string of pseudo-random bits of length 2^{32}, c, generated with C,
- two strings of pseudo-random bits of length 2^{32}, m1, m2, generated with Mathematica,
- the first 2^{32} bits of π.

Test 1

Number of n-bit strings read before all 2^n strings found

1	3	3	2	3	2	3
2	5	13	17	9	4	8
3	13	15	17	25	34	25
4	54	62	52	68	43	37
5	120	104	103	82	110	132
6	225	243	442	279	228	308
7	564	422	758	570	441	753
8	1185	1571	1454	1613	1109	1865
9	3677	2926	3096	3692	3454	2892
10	8964	7973	6006	9900	6720	6952
11	13358	17096	14094	13988	17217	18485
12	32749	39279	32836	38361	38742	45250
13	74587	66850	70310	87066	86169	71285
14	137127	181670	153066	172863	184391	201085
15	328032	301273	346166	314319	315060	349605
16	847256	889388	844576	775828	673381	708482
17	1413895	1487965	1620376	1968527	1711317	1503657
18	3716462	3231268	3771208	3826703	3343063	3320840
19	6992604	7216651	6545990	7157829	7320987	6430151
20	15120999	14372534	14882827	16106197	13630029	14557726
21	31160772	29880910	31671209	28736557	31770079	30036433
22	68792606	64639804	68171469	64045599	65346537	68261628
23	139788457	127688894	133375978	137409826	143073179	123948323

Test 2

All strings of length up to 27 appear in all strings.

Number of missing strings of length n

n	c bits	m1 bits	m2 bits	Π bits	q1 bits	q2 bits
28	34	24	30	31	30	35
29	180595	180055	180320	179708	180411	180181
30	19672741	19674174	19669147	19659988	19666243	19671844

Test 3

A Carmichael number N is a composite number with the property that for every b prime with N we have $b^{N-1} \equiv 1 \bmod N$. We have tested 24683 Carmichael numbers

1	561
2	1105
3	1729
4	2465
5	2821
6	6601
7	8911
...	...
24680	9999447614343265
24681	9999568870200001
24682	9999731048186881
24683	9999924433632001

using Solovay-Strassen probabilistic algorithm powered with bits from the strings c, m1, m1, π, q1 and q2.

Test 3

All 24683 Carmichael numbers were tested n times.
Main entries represent the number of mistakes

n	c bits	m1 bits	m2 bits	Π bits	q1 bits	q2 bits
1	142954	143213	143238	143307	143318	142962
2	88851	89200	89104	88883	88985	88953
3	58435	58940	58767	58515	58750	58684
...						
18	1904	2038	2011	1976	2010	2038
19	1634	1733	1734	1628	1667	1733
20	1396	1543	1495	1474	1498	1543

Tentative conclusions

- In spite of a “theoretical barrier” separating quantum randomness and all other types of pseudo-randomness, it is extremely hard to imagine tests capable of distinguishing these types of randomness.
- The experimental results presented above are tentative and do not have yet statistical significance. More has to be done.
- A number of open questions appear in a natural way.

Tentative conclusions

- In spite of a “theoretical barrier” separating quantum randomness and all other types of pseudo-randomness, it is extremely hard to imagine tests capable of distinguishing these types of randomness.
- The experimental results presented above are tentative and do not have yet statistical significance. More has to be done.
- A number of open questions appear in a natural way.

Tentative conclusions

- In spite of a “theoretical barrier” separating quantum randomness and all other types of pseudo-randomness, it is extremely hard to imagine tests capable of distinguishing these types of randomness.
- The experimental results presented above are tentative and do not have yet statistical significance. More has to be done.
- A number of open questions appear in a natural way.

Open questions

The hybrid computer “PC plus Quantis” (used theoretically to generate an infinite sequence of quantum random bits) trespasses the Turing barrier. The machine exists and was and continue to be used.

1. What is the computational power of the hybrid machine “PC plus Quantis”? If Quantis could generate a “c.e. random sequence”, then the machine “PC plus Quantis” would solve the Halting Problem.
2. How random is quantum randomness? More precisely, to what extent is quantum randomness (partial) algorithmic randomness?

Open questions

The hybrid computer “PC plus Quantis” (used theoretically to generate an infinite sequence of quantum random bits) trespasses the Turing barrier. The machine exists and was and continue to be used.

1. What is the computational power of the hybrid machine “PC plus Quantis”? If Quantis could generate a “c.e. random sequence”, then the machine “PC plus Quantis” would solve the Halting Problem.
2. How random is quantum randomness? More precisely, to what extent is quantum randomness (partial) algorithmic randomness?

Open questions

The hybrid computer “PC plus Quantis” (used theoretically to generate an infinite sequence of quantum random bits) trespasses the Turing barrier. The machine exists and was and continue to be used.

1. What is the computational power of the hybrid machine “PC plus Quantis”? If Quantis could generate a “c.e. random sequence”, then the machine “PC plus Quantis” would solve the Halting Problem.
2. How random is quantum randomness? More precisely, to what extent is quantum randomness (partial) algorithmic randomness?

Open questions

1. How accurate are simulations powered by partial algorithmic random bits?
2. How accurate are simulations powered by quantum random bits?
3. Write software for the hybrid machine “PC plus Quantis”.

Open questions

1. How accurate are simulations powered by partial algorithmic random bits?
2. How accurate are simulations powered by quantum random bits?
3. Write software for the hybrid machine “PC plus Quantis”.

Open questions

1. How accurate are simulations powered by partial algorithmic random bits?
2. How accurate are simulations powered by quantum random bits?
3. Write software for the hybrid machine “PC plus Quantis”.

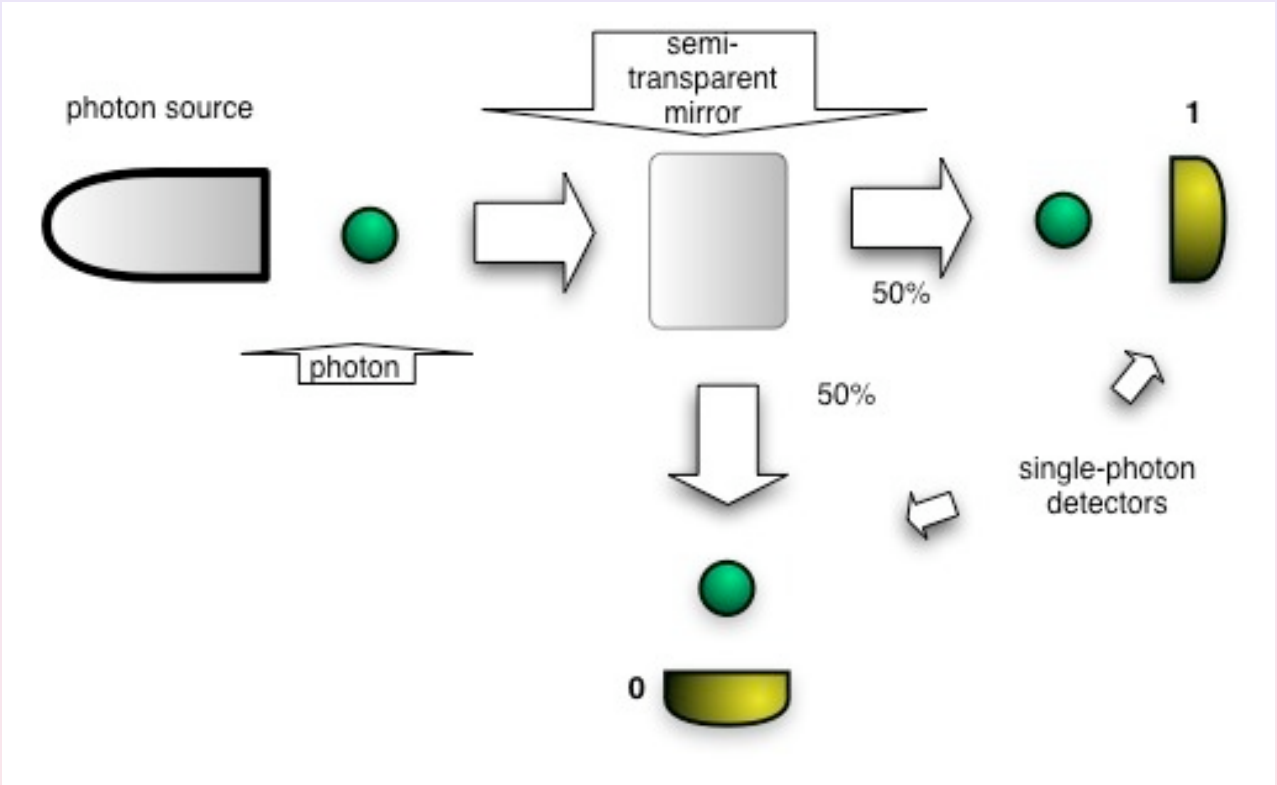
Further Reading


C S. Calude. *Information and Randomness*, 2nd Edition, Revised and Extended, Springer-Verlag, Berlin, 2002.


C. S. Calude. Algorithmic randomness, quantum physics, and incompleteness, in M. Margenstern (ed.). *Proc. MCU'2004*, Lectures Notes in Comput. Sci. 3354, Springer, Berlin, 2005, 1–17.


C. S. Calude, Elena Calude, M. J. Dinneen. What is the value of *Taxicab*(6)?, An Update, *Journal for Multiple-Valued Logic and Soft Computing*, accepted, 2005.


C. S. Calude, J. Casti. The jumble cruncher, *New Scientist*, 25 September 2004, 36–37.



Optical system for generating quantum random bits