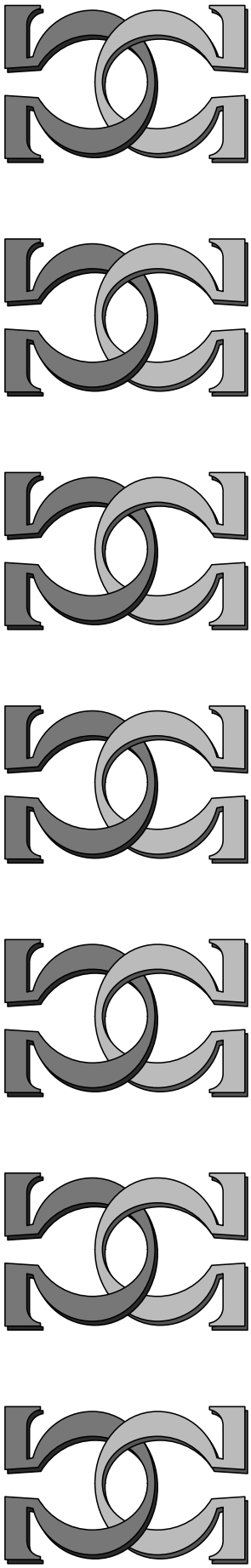

CDMTCS
Research
Report
Series

# Algorithmic Randomness, Quantum Physics, and Incompleteness

**C.S. Calude**
University of Auckland
New Zealand

CDMTCS-248
August 2004

Centre for Discrete Mathematics and
Theoretical Computer Science

*When a distinguished but elderly scientist states that something is possible, he is almost certainly right. When he states that something is impossible, he is almost certainly wrong.* Arthur C. Clarke

# Algorithmic Randomness, Quantum Physics, and Incompleteness

**Cristian S. Calude**

Department of Computer Science
University of Auckland, New Zealand
cristian@cs.auckland.ac.nz

**Abstract.** Is randomness in quantum mechanics "algorithmically random"? Is there any relation between Heisenberg's uncertainty relation and Gödel's incompleteness? Can quantum randomness be used to trespass the Turing's barrier? Can complexity shed more light on incompleteness? In this paper we use variants of "algorithmic complexity" to discuss the above questions.

## 1 Introduction

Whether a $U_{238}$ nucleus will emit an alpha particle in a given interval of time is "random". If we collapse a wave function, what it ends of being is "random". Which slit the electron went through in the double slit experiment, again, is "random".

Is there any sense to say that "random" in the above sentences means "truly random"? When we flip a coin, whether it's heads or tails looks random, but it's not truly random. It's determined by the way we flip the coin, the force on the coin, the way force is applied, the weight of the coin, air currents acting on it, and many other factors. This means that if we calculated all these values, we would know if it was heads or tails without looking. Without knowing this information—and this is what happens in practice—the result *looks* as if it's random, but *it's not truly random.*

*Is quantum randomness "truly random"?* Our working model of "truly random" is "algorithmic randomness" in the sense of Algorithmic Information Theory (see, for example, [5]). In this paper we compare quantum randomness with algorithmic randomness in an attempt to obtain partial answers to the following questions: Is randomness in quantum mechanics "algorithmically random"? Is there any relation between Heisenberg's uncertainty relation and Gödel's incompleteness? Can quantum randomness be used to trespass the Turing's barrier? Can complexity cast more light on incompleteness? Our analysis is tentative and raises more questions than offers answers.

## 2 Algorithmic Randomness

The main idea of *Algorithmic Information Theory* (shortly, AIT) was traced back in time (see [13, 14]) to Leibniz, 1686 ([34], 40–41). If we have a finite

set of points (e.g. say, observations of an experiment), then one can find many mathematical formulae each of which produces a curve passing through them all, in the order that they were given. Can we say that the given set of points satisfy the "law" described by such a mathematical formula? If the set is very large and complex, and the formula is comparatively simpler, then, indeed, we have a law. In Chaitin's words [13], "a scientific theory is a computer program that calculates the observations, and that the smaller the program is, the better the theory."[1] If the mathematical formula *is not* substantially simpler than the data itself, then we don't have a law; still, there may be another mathematical formula qualifying as "law" for the given set. If *no* mathematical formula is substantially simpler than the set itself, the set is unstructured, law-less. Using the computer paradigm, if *no program is substantially simpler than the set itself, then the set is "algorithmically random"*.

Of course, to make ideas precise we need to define the basic notions, complex finite set, (substantially) smaller program, etc. A convenient way is to code all objects as binary strings and use Turing machines as a model of computation.

For technical reasons (see [5, 21]), our model is a *self-delimiting Turing machine*, that is a Turing machine $C$ which processes binary strings into binary strings and has a prefix-free domain: if $C(x)$ is defined and $y$ is either a proper prefix or an extension of $x$, then $C(y)$ is not defined. The self-delimiting Turing machine $U$ is *universal* if for every self-delimiting Turing machine $C$ there exists a fixed binary string $p$ (the simulator) such that for every input $x$, $U(px) = C(x)$: either both computations $U(px)$ and $C(x)$ stop and, in this case they produce the same output or both computations never stop. Universal self-delimiting Turing machines can be effectively constructed. The relation with computability theory is given by the following theorem:

> *A set is computably enumerable* (shortly, *c.e.*) iff *can be generated by some self-delimiting Turing machine.*

The Omega number introduced in [11]

$$\Omega_U = \sum_{U(x) \text{ stops}} 2^{-|x|} = 0.\omega_1\omega_2\ldots\omega_n\ldots \tag{1}$$

is the halting probability of $U$; $|x|$ denotes the length of the (binary) string $x$. Omega is one of the most important concepts in algorithmic information theory (see [5]).

The program-size complexity induced by $C$ is defined by $H_C(x) = \min\{|w| : C(w) = x\}$ (with the convention that strings not produced by $C$ have infinite complexity). The complexity $H_C$ measures the power of $C$ to compress strings. For example, if $H_C(x) \leq |x| - k$, then $C$ can compress at least $k$ bits of $x$; if

[1] The modern approach, equating a mathematical formula with a computer program, would probably not surprise Leibniz, who designed a succession of mechanical calculators, wrote on the binary notation (in 1679) and proposed the famous "let us calculate" dictum; see more in Davis [18], chapter one.

$H_C(x) > |x| - k$, then $C$ cannot compress more than $k-1$ bits of $x$. A string $x$ is algorithmically $k$–random with respect to $C$ if the complexity $H_C(x)$ is maximal up to $k$ among the complexities of all strings of the same length, that is, $H_C(x) \geq \max_{|y|=|x|} H_C(y) - k$.

One might suppose that the complexity of a string would vary greatly between choices of self-delimiting Turing machine. The complexity difference between $C$ and $C'$ is at most the length of the shortest program for $C'$ that simulates $C$. Complexities induced by some self-delimiting Turing machines (called *universal*) are almost optimal, therefore, the complexity of a string is fixed to within an additive constant. This is the "invariance theorem" (see [5], p. 36):

> *For every self-delimiting universal Turing machine $U$ and self-delimiting Turing machine $C$ there exists a constant $\varepsilon > 0$ (which depends upon $U$ and $C$) such that for every string $x$, $H_U(x) \leq \varepsilon + H_C(x)$.*

In what follows we will fix a self-delimiting universal Turing machine $U$, write $H$ instead of $H_U$, and use the term "machine" to denote a "self-delimiting Turing machine".

Algorithmic random strings are defined as above using $U$ instead of $C$. This approach can be extended to "algorithmic random sequences" by requiring that the initial prefixes of the sequence cannot be compressed with more than a fixed number of bits, i.e. they are all "almost random": A sequence $\mathbf{x} = x_1 x_2 \ldots, x_n \ldots$ is *algorithmic random* if there exists a positive constant $c > 0$ such that for all $n > 0$, $H(x_1 x_2 \ldots, x_n) \geq n - c$. Chaitin's theorem [11] states that

> *The bits of $\Omega_U$ (i.e. the sequence $\omega_1 \omega_2 \ldots \omega_n \ldots$ in (1)) form a random sequence.*

## 3 From Algorithmic Randomness to Uncertainty

The randomness of quantum processes has been an integral part of the interpretation of quantum phenomena almost from the very outset of the theory. In fact, quantum physics is the only theory within the fabric of modern physics that integrates and is based on randomness. Quantum randomness has been confirmed over and over again by theoretical and experimental research conducted in physics since the first decades of the 20th century.[2]

But there is a problem: quantum randomness is postulated and is not at all a mathematical consequence of the standard model of quantum mechanics. We don't know whether the randomness of quantum mechanics is genuine or simply an artefact of the particular mathematical apparatus physicists employ to describe quantum phenomena. Being pragmatic, perhaps we can accept the randomness because of the immense success of the applications of quantum mechanics. But even here there is room for doubt. As Wolfram ([54], p. 1064) has

[2] The conclusion of [4] is : "We find no evidence for short- or long-term correlations in the intervals of the quantum jumps or in the decay of the quantum states, in agreement with quantum theory".

pointed out, "a priori, there may in the end be no clear way to tell whether randomness is coming from an underlying quantum process that is being measured, or from the actual process of measurement."

So, *what is the relation between algorithmic randomness and quantum randomness?* A detailed discussion appears in Svozil [51] (see also [50]). Yurtsever [55] argued that a string of quantum random bits is, *almost certainly*, algorithmically random. Here we take a different approach.

First and foremost, there is a strong *computational* similarity: both algorithmic and quantum randomness are *uncomputable*, they cannot be generated/simulated by any machine.[3] From this point of view, both types of randomness are fundamentally different from "deterministic chaos" (computable systems in which unobservably small causes can produce large effects) or pseudo-random numbers (generated by software functions; an elegant solution is the so-called "rule 30" discovered by Wolfram [53]).

The strong uncomputability of algorithmic randomness is expressed by the theorem:

*The set of algorithmic random strings is immune.*

That is, *no infinite set of algorithmic random strings is c.e.* (see [5], p. 119). In particular, the set of prefixes of a random infinite sequence is immune, hence the sequence itself is uncomputable.

Quantum randomness is postulated by Born's measurement postulate: *When a closed quantum physical system in state* $V = (v_{1,1}, v_{2,1}, \ldots, v_{n,1})^T$ *is* measured *it yields outcome* $i$ *with probability* $|v_{i,1}|^2$. In this sense, according to Milburn (see [37], p. 1), the "physical reality is irreducibly random". For Peres [42], "in a strict sense quantum theory is a set of rules allowing the computation of probabilities for the outcomes of tests which follow specific preparations". In the standard model (Copenhagen interpretation) of quantum physics, quantum processes cannot be simulated on a classical Turing machine, not even on a probabilistic Turing machine (in which the available transitions are chosen randomly with equal probability at each step). The reason is Bell's Theorem, which, in Feynman's words ([24], p. 476), reads: *"It is impossible to represent the results of quantum mechanics with a classical universal device."*

A recently proposed complexity-theoretic analysis [9] of Heisenberg's uncertainty principle (see [27]) reveals more facts. The uncertainty principle states that the *more precisely the position is determined, the less precisely the momentum is known in this instant, and vice versa*. In its exact form (first published by Kennard [31]), for all normalized state vectors $|\Psi\rangle$,

$$\Delta_p \cdot \Delta_q \geq \hbar/2,$$

where $\Delta_p$ and $\Delta_q$ are standard deviations of momentum and position, i.e.

$$\Delta_p^2 = \langle\Psi|p^2|\Psi\rangle - \langle\Psi|p|\Psi\rangle^2;\ \Delta_q^2 = \langle\Psi|q^2|\Psi\rangle - \langle\Psi|q|\Psi\rangle^2.$$

[3] It is also very difficult for humans to produce random digits; based on 'history', computer programs can predict, on average, some of the digits humans will write down.

For our analysis it is more convenient to define a variation of the program-size complexity, namely the complexity measure $\nabla_C(x) = \min\{N(w) \mid C(w) = x\}$, the smallest integer whose binary representation produces $x$ via $C$. Clearly, for every string $x$,

$$2^{H_C(x)} \leq \nabla_C(x) \leq 2^{H_C(x)+1} - 1.$$

Therefore we can say that $\Delta_C(x)$, the uncertainty in the value $\nabla_C(x)$, is the difference between the upper and lower bounds given, namely $\Delta_C(x) = 2^{H_C(x)}$.

The invariance theorem can now be stated as follows:

*For every universal machine $U$ and machine $C$ there exists a constant $\varepsilon > 0$ (which depends upon $U$ and $C$) such that for every string $x$, $\Delta_U(x) \leq \varepsilon \cdot \Delta_C(x)$.*

Let $\Delta_s = 2^{-s}$ be the probability of choosing a program of length $s$. Chaitin's theorem (cited at the end of Section 2) stating that the bits of $\Omega_U$ in (1) form a random sequence can now be presented as a "formal uncertainty principle":

*For every machine $C$ there is a constant $\varepsilon > 0$ (which depends upon $U$ and $C$) such that*

$$\Delta_s \cdot \Delta_C(\omega_1 \ldots \omega_s) \geq \varepsilon. \tag{2}$$

The inequality (2) is an uncertainty relation, as it reflects a limit to which we can simultaneously increase both the accuracy with which we can approximate $\Omega_U$ and the complexity of the initial sequence of bits we compute; it relates the uncertainty of the output to the size of the input. When $s$ grows indefinitely, $\Delta_s$ tends to zero in contrast with $\Delta_C(\omega_1 \ldots \omega_s)$ which tends to infinity; in fact, the product is not only bounded from below, but increases indefinitely (see also [9]). From a complexity viewpoint (2) tells us that there is a limit $\varepsilon$ up to which we can uniformly compress the initial prefixes of the binary expansion of $\Omega_U$.

The above "formal uncertainty principle" (much like Heisenberg's uncertainty principle) is a general one; they both apply to *all* systems governed by the wave equation, not just quantum waves. We could, for example, use sound waves instead of a quantum system by playing two pure tones with frequencies $f$ and $f + \Delta_C(\omega_1 \ldots \omega_s)$. Then $\Delta_s$ corresponds to the complementary observable, the length of time needed to perceive a beat.

For the remainder of this section *we assume that quantum randomness is algorithmic randomness.*[4]

[4] This is a disputable assumption. Bohm's interpretation says there are real particles with trajectories determined by a non-local equation, and the randomness is due to our ignorance about the state of the rest of the universe. Penrose says that the wave collapse is deterministic, but uncomputable and occurs when the difference between superposed space-times gets too large. Fredkin, following a tradition that goes back to Schrödinger and Einstein, says the wave collapse is computable and, probably, just a simple pseudo-random function; we have no idea what the structure of space is like at the Planck scale, which is only about $2^{-116}$ metres. Another view sees

The two conjugate coordinates are the random real and the binary numbers describing the programs that generate its prefixes. Then, the uncertainty in the random real given an $n$-bit prefix is $2^{-n}$, and the uncertainty in the size of the shortest program that generates it is, to within a multiplicative constant, $2^n$.

The Fourier transform is a lossless transformation, so all the information contained in the delta function $\delta_{\Omega(x)} = 1$ if $x = \Omega$, $\delta_{\Omega(x)} = 0$, otherwise, is preserved in the conjugate. Therefore, if you need $n$ bits of information to describe a square wave convergent on the delta function, there must be $n$ bits of information in the Fourier transform of the square wave. Since both the information in the transformed square wave and the shortest program describing the square wave increase linearly with $n$, there is an equivalence between the two.

Is (2) a 'true' uncertainty relation? We can prove that the variables $\Delta_s$ and $\Delta_C$ in (2) are standard deviations of two measurable observables in suitable probability spaces, see [9]. For $\Delta_s$ we consider the space of all real numbers in the unit interval which are approximated to exactly $s$ digits. Consider the probability distribution $Prob(v) = P_C(v)/\Omega_C^s$, where $P_C(x) = \sum_{C(y)=x} 2^{-|y|}$ and $\Omega_C^s = \sum_{|x|=s} P_C(x)$. For $\Delta_C$ we consider

$$\beta = (\Delta_C(\omega_1\omega_2\ldots\omega_s))^{1/2} \cdot (Prob(\omega_1\omega_2\ldots\omega_s))^{-1/2} \cdot (1 - Prob(\omega_1\omega_2\ldots\omega_s))^{-1/2},$$

and the same space but the random variable $Y(\omega_1\omega_2\ldots\omega_s) = \beta$ and $Y(v) = 0$ if $v \neq \omega_1\omega_2\ldots\omega_s$. Hence, the relation (2) becomes:

$$\sigma_X \cdot \sigma_Y = \Delta_s \cdot \Delta_C(\omega_1\omega_2\ldots\omega_s) \geq \varepsilon.$$

For example, it is possible to construct a special universal machine $C = U_0$ satisfying the inequality $\Delta_s \cdot \Delta_{U_0}(\omega_1\ldots\omega_s) \geq 1$, for which we have:

$$\sigma_X \cdot \sigma_Y \geq 1.$$

The complexity-theoretic formulation of uncertainty has more "physical meaning", as shown in [9]. If the halting probability of the machine is computable, then we can construct a quantum algorithm to produce a set of qubits whose state is described by the distribution. To illustrate, we consider a quantum algorithm with two parameters, $C$ and $s$, where $C$ is a machine for which the probability of producing each $s$-bit string is computable. We run the algorithm to compute that distribution on a quantum computer with $s$ output qubits; it puts the output register into a superposition of spin states, where the probability of each state $|v\rangle$ is $P_C(v)/\Omega_C^s$. Next, we apply the Hamiltonian operator $H = \beta|\omega_1\ldots\omega_s\rangle\langle\omega_1\ldots\omega_s|$ to the prepared state. A measurement of energy will give $\beta$ with probability $P = Prob(\omega_1\omega_2\ldots\omega_s)$ and zero with probability $1 - P$.

---

the classical world as emerging from the collisional interactions of quantum particles that inherently arise in "hot dense matter". Collisions destroy the purity of otherwise coherent states, so quantum randomness (as well as deterministic chaos) may be a manifestation of the incompleteness of dynamical laws, cf. [39].

The expectation value for energy, therefore, is exactly the same as that of $Y$, but with units of energy, i.e.

$$\Delta_C(\omega_1\omega_2\ldots\omega_s)[J]\cdot\Delta_s \geq \varepsilon[J],$$

where $[J]$ indicates Joules of energy.

Now define

$$\Delta_t \equiv \frac{\sigma_Q}{|d\langle Q\rangle/dt|},$$

where $Q$ is any observable that does not commute with the Hamiltonian; that is, $\Delta_t$ is the time it takes for the expectation value of $Q$ to change by one standard deviation. With this definition, the following is a form of Heisenberg's uncertainty principle:

$$\Delta_E\cdot\Delta_t \geq \hbar/2.$$

We can replace $\Delta_E$ by $\Delta_C(\omega_1\omega_2\ldots\omega_s)$ by the analysis above; but what about $\Delta_t$? If we choose a time scale such that our two uncertainty relations are equivalent for a single quantum system corresponding to a computer $C$ and *one* value of $s$, then the relation holds for $C$ and *any* value of $s$:

$$\Delta_C(\omega_1\omega_2\ldots\omega_s)[J]\cdot\Delta_s\frac{\hbar}{2\varepsilon}[J^{-1}\cdot Js] \geq \frac{\hbar}{2}[Js].$$

In this sense, Heisenberg's uncertainty relation is equivalent to (2).

*The uncertainty principle now says that getting one more bit of $\Omega_U$ requires (asymptotically) twice as much energy.* Note, however, that we have made an arbitrary choice to identify energy with complexity. We could have chosen to create a system in which the position of a particle corresponded to the complexity, while momentum corresponded to the accuracy of $C$'s estimate of $\Omega_U$. In that case, the uncertainty in the position would double for each extra bit. Any observable can play either role, with a suitable choice of units.

If this were the only physical connection, one could argue that the result is merely an analogy and nothing more. However, consider the following: let $\rho$ be the density matrix of a quantum state. Let $R$ be a computable positive operator-valued measure, defined on a finite-dimensional quantum system, whose elements are each labelled by a finite binary string. Then the statistics of outcomes in the quantum measurement is described by $R$: $R(\omega_1\ldots\omega_s)$ is the measurement outcome, and $tr(\rho R(\omega_1\ldots\omega_s))$ is the probability of getting that outcome when we measure $\rho$. Under these hypotheses, Tadaki's inequality (1) (see [52], p. 2), and the relation (2) imply the existence of a constant $\tau$ (depending upon $R$) such that for all $\rho$ and $s$ we have:

$$\Delta_s\cdot\frac{1}{tr(\rho R(\omega_1\ldots\omega_s))} \geq \tau.$$

In other words, there is no algorithm that, for all $s$, can produce an experimental set-up to produce a quantum state and a POVM (positive operator valued measure) with which to measure the state such that the probability of getting the result $\omega_1\omega_2\ldots\omega_s$ is greater than $1/(\tau 2^s)$.

The above analysis is just one small step towards understanding the nature of quantum randomness—more theoretical and experimental facts are needed. One possible avenue of attack might be to experimentally test whether quantum random bits satisfy some properties proven for algorithmic random strings. For example, one such natural property states the existence of a constant $c$ such that for every $n$, the number of algorithmically random strings of length $n$ is greater than $2^{n-c}$.

## 4 Randomness and Computation

As we have seen, there is no such thing as "software generated" genuine randomness. In John von Neumann's words: "Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin". On the other hand, randomness in quantum mechanics is hardly news. So what prevented quantum physics from becoming a dominant source of randomness?

Basically, *practical engineering considerations*. Until recently, the only quantum random number generators were based on the observation of radioactive decay in an element like radium. 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, [45]. The basic table was produced during May and June 1947; exhaustive tests found small but statistically significant biases and adjustments were made. Some of the early methods can be found in Golenko [26] who describes noise generators based on a germanium triode, on a gas-discharge tube with magnet, on an electronic trigger circuit with a switch in its anode supply (photograph in Fig. 44), on a gasotron with magnet, and on subharmonic generators. But such generators are quite bulky and the use of radioactive materials may cause health problems.

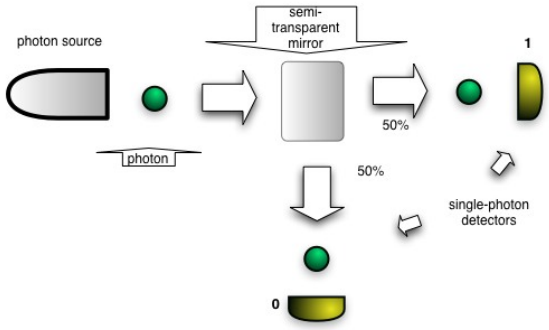


**Fig. 1.** Optical system for generating quantum random bits

Fortunately, a beam of light offers an excellent alternative source of randomness (see [30]). Light consists of elementary particles called photons; they exhibit in certain situations a random behaviour. The transmission upon a semitransparent mirror is an example. A photon generated by a source beamed to a semitransparent mirror is reflected or transmitted with 50 per cent chance (see Fig. 1), and these measurements can be translated into a string of quantum random bits. Such a device can be (and was) manufactured, and its functioning can be monitored in order to confirm that is operating properly.

A spin-off company from the University of Geneva, *id Quantique*, [28], markets a quantum mechanical random number generator called *Quantis*, see Fig. 2. *Quantis* is available as an OEM component which can be mounted on a plastic circuit board or as a PCI card; it can supply a (theoretically, arbitrarily) long string of quantum random bits sufficiently fast for cryptographic applications.[5]

**Fig. 2.** Quantis: OEM and PCI, cf. [29]

*Plug these quantum random bits into a PC and we can, in theory at least, leapfrog Turing's barrier,* that is *we obtain a computing device with capability surpassing that of classical Turing machines.* Indeed, as we have already noticed, no Turing machine can generate quantum random bits! So, the above statement is true *independently* of whether quantum random bits are or not algorithmically random.

Is this interesting? For some authors, the analysis of this type of 'oracle' machine is pointless and "one can only pity those engaged in this misguided enterprise" (cf. [19], p. 207). As the reader arriving at this point can expect, I do not share this view.

First, it seems that the computing device "PC plus a quantum generator of random bits", whose existence can be hardly doubted, is a serious threat to the Church-Turing Thesis, which, in one variant, states that *every effective computation can be carried out by a Turing machine.*

[5] *id Quantique* also supplies quantum random numbers over the internet [40], as well as *HotBits*, [25], which generates them via radioactive decay.

Secondly, understanding this device may help coping with complex computations. Here is a relevant example. Testing whether a number is prime—showing that it has no factors beside itself and 1—is a crucial process in cryptography, and although theoretically fast deterministic algorithms for primality testing have been discovered (see [1][6]), in practice they are quite slow and do not pose any immediate risk to the security of electronic communication.

Probabilistic algorithms, first discovered in the mid 1970s, [41, 44], can help speed things up, but such probabilistic tests—which essentially use a coin-flipping source of pseudo-random bits to search for a number's factors—are only "probably" correct. [7] If you run the probabilistic algorithm using a source of algorithmically random bits, however, it would not only be fast, it would also *be correct every single time* (cf [15]). One of the principal tools used in computer simulation, known as fast Monte-Carlo algorithms, can derive a similar benefit from the use of algorithmically random numbers (cf. [10]; see more in [5]). *It is an open question whether these results are true for quantum random bits.*

Of course, quantum random bits may be imperfect in a *practical setting*. For example, as time goes on, the number of radioactive nuclei in a radioactive substance decreases. A quantum binary random generator may be become biased when the probability of one outcome is not equal to the probability of the other outcome. It is however less of a problem than one might expect at first sight. Post-processing algorithms can be used to remove bias from a sequence of quantum random numbers affected by bias. The simplest unbiasing procedure was first proposed by von Neumann [38].[8] The bits of a sequence are grouped in strings of two bits. The strings 00 and 11 are discarded; the string 01 is replaced by 0 and the string 10 is replaced by 1. After this procedure, the bias is removed from the sequence. The cost of applying an unbiasing procedure to a sequence is that it is shortened; in the case of von Neumann's procedure, the length of the unbiased sequence will be at most 25% of the length of the raw sequence. Other, more efficient, unbiasing procedures exist. Peres [43] proved that the number of bits produced by iterating von Neumann's procedure is arbitrarily close to the entropy bound.

Thirdly, another open question is: *Exactly how much more powerful a Turing machine working with "an oracle of quantum random bits" can be?* [9] This "machine" (which is different from the classical probabilistic Turing machine) can, at any time of the computation, ask the "quantum oracle" to supply an arbitrarily long (but finite) quantum random string. It won't have access to an infinite

[6] The asymptotic time complexity of the algorithm is bounded by $(\log n)^{12} q(\log \log n)$, where $q$ is some polynomial.

[7] See [22] for pitfalls in using traditional pseudo-random number generation techniques and [20] for Nescape error.

[8] Unbiasing is a compression procedure.

[9] The idea of computing with deterministic chaos was investigated in [47, 48].

sequence, but (theoretically) to an unbounded finite set of quantum random bit strings.[10] *Can this immense power be exploited?* [11]

A superficial attack suggests that it is unlikely that a Turing machine augmented with a source of quantum random strings will be capable of solving the Halting Problem: even stepping across the Turing barrier is no guarantee of being able to solve the Halting Problem. But what is the Halting Problem, and why it is the "Philosopher's Stone" of computer science?

The Halting Problem is the problem to decide whether an arbitrarily specified Turing machine halts after a finite number of steps for a given input; the problem cannot be solved by any Turing machine, as Turing proved in 1936! Solving this problem would open a huge box of knowledge. For example, assume you want to know whether every even number greater than 3 is the sum of two primes. We can see that $4 = 2+2, 6 = 3+3, 8 = 3+5$, and so on. In fact, this conjecture has been verified computationally for numbers up to several billion. But is it true for all natural numbers?

One can construct a Turing machine that generates the numbers $4, 6, 8, \ldots$ one after another and checks each of them to see whether it has the above property or not. The machine stops when it finds the first number not having the property; otherwise it continues on to the next number. Clearly, knowing whether this machine stops or not answers the original question. Incidentally, this question is one of the oldest unsolved problems in number theory, a question formulated by Goldbach 262 years ago in a letter to Euler (see [35]). Many other problems can be solved in a similar manner, including the famous Riemann Hypothesis (see [17]), as Matiyasevich proved in [36], p. 121–122.[12]

All theoretical proposals for transcending the Turing barrier (see, for example, [23, 8, 32]) have been challenged on grounds of physical in-feasibility (see [19]): they require infinite time, infinite memory resources (or both), infinite precision measurements, etc. It's ironic that we now have a method that works in the physical world, but one that seems difficult to justify mathematically because it rests on the assumption that quantum processes are genuinely random. The relation between the Riemann Hypothesis and quantum randomness seems to be more profound (see [46] and [7], chapter 11). Maybe it's not a random fact that in both of them as well as in the fast Monte-Carlo simulation, primes play a central role.

---

[10] The insight provided by the refutation (see [33, 16]) of Bennett and Gill's "Random Oracle Hypothesis", [3]—which basically states that the relationships between complexity classes which hold for almost all relativized worlds must also hold in the unrelativized case—suggests that random oracles are extremely powerful; contrast this scenario with the behaviour of probabilistic primality tests run with algorithmically random bits, cf. [15].

[11] Related results can be found in [2].

[12] A rough estimation shows that solving the Goldbach Conjecture is equivalent to deciding the halting status of a RAM program of less than 2,000 bits; for Riemann Hypothesis the program will have about 10,000 bits.

## 5 Complexity and Incompleteness

Gödel's incompleteness theorem states that every finitely-specified consistent theory which is strong enough to include arithmetic is either inconsistent, incomplete or both. Zermelo-Fraenkel set theory with the Axiom of Choice ($ZFC$) is such a theory.

Gödel's original proof as well as most subsequent proofs are based on the following idea: a theory which is consistent and strong enough can express statements about provability within the theory, statements which are not provable by the theory, but which through a proof by contradiction, turn out to be true. This type of proof of incompleteness does not answer the questions of whether independence (a true and unprovable statement is called independent) is a widespread phenomenon nor which kinds of statements can be expected to be independent.

Recall that we fixed the universal machine $U$ and $H$ denotes $H_U$.

The first complexity-theoretic version of incompleteness was discovered by Chaitin [11]:

> *Consider a consistent, sound, finitely-specified theory strong enough to formalise arithmetic and denote by $\mathcal{T}$ its set of theorems. Then, there exists a constant $M$, which depends upon $U$ and $\mathcal{T}$, such that whenever the statement "$H(x) > n$" is in $\mathcal{T}$ we have $n \leq M$.*

As the complexity $H(s)$ is unbounded, each true statement of the form "$H(x) > m$" with $m > M$ (and, of course, there are infinitely many such statements) is unprovable in the theory.

The high $H$-complexity of the statements "$H(x) > m$" with $m > M$ is a source of their unprovability. Is every true statement $s$ with $H(s) > M$ unprovable by the theory? Unfortunately, the answer is *negative* because only finitely many statements $s$ have complexity $H(s) \leq M$ in contrast with the fact that the set of all theorems of the theory is infinite. For example, $ZFC$ or Peano Arithmetic trivially prove all statements of the form "$n + 1 = 1 + n$", but the $H$-complexity of the statement "$n + 1 = 1 + n$" grows unbounded with $n$.

It is an open question whether the "heuristic principle" proposed by Chaitin in [12], p. 69, namely that

> *a set of axioms of complexity $N$ cannot yield a theorem of complexity substantially greater than $N$*

can be proven for an adequate and interesting complexity measure.

An "approximation" of this principle supported by Chaitin's proof in [11] is the following (see [12], p. 69):

> *one cannot prove, from a set of axioms, a theorem that is of greater $H$–complexity than the axioms and know that one has done it.*

We are now in a position to show that the formal uncertainty principle discussed in Section 3 implies incompleteness. Indeed, using the complexity $\nabla$ we can re-obtain (for proofs see [9]) Chaitin's incompleteness result [11] for Omega:

*Consider a consistent, sound, finitely-specified theory strong enough to formalise arithmetic. Then, we can effectively compute a constant $N$ such that the theory cannot determine more than $N$ scattered digits of $\Omega_U = 0.\omega_1\omega_2\ldots$*

The complexity-theoretic characterisation of the randomness of $\Omega_U$, recast as a "formal uncertainty principle" in terms of the complexity $\nabla$, implies Chaitin's information-theoretic version of incompleteness for $\Omega_U$. This shows that *uncertainty implies algorithmic randomness which, in turn, implies incompleteness.* In terms of $\delta$–complexity, high complexity is a source of incompleteness which implies that probabilistically incompleteness is not artificial—it's ubiquitous, pervasive.

We can ask ourselves: How large is the constant $N$ in the above theorem? The answer depends on the chosen universal machine $U$. Indeed, in Calude [6] one proves the following result:

*Consider a consistent, sound, finitely-specified theory strong enough to formalise arithmetic. Then, for each universal machine $U$ we can effectively construct a universal machine $W$ such that $\Omega_U = \Omega_W$ such that the theory can determine at most the initial run of 1's in the expansion of $\Omega_U = 0.11\ldots1\mathbf{0}\ldots$.*

As soon as the first **0** appears, the theory becomes useless. If $\Omega_V < 1/2$, then the binary expansion of $\Omega_V$ starts with 0, and so we obtain Solovay's theorem [49]:

*Consider a consistent, sound, finitely-specified theory strong enough to formalise arithmetic. There effectively exists a universal machine $V$ such that the theory can determine no digit of $\Omega_V$.*

We finally note that a Turing machine working with an "oracle of quantum random bits" will outperform a standard Turing machine in generating mathematical theorems from any given set of axioms. Still, even this machine cannot generate all true statements of arithmetic.

## 6 Final Comments

Is the question "Why did the electron go through this slit instead of the other one?", as unanswerable as the question "Why the $n$th bit of $\Omega_U$ is zero?"? This is a difficult question and we don't answer it; the paper brings some (pale) light into this rather dark picture. Namely, we showed that uncertainty implies algorithmic randomness which, in turn, implies incompleteness. For the machines $C$ whose halting probabilities $\Omega_C$ are computable, one can construct a quantum computer for which the uncertainty relation describes conjugate observables. Therefore, in these particular instances, the uncertainty relation is equivalent to Heisenberg's.

We have also argued that even in case quantum randomness is weaker than algorithmic randomness, still the "Turing machine augmented with a source of quantum random bits" is more powerful than any Turing machine. This suggests a new attack on the Church-Turing Thesis, and the following interesting (from both practical and theoretical points of view) open question: *how much power has this hybrid machine?* Finally, we have discussed the role of complexity (in particular, algorithmic randomness) in understanding incompleteness.

## Acknowledgment

I am much indebted to the chairs of the fourth edition of the conference "Machines, Computation and Universality", Anatoly Beltiukov, Nikolai Kossovskii and Maurice Margenstern, for their invitation to give this talk. I am very grateful to John Casti, Greg Chaitin, Tien Kieu, David Oliver, Mike Stay, Karl Svozil and Garry Tee for illuminating discussions.

## References

1. M. Agrawal, N. Kayal, N. Saxena. PRIMES is in P, `http://www.cse.iitk.ac.in/primality.pdf`, 6 August 2002.
2. E. Allender, H. Buhrman, M. Koucký. What can be efficiently reduced to the Kolmogorov-random strings? *Electronic Colloquium on Computational Complexity, Report 44*, 2004, 19 pp.
3. C. H. Bennett, J. Gill. Relative to a random oracle $A$, $P^A \neq NP^A \neq$ co-$NP^A$ with probability 1, *SIAM Journal on Computing* 10, 1(1981), 96–113.
4. D. J. Berkeland, D. A. Raymondson, V. M. Tassin. Tests for non-randomness in quantum jumps, Los Alamos preprint archive, `http://arxiv.org/abs/physics/0304013`, 2 April 2004.
5. C. S. Calude. *Information and Randomness. An Algorithmic Perspective*, Springer Verlag, Berlin, 2nd Edition, Revised and Extended, 2002.
6. C. S. Calude. Chaitin $\Omega$ numbers, Solovay machines and incompleteness, *Theoret. Comput. Sci.* 284 (2002), 269–277.
7. C. Calude, P. Hertling, B. Khoussainov. Do the zeros of Riemann's zeta–function form a random sequence ? *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS* 62 (1997), 199–207.
8. C. S. Calude, B. Pavlov. Coins, quantum measurements, and Turing's barrier, *Quantum Information Processing* 1, 1–2 (2002), 107–127.
9. C. S. Calude, M. A. Stay. From Heinsenberg to Gödel via Chaitin, *International Journal of Theoretical Physics*, accepted. E-print as *CDMTCS Research Report* 235, 2004, 15 pp. and Los Alamos preprint archive, `http://arXiv:quant-ph/0402197`, 26 February 2004.
10. C. Calude, M. Zimand. A relation between correctness and randomness in the computation of probabilistic algorithms, *Internat. J. Comput. Math.* 16 (1984), 47–53.
11. G. J. Chaitin. A theory of program size formally identical to information theory, *J. Assoc. Comput. Mach.* 22 (1975), 329–340.

12. G. J. Chaitin. *Information–Theoretic Incompleteness*, World Scientific, Singapore, 1992.
13. G. J. Chaitin. *Leibniz, Information, Math and Physics*, `http://www.cs.auckland.ac.nz/CDMTCS/chaitin/kirchberg.html`.
14. G. J. Chaitin. *META MATH! The Quest for Omega*, Pantheon Books, New York, 2005 (to appear).
15. G. J. Chaitin, J. T. Schwartz. A note on Monte-Carlo primality tests and algorithmic information theory, *Comm. Pure Appl. Math.* 31(1978), 521–527.
16. R. Chang, B. Chor, O. Goldreich, J. Hartmanis, J. Hastad, D. Ranjan, P. Rohatgi. The random oracle hypothesis is false, *J. Comput. System Sci.* 49, 1 (1994), 24–39.
17. `http://www.claymath.org/millennium/Riemann_Hypothesis/`.
18. M. Davis. *The Universal Computer: The Road from Leibniz to Turing*, Norton, New York, 2000.
19. M. Davis. The myth of hypercomputation, in C. Teuscher (ed.). *Alan Turing: Life and Legacy of a Great Thinker*, Springer-Verlag, Heidelberg, 2003, 195–211.
20. J-P. Delahaye. *L'Intelligence and le Calcul*, BELIN, Pour la Science, Paris, 2002.
21. R. Downey, D. Hirschfeldt. *Algorithmic Randomness and Complexity*, Springer-Verlag, Heidelberg, 2005 (to appear).
22. D. E. Eastlake 3rd, S. Crocker, J. Schiller, *Randomness Recommendations for Security*, RFC 1750, December 1994, 30 pp.
23. G. Etesi, I. Németi. Non-Turing computations via Malament-Hogarth space-times, *International Journal of Theoretical Physics* 41 (2002), 341-370.
24. R. P. Feynman. Simulating physics with computers, *International Journal of Theoretical Physics* 21 (1982), 467–488.
25. `http://www.fourmilab.ch/hotbits/`.
26. D. I. Golenko. Generation of uniformly distributed random variables on electronic computers, in Yu. A. Shreider (ed.), translated from Russian by G. J. Tee. *The Monte Carlo Method: The Method Statistical Trials*, Pergamon Press, Oxford, 1966, 257–305.
27. W. Heisenberg. Über den Anschaulichen Inhalt der Quantentheoretischen Kinematik und Mechanik, *Zeitschrift für Physik* 43 (1927), 172–198. English translation in J. A. Wheeler, H. Zurek (eds.). *Quantum Theory and Measurement*, Princeton Univ. Press, Princeton, 1983, 62–84.
28. `http://www.idquantique.com/`.
29. `http://www.idquantique.com/img/QuantisBoth.jpg`.
30. T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, A. Zeilinger. A fast and compact quantum random number generator, *Rev. Sci. Instr.* 71 (2000), 1675–1680.
31. E. H. Kennard. Zur Quantenmechanik einfacher Bewegungstypen, *Zeitschrift für Physik* 44 (1927), 326–352.
32. T. D. Kieu. Computing the non-computable, *Contemporary Physics* 44, 1 (2003), 51–71.
33. S. A. Kurtz. On the random oracle hypothesis, *Information and Control* 57, 1 (1983), 40–47.
34. G. W. Leibniz. *Discours de métaphysique*, Gallimard, Paris, 1995.
35. `http://www.mathstat.dal.ca/~joerg/pic/g-letter.jpg`.
36. Yu. V. Matiyasevich. *Hilbert's Tenth Problem*, MIT Press, Cambridge, MA, 1993.
37. G. Milburn. *The Feynman Processor. An Introduction to Quantum Computation*, Allen & Unwin, St. Leonards, 1998.
38. J. von Neumann. Various techniques used in connection with random digits, *National Bureau of Standards Applied Mathematics Series* 12 (1951), 36–38.

39. D. Oliver. Email to C. Calude, 20 August 2004.
40. http://www.randomnumbers.info.
41. G. L. Miller. Riemann's hypothesis and tests of primality, *J. Comput. System Sci.* 13 (1976), 300–317.
42. A. Peres. *Quantum Theory: Concepts and Methods*, Kluwer Academic Publishers, Dordrecht, 1993.
43. Y. Peres. Iterating von Neumann's procedure for extracting random bits, *Ann. Stat.* 20 (1992), 590–597.
44. M. O. Rabin. Probabilistic algorithms, in J. F. Traub (ed.). *Algorithms and Complexity, New Directions and Recent Results*, Academic Press, New York, 1976, 21–39.
45. *A Million Random Digits with 100,000 Normal Deviates*, The RAND Corporation, The Free Press, Glencoe, IL, 1955; online edition: http://www.rand.org/publications/classics/randomdigits/.
46. M. du Sautoy. *The Music of the Primes*, HarperCollins, New York, 2003,
47. S. Sinha, W. L. Ditto. Dynamics based computation, *Physical Letters Review* 81, 10 (1998), 2156–2159.
48. S. Sinha, W. L. Ditto. Computing with distributed chaos, *Physical Review E* 60, 1 (1999), 363–377.
49. R. M. Solovay. A version of $\Omega$ for which $ZFC$ can not predict a single bit, in C. S. Calude, G. Păun (eds.). *Finite Versus Infinite. Contributions to an Eternal Dilemma*, Springer-Verlag, London, 2000, 323–334.
50. K. Svozil. The quantum coin toss-testing microphysical undecidability, *Physics Letters* A143, 433–437.
51. K. Svozil. *Randomness & Undecidability in Physics*, World Scientific, Singapore, 1993.
52. K. Tadaki. Upper bound by Kolmogorov complexity for the probability in computable POVM measurement, Los Alamos preprint archive, http://arXiv:quant-ph/0212071, 11 December 2002.
53. S. Wolfram. Statistical mechanics of cellular automata, *Reviews of Modern Physics* 55 (1983), 601–644.
54. S. Wolfram. *A New Kind of Science*, Wolfram Media, Champaign, IL, 2002.
55. U. Yurtsever. Quantum mechanics and algorithmic randomness, *Complexity* 6, 1 (2002), 27–31.