"Algorithmic Randomness, Quantum Physics, and Incompleteness"

Source: http://sci.tech−archive.net/Archive/sci.logic/2006−05/msg00492.html

---

- *From*: Stephen Harris <cyberguard−1048@xxxxxxxxx>
- *Date*: Sat, 13 May 2006 20:52:33 GMT

---

www.idquantique.com/products/files/quantis−mcu04.pdf

"Algorithmic Randomness, Quantum Physics, and Incompleteness"
by Cristian S. Calude
"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 Goedel s incompleteness?"

Another (earlier I think) related paper,:
www.cs.auckland.ac.nz/CDMTCS/researchreports/235cris.pdf
"From Heisenberg to Goedel via Chaitin" C.S. Calude & M.A. Stay
Mike Stay wrote:
...."we emphatically did NOT show that HUP implies Goedel's
theorem (we showed that AUP, the algorithmic uncertainty
principle, implies Goedel's theorem), and we only showed
that AUP and HUP were equivalent for computable distributions."

−−−

I was thinking of Torkel Franzen when I composed this post.
He was kind enough to review my website on "Randomness" and
corrected several of my misunderstandings regarding Goedel's
Incompleteness Theorems. This is my limited capability
tribute which no doubt would have fallen prey to Torkel.

−−−

When I researched my randomness website project, I noticed that
quite a few books on the subject of randomness failed to define
randomness, the definition is elusive. Every definition seems to
have some shortcoming and I am not sure about the definition
provided by Algorithmic Information Theory (AIT): quantis−mcu04.pdf

"Using the computer paradigm, if no program is substantially simpler
than the set itself, then the set is "algorithmically random".

(which is perhaps equivalent to "For a random string x, the shortest program whose output is x is simply "print x".")

––

I'm going to offer my understanding of randomness.

Defining randomness is a bit paradoxical. People will say that randomness means that there is no unique pattern. However, the way that this is accomplished conceptually is all finite sequences are to be found infinitely often and permutated or concatenated in infinitely many different finite sequences. Truly random numbers differ by their first initial digit but otherwise they all share the property of including all finite sequences, though they do not appear in the same order because then they would not be different members of the infinite set of random numbers. At some point, the finite prefix of different random numbers must diverge or the numbers are the same number.

Thus you find a million, a billion, or a trillion etc of the digital expansions of Pi in every truly random number repeated infinitely often, interspersed or interwoven with all other possible finite patterns of varying lengths.

The finite prefixes of a random number can be partially generated algorithmically, but not completed since all random numbers are of infinite length. A finite sequence can never be proven random; one can say a finite sequence was generated by random means, but that finite sequence can have the same expansion as Pi or other computable numbers. Truly random numbers are proven to exist by a diagonal argument, but individual numbers cannot be proven to be random by examining their structure since their structure is infinitely complex and nonterminated.

Just using random numbers with initial digits 1 thru 9, the first finite digit of all possible expansions of the infinity of random numbers can be listed:

1,2,3,4,5,6,7,8,9 the second iteration produces

11,12,13,14,15,16,17,18,19
21,22,23,24,25,26,27,28,29
31,32,33, etc 39
41,42,43, etc 49
51,52,53, etc 59
61,62,63, etc 69

71,72,73, etc 79
81,82,83, etc 89
91,92,93, etc 99

and the third iteration produces
111,112,113, etc
121,122,123, etc
131,132,133, etc

etc

191,192,193, 199

211, 212 213
221, 222, 223, etc
291,292,293 etc 299

and then on to all possible 3,4,5,6,7,8,9 three digit expansions until they are exhausted, then one can go on to listing all four digit, five digit, six digit and so on, finite expansions until they are exhausted.

The finite listing generated with this method will form a great tree with every possible digital expansion enumerated. If you iterate this tree with branches out to a million digits, then you will have listed all possible initial finite prefix sequences for a random number which is displaying all finite sequences of one to a million digits which comprise itself. Though itself, the random numbers is actually of infinite length, comprised of an infinite variety of finite sequences of infinitely different finite lengths.

A random number cannot exclude any particular finite sequence or that would establish a unique pattern. So a truly random number contains all finite sequences.

I brought this up to show that truly random numbers enjoy the property of equinumerosity. One can see this from the method of generating all sequences an unboundedly finite length. All digits get added to the collective of digits by adding a 1 or 2 or 3 or 4 or 5 or 6 or 7 or 8 or 9 to every tip of the next iterated branch. This equal contribution of each digit never diminished as the sequences grow ever longer as they approach to infinity.

This method generates all the possible finite sequences contained in an infinite random number whose initial finite prefix sequences are becoming longer and more complex. But of course a truly random number will not have its inner structure identical in order to this construction method since there are infinite randoms. The huge tree with many branches will have infinitely many permuations as the finite expansion continues.

This can be accomplished by having a quantis random device mentioned by Calude interweave or juxtapose different areas and lengths of all the finite sequences of the entire tree so that the algorithmic structure is randomly mixed.

This will mean that any sampling of a random finite sequence generated by this means, not matter how huge will only approach equal numerosity in an artifactual manner. Similar to and with the same quirks as when a "fair" coin is tossed a trillion times, the law of large numbers indicates it is likely that the ratio of heads to tails averages towards 50/50 though the discrepancy differential continues to increase.

Imagine a program that lists all possible finite sequences of 1−>9, up to some finite limit, say expansions of up to a million digits. The quantis randomizer (a PC card) shuffles these various finite length outputs. As if each finite sequence were represented on a deck of cards; the shuffle would sometimes intersperse or mix the cards in a tight interweave and at other times in clumps. This produces all truly random numbers with initial sequences of up to a million digits. But truly random numbers are infinitely long. This corresponds to an infinite deck of cards each with finite numbers printed on it which are randomly (quantis device) shuffled so that there is no particular pattern in the output because it is just one of all possible outputs (viewed at a finite length). It is as if after a shuffle of all the infinitude of cards, the cards are then turned face up to read in one long string; but since the string is infinitely long there is no finite last digit to be recognized and read.

AIT prohibits Pi from being considered truly random because it generated by a shorter input algorithm than the length of the output sequence it can produce. Calculating Pi can produce an unboundedly finite series of finitely incremented expansions of digits. The definition of algorithm says that it is a finite process. Knuth has introduced another term for algorithm which is defined like an algorithm but does not have the finitess requirement: "computational method" which describes unending computable numbers.

There isn't any way to input an infinitely long input X which is the only way to produce infintely long output Y. One can only input some finite segment of a random number which is assumed to be random because of the method by which it was obtained.

How does one know that the output is uniquely random by the AIT definition of randomness in respect to the input? Since the output is finite, it could coincide with say a million digits of Pi's expansion since all random numbers will include all finite sequences including the expansion of Pi to all the various finite lengths.

Suppose somewhere towards the end in a billion digit expansion of Pi we randomly select a 100 digits in that sequence. Suppose we now randomly shuffle those digits. Since we don't know for sure that Pi is random, we don't know if that reshuffled sequence of 100 digits is contained somewhere else in the expansion of Pi. In a random generation of digits those hundred digits could occur. How do we know that those hundred digits are not the result of a shorter algorithm (in the same category as Pi) which would violate the definition requiring the input string and the output string to be of the same length in order to qualify as AIT random?

Any truly random number (infinite) can be approximated to any given finite degree by an algorithm which can be used as input. One cannot distinguish if the reason a truly random number input cannot be generated by a shorter method of input is due to the fact that the random number is infinite or whether it is random. The paper says:

...."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."

There is no way of checking some finite input, even if you know that it was generated by random means such as a quantis device, and tell for sure there is no shorter algorithmic method of generating the corresponding output. The paper describes this:

"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"." ...

−−

Regarding "unstructured, law−less" which I think means that there is no rule. http://www.cis.udel.edu/~case/colt.html

John Case's COLT Page

"Consider the problem of finding a rule for generating a sequence of numbers such as 9, 61, 52, 63, 94, 46, 18, 1, 121, 441, ... Here is a rule for this sequence. First compute the squares of successive integers beginning with 3, but, then, to generate the sequence, use, in place of these squares, the squares each with its decimal digits written down in reverse order (ignoring any lead zeros). N.B. This rule can be written as a formal algorithm (or computer program). The problem of finding such rules gets harder as the sequences to generate get more complicated than the one above. Can the rule finding itself be done by some computer program? Interestingly, it is mathematically proven that there can be no computer program which can eventually find (synonym: learn) these (algorithmic) rules for all sequences which have such rules!"

So I don't see how one can use this definition below

"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"." ...

to determine what is "unstructured, law−less" since one can't tell what is ruleless (random) from that which has a rule, but the rule is too complex to be discovered. This latter situation is a lack of information which also generally describes the Heisenberg Uncertainty Principle (HUP). So from the paper: "Is there any relation between Heisenberg s uncertainty relation and Goedel s incompleteness?" ... "that the more precisely the position is determined, the less precisely the momentum is known in this instant, and vice versa." ...

"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 Schrodinger 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‘16 metres. Another view sees 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. [41]." ... [regarding the quantis device]

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

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 nth 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."

Regards,
Stephen

.