The complexity of proving chaoticity and the Church–Turing thesis

Cristian S. Calude,[1,a)] Elena Calude,[2,b)] and Karl Svozil[3,c)]
[1]*Department of Computer Science, University of Auckland, Private Bag 92019, 1142 Auckland, New Zealand*
[2]*Institute of Information and Mathematical Sciences, Massey University at Albany, Private Bag 102-904, North Shore, 0745 Auckland, New Zealand*
[3]*Institute for Theoretical Physics, Vienna University of Technology, Wiedner Hauptstrasse 8-10/136, 1040 Vienna, Austria*

(Received 16 June 2010; accepted 23 August 2010; published online 28 September 2010)

Proving the chaoticity of some dynamical systems is equivalent to solving the hardest problems in mathematics. Conversely, classical physical systems may "compute the hard or even the incomputable" by measuring observables which correspond to computationally hard or even incomputable problems. © *2010 American Institute of Physics*. [doi:10.1063/1.3489096]

The relationship between classical physical systems modeled by continua and their theory of computation is still not entirely settled. On the one hand, chaotic systems indicate that it might be "computationally hard" to formally prove many of their properties, in particular chaoticity. On the other hand, it might not be too unreasonable to speculate that such physical systems, due to their capabilities of utilizing the continuum, may be capable of performing tasks which are beyond the scope of universal Turing machines based on discrete entities.

I. INTRODUCTION

Proving that a dynamical system is chaotic is an important problem in chaos theory.[1] Despite causality,[2] *virtually any "interesting" question about nontrivial dynamical systems appears to be undecidable*,[3] but is there a way to mathematically prove *this statement*? Closely related is the question: *Is there a way to measure the difficulty of proving the chaoticity of a dynamical system?* There are only few "bridges" between chaotic dynamical systems and complexity theories, in particular algorithmic information theory.[4–7] The unpredictability of the systems studied in this article comes from a combination of chaoticity and a "decision problem" embedded in the system; the complexity of the decision problem (in the sense to be precisely described in Sec. II) may be arbitrarily large, including high incomputability. We shall show that "proving the chaoticity of some dynamical systems" amounts to "solving the hardest problems in mathematics" and vice versa.

We will study a class of mathematical sentences called Π_1-statements. A sentence of the form $\pi = \forall n$ Pred(n), where Pred is a computable predicate (n is always a nonnegative integer), is called a Π_1-*statement*. The Greek letter π is used as a generic notation for such a statement; it has no relation with the famous constant 3.145… . Clearly, π is true iff all instances of Pred, Pred(0), Pred(1), …, Pred(n), … are true. Every Π_1-statement is *finitely refutable* because a single false instance of Pred makes π false. For example, $\forall n[n^2+1>0]$ is true, but $\forall n[2n+3$ is prime$]$ is false.

We deal with formal proofs by using the Zermelo–Fraenkel set theory axiomatic system with the axiom of choice (ZFC), the standard system for doing mathematics. So, we say that "ZFC proves π" in case there is a proof in ZFC for π.

Da Costa and Doria[8] and Da Costa, Doria, and do Amaral[9] constructed a two-dimensional Hamiltonian system $\mathcal{H}$—a system of first-order differential equations which can be written in the form of Hamilton's equations, in which the Hamiltonian function represents the total energy of the system—with the property that (formally) proving the existence of a Smale horseshoe in $\mathcal{H}$ is equivalent to (formally) proving Fermat's last theorem. Contrary to the opinion expressed in the above articles, it was shown that proving that the two-dimensional Hamiltonian system $\mathcal{H}$ has a Smale horseshoe has low complexity[10] because Fermat's last theorem has a low complexity.

As Fermat's last theorem is a Π_1-statement, *it is natural to ask whether the above results can be extended to any* Π_1-*statement*. In this note we show that to every Π_1-statement π one can associate a dynamical system $\mathcal{H}_\pi$ such that proving in ZFC the chaoticity of $\mathcal{H}_\pi$ is equivalent to proving π in ZFC. By applying the computational method[11–13] to Π_1-statements we show that there are dynamical systems whose ZFC proofs of their chaoticity are arbitrarily complex and there are chaotic systems for which ZFC cannot prove their chaoticity. The techniques are related to (i) the construction of a Poincaré box as a classical physical random number generator (akin to a quantum Born box) and (ii) the conceivable capability of classical physical systems to "compute the hard or even the incomputable" by measuring observables which correspond to computationally hard or even incomputable problems.

a)Electronic mail: cristian@cs.auckland.ac.nz. URL: http://www.cs.auckland.ac.nz/~cristian.
b)Electronic mail: e.calude@massey.ac.nz. URL: http://www.massey.ac.nz/~ecalude.
c)Electronic mail: svozil@tuwien.ac.at. URL: http://tph.tuwien.ac.at/~svozil.

II. Π_1-STATEMENTS AND THE COMPLEXITY MEASURE

In this section we present a complexity measure[11–13] for Π_1-statements defined by means of register machine programs.

We use a fixed "universal formalism" for programs, more precisely, a universal self-delimiting Turing machine U. The machine U (which is fully described below) has to be *minimal* in the sense that none of its instructions can be simulated by a program for U written with the remaining instructions.

To every Π_1-statement $\pi=\forall m\ \mathrm{Pred}(m)$ we associate the algorithm $\Pi_{\mathrm{Pred}}=\inf\{n:\mathrm{Pred}(n)=\mathrm{false}\}$ which systematically searches for a counterexample for π. There are many programs (for U) which implement Π_{Pred}; without loss of generality, any such program will be denoted also by Π_{Pred}. Note that π is true iff $U(\Pi_{\mathrm{Pred}})$ never halts.

The complexity (with respect to U) of a Π_1-statement π is defined by the length of the smallest-length program (for U) Π_{Pred}—defined as above—where minimization is calculated for all possible representations of π as $\pi=\forall n\mathrm{Pred}(n)$: $C_U(\pi)=\min\{|\Pi_{\mathrm{Pred}}|:\pi=\forall n\mathrm{Pred}(n)\}$.

For C_U it is irrelevant whether π is known to be true or false. In particular, the program containing the single instruction halt is not a Π_{Pred} program for any Pred. As the exact value of C_U is not important (C_U is incomputable), following a previous article by two of the authors[13] we classify Π_1-statements into the following classes: $\mathfrak{C}_{U,n}=\{\pi:\pi$ is a Π_1-statement, $C_U(\pi)\leq n$ kbit$\}$. [Recall that a kilobit (kbit) is equal to 2^{10} bits.]

We briefly describe the syntax and the semantics of a register machine language which implements a (natural) minimal universal prefix-free binary Turing machine U. Any register program (machine) uses a finite number of registers, each of which may contain an arbitrarily large non-negative integer. By default, all registers, named with a string of lower or upper case letters, are initialized to 0. Instructions are labeled by default with $0,1,2,\ldots$.

The register machine instructions are listed below. Note that in all cases R2 and R3 denote either a register or a non-negative integer, while R1 must be a register. When referring to R we use, depending upon the context, either the name of register R or the non-negative integer stored in R.

=R1,R2,R3: if the contents of R1 and R2 are equal, then the execution continues at the R3th instruction of the program; if the contents of R1 and R2 are not equal, then execution continues with the next instruction in sequence, and, if the content of R3 is outside the scope of the program, then we have an illegal branch error.

&R1,R2: the contents of register R1 is replaced by R2.

+R1,R2: the contents of register R1 is replaced by the sum of the contents of R1 and R2.

!R1: one bit is read into the register R1, so the contents of R1 becomes either 0 or 1; any attempt to read past the last data-bit results in a run-time error.

%: this is the last instruction for each register machine program before the input data; it halts the execution in two possible states: either successfully halts or it halts with an under-read error.

TABLE I. Binary encoding of special characters (instructions and comma); ε is the empty string.

Special characters code		Instruction code	
,	ε	+	111
&	01	!	110
=	00	%	100

A *register machine program* consists of a finite list of labeled instructions from the above list, with the restriction that the halt instruction appears only once, as the last instruction of the list.

To compute an upper bound on the complexity of a Π_1-statement π we need to compute the size in bits of the program Π_π, so we need to uniquely code in binary the programs for U. To this aim we use a prefix-free coding as follows.

Table I enumerates the binary coding of special characters. For registers we use the prefix-free regular code, $\mathrm{code}_1=\{0^{|x|}1x|x\in\{0,1\}^*\}$. The register names are chosen to optimize the length of the program, i.e., the most frequent registers have the smallest code_1 length.

For non-negative integers we use the prefix-free regular code, $\mathrm{code}_2=\{1^{|x|}0x|x\in\{0,1\}^*\}$. The instructions are coded by self-delimiting binary strings as follows (see more details in Refs. 11–13):

(i) `&R1,R2` is coded in two different ways, depending on R2 (we omit ε): $01\mathrm{code}_1(\mathrm{R1})\mathrm{code}_i(\mathrm{R2})$, where $i=1$ if R2 is a register and $i=2$ if R2 is a non-negative integer.

(ii) `+R1,R2` is coded in two different ways depending on R2: $111\mathrm{code}_1(\mathrm{R1})\mathrm{code}_i(\mathrm{R2})$, where $i=1$ if R2 is a register and $i=2$ if R2 is a non-negative integer.

(iii) `=R1,R2,R3` is coded in four different ways depending on the data types of R2 and R3: $00\mathrm{code}_1(\mathrm{R1})\mathrm{code}_i(\mathrm{R2})\mathrm{code}_j(\mathrm{R3})$, where $i=1$ if R2 is a register and $i=2$ if R2 is a non-negative integer, $j=1$ if R3 is a register and $j=2$ if R3 is a non-negative integer.

(iv) `!R1` is coded by $110\mathrm{code}_1(\mathrm{R1})$.

(v) `%` is coded by 100.

For example, Goldbach's conjecture (included in Hilbert's eighth problem[14]) states that *all positive even integers greater than two can be expressed as the sum of two primes.* The program Π_{Goldbach} listed in Table II gives the upper bound $C_U(\mathrm{Goldbach})\leq 540$ which proves that the Goldbach conjecture is in the lowest class $\mathfrak{C}_{U,1}$.

III. MAIN RESULTS

We start with a result relating Π_1-statements and Hamiltonians.

Theorem 1: *Assume ZFC is arithmetically sound, i.e., every statement ZFC proves is true. Then, to each* Π_1*-statement* $\pi=\forall m\mathrm{Pred}(m)$ *one can effectively construct in the formal language of ZFC a Hamiltonian system* $\mathcal{H}_\pi$

TABLE II. Program Π_{Goldbach} for the Goldbach conjecture.

```
00:    = a a 16
01:    & e 2
02:    & d 1
03:    = a e c
04:    & d 0
05:    & f e
06:    = f a 13
07:    +f 1
08:    +d 1
09:    = d e 11
10:    = a a 6
11:    & d 0
12:    = a a 6
13:    = d 0 c
14:    + e 1
15:    = a a 2
16:    & g 4
17:    & h 2
18:    = g h 38
19:    & c 22
20:    & a h
21:    = a a 1
22:    = d 0 35
23:    & i 0
24:    & k h
25:    = k g 29
26:    +i 1
27:    +k 1
28:    = a a 25
29:    & c 32
30:    & a i
31:    = a a 1
32:    = d 0 35
33:    +g 2
34:    = a a 17
35:    +h 1
36:    = a a 18
37:    & d 0
38:    %
```

such that ZFC proves that the system $\mathcal{H}_\pi$ has a Smale horseshoe iff ZFC proves π.

We denote by h and k the Hamiltonian for the two-dimensional system with a Smale horseshoe as defined by Holmes and Marsden[15] (their example 4) and the Hamiltonian for the free particle, respectively. Clearly, the systems h and k can be represented in the formal language of ZFC. Define the Hamiltonian $\mathcal{H}_\pi^m$ as a linear combination of h,k,

$$\mathcal{H}_\pi^m(q_1,\ldots,q_n,p_1,\ldots,p_n) = \text{Pred}(m)\cdot h(q_1,\ldots,q_n,p_1,\ldots,p_n) + (1-\text{Pred}(m)) \cdot k(q_1,\ldots,q_n,p_1,\ldots,p_n). \quad (1)$$

Fix a positive integer i. In view of Eq. (1), $\mathcal{H}_\pi^i$ can be represented in the formal language of ZFC and $\mathcal{H}_\pi^i(q_1,\ldots,q_n,p_1,\ldots,p_n)=h(q_1,\ldots,q_n,p_1,\ldots,p_n)$ iff ZFC proves π. In case the above equivalence holds true, $\mathcal{H}_\pi^i(q_1,\ldots,q_n,p_1,\ldots,p_n)=\mathcal{H}_\pi^j(q_1,\ldots,q_n,p_1,\ldots,p_n)$, for all non-negative integers i,j, hence we can name each $\mathcal{H}_\pi^t$ by $\mathcal{H}_\pi$.

We have shown that

> ZFC proves π iff ZFC proves that $\mathcal{H}_\pi$ has a Smale horseshow,

hence ending the proof of Theorem 1.

If π is true but unprovable in ZFC, then the equality $\mathcal{H}_\pi^i(q_1,\ldots,q_n,p_1,\ldots,p_n)=h(q_1,\ldots,q_n,p_1,\ldots,p_n)$ is true but unprovable in ZFC.

In case π is the Fermat's last theorem, Theorem 1 is exactly the result proved;[8,9] our direct proof does not need the machinery involving Richardson lemma used in Refs. 8 and 9.

Theorem 1 can be applied to a variety of Π_1-statements including Goldbach's conjecture, Riemann's hypothesis, the four color theorem, and many others.

We address now the complexity issue: How difficult is it to prove in ZFC that the system $\mathcal{H}_\pi$ in Eq. (1) is chaotic? Using the complexity C_U we can show that Fermat's last theorem and Goldbach's conjecture are in $\mathfrak{C}_{U,1}$, the Riemann hypothesis is in $\mathfrak{C}_{U,3}$, and the four color theorem is in $\mathfrak{C}_{U,4}$;[13,16,17] their corresponding dynamical system produced by Theorem 1 has the property that the complexity of its chaoticity proof is in the corresponding class.

As for every natural n there exists a natural m_n such that $\mathfrak{C}_{U,n}\subset\mathfrak{C}_{U,m_n}$. It follows that, according to C_U, there exist arbitrarily complex Π_1-statements; hence proving the chaoticity of the system $\mathcal{H}_\pi$ can be arbitrarily complex.

Finally, there are infinitely many true, but unprovable in ZFC, Π_1-statements π,[18] such that the corresponding systems $\mathcal{H}_\pi^i$ are chaotic but ZFC cannot prove their chaoticity. For example, from the negation of the halting problem for U we get infinitely many Π_1-statements π_x= "$\forall n$ [$U(x)$ does not stop in time n]" which are undecidable in ZFC.

IV. COMPUTATIONAL CAPABILITIES OF CHAOTIC MOTION

One of the intriguing possibilities of the aforementioned equivalences between certain statements in ZFC and chaotic motion is the hypothetical possibility to "decide" hard problems in ZFC or "perform incomputable tasks" by observing the corresponding chaos.[3,9,19–21] Indeed, if such methods and procedures have an "effective" physical implementation, then, strictly speaking, the Church–Turing thesis identifying the informal notion of *computable algorithm* with *Turing computability*, or, equivalently, *recursive functions*, is too restricted and has to be adapted to the physical capacities[22–24] (for a converse viewpoint restricting operations to strictly finitistic means, see Refs. 25–27).

It is rather intriguing that, at least in this respect, the situation resembles the famous Einstein, Podolski, and Rosen (EPR) argument[28] for a possible "incompleteness" of quantum mechanics. According to EPR, whereas quantum theory does not allow complementary physical observables to simultaneously "exist," experiment (augmented with counterfactual reasoning) allows for such "elements of physical reality."

In the case of chaotic systems, our present theory of computability, formalized by recursion theory, does not allow the "execution" of certain "hard" tasks; but the equivalent chaotic systems would perform just such tasks, sometimes with relative ease on the side of the experimenter. One example of such seemingly mismatch—in the sense of EPR—of computability theory and physical computation is the construction of "oracles producing random bits," as discussed in Sec. V.

V. POINCARÉ BOX AS PHYSICAL RANDOM NUMBER GENERATOR

Chaotic systems can be used as a physical device for incomputability. In the "extreme" algorithmically incompressible case, a chaotic dynamical system can serve as a source (oracle) of random bits; i.e., as a physical *random number generator* (RNG). This RNG can be conceptualized by enclosing a chaotic system in a "black box" with an output interface which communicates the consecutive physical states of the chaotic evolution[29] in a properly encoded symbolic form. In order for these, say, strings of bits, to be physically certified random, it is necessary to ascertain chaoticity; a property which relates to the proofs of chaoticity discussed above.

This scenario can be elucidated by considering the shift map σ (a form of generalized shift studied by Moore[6]) which "pushes" up successive bits of the sequence $s=0.s_1s_2s_3\cdots$; i.e., $\sigma(s)=0.s_2s_3s_4\cdots$, $\sigma(\sigma(s))=0.s_3s_4s_5\cdots$, and so on. Suppose one starts with an initial "measurement" precision of, say, just one bit after the dot, indicated by a "window of measurability;" all other information "beyond the first bit after the comma" is hidden to the experimenter at this point. Consider an initial state represented by an algorithmically random real s. At first the experimenter records the first position s_1 of s, symbolized by $0.[[s_1]]s_2s_3\cdots$, where the square brackets $[[\cdots]]$ indicate the boundaries of the experimenter's sliding window of measurability. Successive iterations of the shift map "bring up" more and more bits of the initial sequence of s; i.e., $\sigma(s)$ yields $0.s_1[[s_2]]s_3s_4\cdots$, $\sigma(\sigma(s))$ yields $0.s_1s_2[[s_3]]s_4s_5\cdots$, and in general $\sigma^{(i)}(s)$ yields $0.\cdots s_{i-1}s_i[[s_{i+1}]]s_{i+2}s_{i+3}\cdots$ after i iterations of the shift map. Thus effectively, the algorithmic information content of s "unfolds" at a rate of 1 bit per time cycle. If s is algorithmically random, then (at least ideally) the empirical recording of its successive bits generates a random sequence (in the asymptotic limit).

It is not totally unreasonable to conjecture that, with respect to algorithmic (hence also statistical) tests of randomness, *Poincaré boxes cannot* be differentiated from another type of physical RNGs termed *Born boxes*, which are based on quantum indeterminism (e.g., photons impinging on beam splitters and detectors[30–37]). Considering the different physical origins of physical indeterminism exploited by the Poincaré and Born boxes—in the first, classical case, indeterminism resides in the continuum, whereas in the second, quantum case, in the postulated[38–41] irreducible randomness of certain individual outcomes involving photons—why should the two physical RNGs perform equally from an algorithmic information theoretic[42,43] point of view? Because, one could argue, both would produce (in the asymptotic regime) random strings with high probability.

The Poincaré box derives its random behavior from a *single, individual* initial value containing incompressible algorithmic information with probability one,[4,5] whereas the Born box utilizes *successive, independent* ideal coin tosses. Whether or not these speculations are justified or not, only experiment can tell. So far, no empirical evidence either for or against the conjectured equivalence of Poincaré and Born boxes exists.

It is not too difficult to "construct" a Poincaré box by utilizing a shift map which "pumps" up the bits of the binary representation of the initial value by 1 bit per (discrete iteration) cycle. Of course, assuring the physical representability of this extreme chaotic regime for concrete classical chaotic systems might turn out to be a hard task, as has been argued above. With this proviso, and by further assuming that the initial value is some element of the continuum (in ZFC the "selection" of an initial value is guaranteed by the axiom of choice), the shift map is, at least asymptotically, capable of yielding a random number with probability one.

VI. SUMMARY AND OUTLOOK

We have argued that every Π_1-statement π can be associated with a dynamical system $\mathcal{H}_\pi$ such that ZFC proves the chaoticity of $\mathcal{H}_\pi$ iff ZFC proves π. Many hard problems, such as, for example, the Riemann hypothesis and the four color theorem, are Π_1-statements. The computational method[11–13] has been applied to Π_1-statements, resulting in a complexity measure for proving the chaoticity of some dynamical systems. Consequently, there are dynamical systems for which the ZFC proofs of their chaoticity are arbitrarily complex according to the above complexity measure. Furthermore, there are infinitely many chaotic systems for which ZFC cannot prove their chaoticity.

One of the challenging conceptual questions which are motivated by these results is the issue of relating physical entities to formal ones. In particular at stake is the Church–Turing thesis, which is challenged from a classical physical perspective. As classical chaotic motion seems to be capable to "perform" incomputable tasks—a criterion which might, as we argue, be hard to certify for a wide variety of Hamiltonian systems, but which nevertheless is a feasible scenario—it might not be too unreasonable to speculate that the present formal theories of computability would have to be adapted in accordance with our physical capabilities originating from chaotic motion.

ACKNOWLEDGMENTS

We thank Alastair Abbott and the anonymous referees for critical comments which improved the article.

[1] M. Hirsch, *Chaos, Fractals, and Dynamics*, Lecture Notes in Pure and Applied Mathematics Vol. 98, edited by P. Fischer and W. R. Smith (Marcel Dekker, New York, 1985).
[2] P. Suppes, Midwest Stud. Philos. **18**, 242 (1993).
[3] I. Stewart, Nature (London) **352**, 664 (1991).
[4] A. A. Brudno, Trans. Mosc. Math. Soc. **44**, 127 (1983).
[5] J. P. Crutchfield and N. H. Packard, Int. J. Theor. Phys. **21**, 433 (1982).

[6]C. D. Moore, Phys. Rev. Lett. **64**, 2354 (1990); cf. Ch. Bennett, Nature (London) **346**, 606 (1990).
[7]P. Gács, M. Hoyrup, and C. Rojas, in *26th International Symposium on Theoretical Aspects of Computer Science-STACS 2009*, edited by S. Albers and J.-Y. Marion (IBFI Schloss Dagstuhl, Freiburg, 2009), pp. 469–480; e-print arXiv:cond-mat/0902.1939.
[8]N. C. A. Da Costa and F. A. Doria, Int. J. Theor. Phys. **30**, 1041 (1991).
[9]N. C. A. Da Costa, F. A. Doria, and A. F. F. do Amaral, Int. J. Theor. Phys. **32**, 2187 (1993).
[10]E. Calude, "Fermat's last theorem and chaoticity," CDMTCS Research Report 383, 2010.
[11]C. S. Calude, E. Calude, and M. J. Dinneen, J. Multiple-Valued Logic and Soft Comput. **12**, 285 (2006); CDMTCS Research Report No. 277, 2006.
[12]C. S. Calude and E. Calude, Complex Systems **18**, 267 (2009); CDMTCS Research Report No. 343, 2008.
[13]C. S. Calude and E. Calude, Complex Systems **18**, 387 (2010); CDMTCS Research Report No. 369, 2009.
[14]D. Hilbert, Bull. Am. Math. Soc. **8**, 437 (1902).
[15]P. J. Holmes and J. E. Marsden, Commun. Math. Phys. **82**, 523 (1982).
[16]C. S. Calude and E. Calude, "The complexity of the four colour theorem," LMS Journal of Computation and Mathematics (2010); CDMTCS Research Report No. 368, 2009.
[17]E. Calude, "The complexity of the Goldbach's conjecture and Riemann's hypothesis," CDMTCS Research Report 370, 2009.
[18]C. Calude and G. Păun, RAIRO Inform. Theor. Appl. **17**, 49 (1983).
[19]B. Scarpellini, Z. Math. Logik Grundlagen Math. **9**, 265 (1963).
[20]B. Scarpellini, Minds and Machines **13**, 49 (2003).
[21]B. Scarpellini, Minds and Machines **13**, 79 (2003).
[22]M. Davis, *Computability and Unsolvability* (McGraw-Hill, New York, 1958).
[23]H. Rogers, Jr., *Theory of Recursive Functions and Effective Computability* (MacGraw-Hill, New York, 1967).
[24]G. Kreisel, Synthese **29**, 11 (1974).
[25]P. W. Bridgman, Scripta Mathematica **2**, 101 (1934); **2**, 224 (1934), cf. R. Landauer (Ref. 44).
[26]R. O. Gandy, in *The Kleene Symposium*, Studies in Logic and Foundations of Mathematics Vol. 101, edited by J. Barwise, H. J. Kreisler, and K. Kunen (North-Holland, Amsterdam, 1980), pp. 123–148.
[27]R. O. Gandy, in *Logic Colloquium '82*, edited by D. van Dalen, D. Lascar, and J. Smiley (North-Holland, Amsterdam, 1982), pp. 129–146.
[28]A. Einstein, B. Podolsky, and N. Rosen, Phys. Rev. **47**, 777 (1935).
[29]N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, Phys. Rev. Lett. **45**, 712 (1980).
[30]K. Svozil, Phys. Lett. A **143**, 433 (1990).
[31]J. G. Rarity, M. P. C. Owens, and P. R. Tapster, J. Mod. Opt. **41**, 2435 (1994).
[32]T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, and A. Zeilinger, Rev. Sci. Instrum. **71**, 1675 (2000); e-print arXiv:quant-ph/9912118.
[33]A. Stefanov, N. Gisin, O. Guinnard, L. Guinnard, and H. Zbinden, J. Mod. Opt. **47**, 595 (2000).
[34]M. Hai-Qiang, W. Su-Mei, Z. Da, C. Jun-Tao, J. Ling-Ling, H. Yan-Xue, and W. Ling-An, Chin. Phys. Lett. **21**, 1961 (2004).
[35]P. X. Wang, G. L. Long, and Y. S. Li, J. Appl. Phys. **100**, 056107 (2006).
[36]M. Fiorentino, C. Santori, S. M. Spillane, R. G. Beausoleil, and W. J. Munro, Phys. Rev. A **75**, 032334 (2007).
[37]K. Svozil, Phys. Rev. A **79**, 054306 (2009); e-print arXiv:quant-ph/0903.2744.
[38]M. Born, Z. Phys. **37**, 863 (1926).
[39]M. Born, Z. Phys. **38**, 803 (1926).
[40]A. Zeilinger, Nature (London) **438**, 743 (2005).
[41]C. S. Calude and K. Svozil, Advanced Science Letters **1**, 165 (2008); e-print arXiv:quant-ph/0611029.
[42]G. J. Chaitin, *Algorithmic Information Theory* (Cambridge University Press, Cambridge, 1987).
[43]C. Calude, *Information and Randomness—An Algorithmic Perspective*, 2nd ed. (Springer, Berlin, 2002).
[44]R. Landauer, in *On Limits*, edited by J. L. Casti and J. F. Traub (Santa Fe Institute, Santa Fe, 1994), p. 39; Santa Fe Institute Report No. 94-10-056, 1994.