How to Acknowledge Hypercomputation?

Alexander Leitsch*
Günter Schachner
Institut für Computersprachen
Vienna University of Technology
Favoritenstr.9/185, 1040 Vienna, Austria
**leitsch@logic.at*

Karl Svozil†
Institute for Theoretical Physics,
Vienna University of Technology,
Wiedner Hauptstraße 8-10/136, 1040 Vienna, Austria
†*svozil@tuwien.ac.at*

We discuss the question of how to operationally validate whether or not a "hypercomputer" performs better than the known discrete computational models.

1. Introduction

It is widely acknowledged [1, 2], that every physical system corresponds to a computational process, and that every computational process, if applicable, has to be physically and operationally feasible in some concrete realization. In this sense, the physical and computational capacities should match, because if one is lagging behind the other, there is a lack in the formalism and its potential scientific (and ultimately, technological) applicability. Therefore, the exact correspondence of the mathematical formalism on the one hand, and the particular physical system that is represented by that formalism on the other hand, demand careful attention.

If one insists on operationalizability [3], one need not go very far in the history of mathematics to encounter suspicious mathematical objects. Surely enough, the number π can be defined and effectively computed as the ratio of the circumference to the diameter of a "perfect" (platonic) circle. Likewise, the numbers $\sqrt{2}$ and $\sqrt{3}$ can be interpreted as the ratio between the length of the diagonal to the side length of any square and cube, respectively. But it is not totally unjustified to ask whether or not these numbers have any operational meaning in a strict physical sense; that is, whether such numbers could, at least in principle, be constructed and measured with arbitrary, or even absolute, precision.

At the heart of most of the problems seems to lie the ancient issue of the "very large/small" or even potential infinite versus the actual

https://doi.org/10.25088/ComplexSystems.18.1.131

infinite. Whereas the mathematical formalism postulates the existence of actual infinite constructions and methods (such as the summation of a [convergent] geometric series, or diagonalization) the physical processes, methods, and techniques are never infinite.

Suppose, as an example, one would attempt the operationalization of π. Any construction of a "real" circle and one of its diameters, and a subsequent measurement thereof, would find its natural scale bound from below by the atomistic structure of matter upon which any such circle is based. Long before those molecular or atomic scales, the physical geometry might turn out to be not as straightforward as it appears from larger scales; for example, the object might turn out to be a fractal.

Chaitin's omega number [4], which is interpretable as the halting probability of a universal computer, can be "computed in the limit" (without any computable radius of convergence) by a finite-size program in infinite time and with infinite space. Just as for π—the difference being the absence of any computable radius of convergence—the first digits of omega are well known [5], yet omega has been proved to be algorithmically incompressible and thus random. Nevertheless, presently, for all practical purposes, the statement that "the $10^{10^{10^{10}}}$ digit in a decimal expansion of π is 5" is as unverifiable as a similar statement for omega. Omega encodes all decision problems which can be algorithmically interpreted. For instance, for a particular universal computer, Goldbach's conjecture and Riemann's hypothesis could be decided with programs of size 3484 and 7780 bits, respectively [6]. Yet, omega appears to have two features which are normally considered contradictory: it is one of the most informative mathematical numbers imaginable, yet at the same time this information is so compressed that it cannot be deciphered. Thus omega appears to be totally structureless and random. In this sense, for omega, total information and total randomness seem to be "two sides of the same coin". On a more pragmatic level, it seems impossible here to differentiate between order and chaos, or between knowledge and chance. This gives a taste of what can be expected from any "hypercomputation" beyond universal computability as defined by Turing.

It should always be kept in mind that all our sense perceptions are derived from elementary discrete events, such as clicks in photon or particle detectors, even if they appear to be analog: the apparently smooth behavior has a discrete fine structure. Among other issues, such as finiteness of system resources, this discreteness seems to prohibit the "physical realization" of any actual infinities.

What is the physical meaning of infinite concepts, such as spacetime singularities, point particles, or infinite precision? For instance, are infinity machines with geometrically squeezed time cycles, such as the ones envisioned by Weyl [7] and others [8-18], physically feasible? Motivated by recent proposals to utilize quantum computation for trespassing the Turing barrier [19-22], these accelerating Turing

https://doi.org/10.25088/ComplexSystems.18.1.131

machines have been intensively discussed [23] among other forms of hypercomputation [24-26].

Certainly, the almost ruthless and consequential application of seem ingly mind-boggling theories such as quantum mechanics, as far as finitistic methods are concerned, has yielded one victory after another. But it should be kept in mind that the use of actual transfinite concepts and methods remains highly conjectural.

A priori, while it may appear rash to exclude the transfinite in general, and transfinite set theory in particular, from physics proper, one should be aware of its counterintuitive consequences (such as, for instance, the Banach-Tarski paradox) and be careful in claiming its physical applicability. Recall the old phrase attributed to Einstein and Infeld [27, p. 31]: "Physical concepts are free creations of the human mind, and are not, however it may seem, uniquely determined by the external world."

To this point, we are not aware of any test, let alone any application, of the actual transfinite in nature. While general contemplations about hypercomputations and the applicability of transfinite concepts for physics may appear philosophically interesting, our main concern will be operational *testability*: if presented with claims that hypercomputers exist, how could we possibly falsify, or even verify and test, such propositions [28]?

In what follows, hypercomputation will be conceptualized in terms of a black box with its input/output behavior. Several tests and their rather limited scope will be evaluated. Already in 1958, Davis [29, p. 11] sets the stage of the following discussion by pointing out "…how can we ever exclude the possibility of our being presented, some day (perhaps by some extraterrestrial visitors), with a (perhaps extremely complex) device or 'oracle' that 'computes' a noncomputable function?" While this may have been a remote, amusing issue in the days written, the advancement of physical theory in the past decades has made necessary a careful evaluation of the possibilities and options for verification and falsification of certain claims that a concrete physical system "computes" a noncomputable function.

2. On Black Boxes That Are Hypercomputers

In what follows we shall consider a device, an agent, or an oracle that one knows nothing about, and for which there is no rational understanding (in the traditional algorithmic sense) of its intrinsic working. This device may, for the sake of further discussion, be presented to us as an alleged "hypercomputer".

https://doi.org/10.25088/ComplexSystems.18.1.131

The following notation is introduced. Let B be a subset of $X_1 \times \cdots \times X_m$. The i^{th} projection of B (for $i = 1, \ldots, m$), written as B_i, is defined by:

$$B_i = \{x \mid x \in X_i, \\ (\exists\, y \in X_1 \times \cdots \times X_{i-1})(\exists\, z \in X_{i+1} \times \cdots \times X_m)(y, x, z) \in B\}.$$

For any $x \in N^m$ we define

$$|x| = \max\{x_i \mid i \in \{1, \ldots, m\}\}.$$

Then, a hypercomputer can be defined via its input/output behavior of black boxes as follows.

Definition 1 (black box). Let X, Y be sets and $\mathbb{N}$ be the set of natural numbers. A subset B of $X \times Y \times N$ is called a *black box* if $B_1 = X$. X is called the *input set* and Y the *output set*.

Note that the condition $B_1 = X$ models a computing device that is total; that is, an output always exists.

Definition 2. Let B be a black box. We define

$$f_B = \{(x, y) \mid (\exists\, z)(x, y, z) \in B\},$$
$$t_B = \{(x, z) \mid (\exists\, y)(x, y, z) \in B\}.$$

f_B is called the *input/output relation* of B and t_B the *computing time* of B. If f_B and t_B are functions, then B is called deterministic.

Every halting deterministic Turing machine defines a black box. Indeed, let M be a Turing machine (computing a total function), $f_M : X \to Y$ be the function computed by M, and t_M be the computing time of M. Then

$$\{(x, f_M(x), t_M(x)) \mid x \in X\}$$

is a (deterministic) black box. Similarly, all halting nondeterministic Turing machines define black boxes.

Definition 3 (hypercomputer). A *strong hypercomputer* is a black box B where f_B is not Turing-computable.

Definition 4. Let C be a class of computable monotone functions $\mathbb{N} \to \mathbb{N}$ containing the polynomials (over $\mathbb{N}$ with non-negative coefficients). Then C is called a *bound class*.

Definition 5. A *weak hypercomputer* is a black box B with the following property: There exists a bound class C such that

- $t_M(x) > g(|x|)$ almost everywhere for all $g \in C$ and for all Turing machines M with $f_M = f_B$.
- There exists an $h \in C$ such that $t_B(x) \leq h(|x|)$ for all $x \in B_1$.

A strong hypercomputer computes either a noncomputable function or decides an undecidable problem. A weak hypercomputation

https://doi.org/10.25088/ComplexSystems.18.1.131

outperforms all Turing machines. A possible scenario for a weak hypercomputer B is the following: f_B is an EXPTIME-complete problem; therefore, there exists no polynomial p and no Turing machine M computing f_B with $t_M(x) \leq p(|x|)$ for all $x \in X$, but $t_B(x) \leq p(|x|)$ for all $x \in X$ and for a polynomial p.

For nondeterministic hypercomputers we may distinguish between the following cases:

- f_B is not a function,
- f_B is a function, but t_B is not.

For stochastic hypercomputers, either t_B or both f_B and t_B are random variables, and the requirements on the computation have to be specified.

3. Tests

Having set the stage for a general investigation into hypercomputers that are presented to us as black boxes, we shall consider a few cases and tests. These test methods will be essentially heuristic and present no way of systematically addressing the issue of falsifying or even verifying hypercomputation.

One strategy for creating tests will be to consider problems that are *asymmetric* with respect to their *creation* and *verification*—which should be "easy"—on the one hand, and their *solution*—which should be "hard"—on the other hand.

3.1 NP-Complete Cases

It may be conjectured that, by operational means, it is not possible to go beyond tests of hyper-NP-completeness. Even for an NP-complete problem—for instance, the satisfiability problem of propositional logic (SAT)—it is not trivial to verify that a hypercomputer solves the problem in polynomial time. Without insight into the internal structure of the hypercomputer, we cannot obtain a proof of polynomial time computation, which is an asymptotic result. Even here we rely on experiments to test a "large" number of problems. A central problem consists in the right selection of problem sequences. If the selection is based on random generators, we merely obtain results on average complexity, which would not be significant.

Furthermore, we need at least some information about the polynomial in question (e.g., its maximum degree). Otherwise it remains impossible to decide by finite means whether some behavior is polynomial or not.

3.2 Harder Cases with Tractable Verification

Do there exist (decision) problems that are harder than the known NP-complete cases, possibly having no recursively enumerable solu-

https://doi.org/10.25088/ComplexSystems.18.1.131

tion and proof methods, whose results nevertheless are tractable verifiable? For example, the problem of graph nonisomorphism (GNI) is one that is not known to be in NP, not even in NP $\bigcup$ BPP. Nevertheless, it is possible to "efficiently verify" whether a "prover" solves this problem correctly.

If the prover claims that two graphs G_1 and G_2 are isomorphic, he can convince us by providing a graph isomorphism. That can be checked in polynomial time, which also means that GNI $\in$ coNP. If, on the other hand, the prover claims that G_1 and G_2 are nonisomorphic, we can verify this by the following *interactive proof*.

1. Choose one of the graphs G_1 and G_2 with equal probability.
2. Apply an arbitrary permutation to its vertices; this yields graph H.
3. The prover must decide whether H is equivalent to G_1 or G_2.
4. Repeat for N rounds.

If the initial answer was wrong and the graphs G_1 and G_2 are actually isomorphic, the prover can in step 3 only guess which graph was chosen in step 3 (since now H could have been derived from either). Hence, after N rounds we can be sure with probability $1 - 2^{-N}$ that the graphs G_1 and G_2 are nonisomorphic.

By denoting the class of interactive proofs by IP, we have shown that GNI $\in$ IP. Interactive proofs further exist for every language in PSPACE (which is assumed to be much larger than NP). In fact, it can be shown [30] that IP equals PSPACE. This means, in particular, that IP is closed under complement.

The protocol in the example given has the property that in each round a constant number of messages is sent. In a generic interactive proof system for PSPACE this is not necessarily true; but at any instance the number of messages depends polynomially on the input length.

In the literature, specific classes of interactive proof systems are investigated as well, for example, the Arthur-Merlin class [31] and the Goldwasser-Micali-Rackoff (GMR) class [32]. The former uses public coin tosses, with the intention of accommodating certain languages in the lowest complexity class possible. The latter uses private coin tosses, with the intention of covering the widest possible class of efficiently verifiable languages; additionally, it has the feature of providing *zero-knowledge proofs,* which is of great significance in cryptography. (The protocol we presented does not have the zero-knowledge property, unless GNI $\in$ BPP, but can be modified to have it.) For further information on interactive proof systems see [33, 34].

3.3 Inference of Problems

One may confront the hypercomputer with the problem of comparing the solutions of multiple tasks. Such a comparison need not necessar-

https://doi.org/10.25088/ComplexSystems.18.1.131

ily involve the separate computation of the solutions of these multiple tasks.

As an analogy, consider Deutsch's problem as one of the first problems that quantum computers could solve effectively. Consider a function that takes a single (classical) bit into a single (classical) bit. There are four such functions f_1, ..., f_4, corresponding to all variations. One can specify or "prepare" a function bitwise, or alternatively, one may specify it by requiring that, for instance, such a function acquires different values on different inputs, such as $f(0) \neq f(1)$. Thereby, we may, even in principle, learn nothing about the individual functional values alone.

3.4 Generation of Random Sequences

By implementing Chaitin's "algorithm" to compute Chaitin's omega [35] or variants thereof [36], it would in principle be possible to "compute" the first bits of random sequences. Such random sequences could, in principle, be subject to the usual tests of stochasticity [37, 38].

Note that in quantum mechanics, the randomness of certain microphysical events, such as the spontaneous decay of excited quantum states [39, 40], or the quantum coin toss experiments in complete context mismatches [37], is postulated as an axiom. This postulate is then used as the basis for the production of quantum randomness oracles such as the commercially available *Quantis* interface [41].

4. Impossibility of Unsolvable Problems Whose Solution Is Polynomially Verifiable

Let Σ_0 be a finite (nonempty) alphabet and $X \subset \Sigma_0^*$ be a semidecidable, but not decidable, set. That means there exists a Turing machine that accepts the language X, but does not terminate on all $x \in \Sigma_0^*$. The concept of "acceptable by Turing machines" is equivalent to "derivable by inference systems" or "producible by grammars". We choose the approach of a *universal proof system*, that is, of a system which simulates every Turing machine.

Let P be such a proof system. Let V be an infinite set of variables (over strings in Σ^*). A *metastring* is an object $x_1 \ldots x_n$ where $x_i \in \Sigma$ or $x_i \in V$. If X is a metastring and θ is a substitution (i.e., a mapping $V \to (V \bigcup \Sigma)^*$), then $X\theta$ is called an *instance* of X. If $X\theta \in \Sigma^*$, we call $X\theta$ a *ground instance* of X.

https://doi.org/10.25088/ComplexSystems.18.1.131

We may define $P = (\mathcal{Y}, \mathcal{X}, \Sigma, \Sigma_0)$, where $\mathcal{Y}$ is a finite set of metastrings (the axioms) and $\mathcal{X}$ is a finite set of *rules*, that is, expressions of the form

$$\frac{X_1 \dots X_n}{X}$$

where $X_1, \dots, X_n, X$ are metastrings such that the set of variables in X is contained in the set of variables in $X_1, \dots, X_n$.

Σ_0 is a (nonempty) subset of Σ (defining the strings of the theory to be generated).

A *derivation* φ *in* P is a tree such that all nodes are labeled by strings in Σ^*. In particular:

- The leaves of φ are labeled by ground instances of axioms.
- Let N be a node in φ which is not a leaf and $(N, N_1), \dots, (N, N_k)$ be the nodes from N, then $N_1 \dots N_k / N$ is a ground instance of a rule in X.

A proof of an x in Σ_0^* in P is a derivation in P with the root node labeled by x. We call x *provable* in P if there exists a proof of x in P.

Fact: As Σ is finite there are only finitely many derivations of length less than or equal to k for any natural number k, where *length* is the number of symbol occurrences. Let $P[k]$ be the set of all derivations of length less than or equal to k.

We prove now that there is no recursive function g such that for all $x \in X$:

> $(*)$ x is provable in P if and only if there exists a proof φ of x with $|\varphi| \leq g(|x|)$.

Proof. Assume that there exists a recursive g such that $(*)$ holds. We construct a decision procedure of X:

input : $x \in X$

• compute $g(|x|)$

• construct $P[g(|x|)]$

• if $P[g(|x|)]$ contains a proof of x then $x \in X$

else $x \notin X$.

But we assumed X to be undecidable, thus we arrive at a complete contradiction. ■

It follows as a corollary that there exists no proof system that generates an undecidable problem X and X is polynomially verifiable.

The result given illustrates one of the problems in acknowledging hypercomputation. Even if we have a strong hypercomputer solving, let us say, the halting problem, the verification of its correctness is ultimately unfeasible. Due to the absence of recursive bounds we cannot

https://doi.org/10.25088/ComplexSystems.18.1.131

expect to obtain a full proof of the corresponding property (halting or nonhalting) from the hypercomputer itself.

When we consider the halting problem and the property of nonhalting, this can only be verified by a proof (and not by simulating a Turing machine). By the undecidability of the problem there is no complete (recursive) proof system doing the job. So when we obtain a verification from the hypercomputer concerning nonhalting, the form of this verification lies outside computational proof systems.

However, we might think about the following test procedure for hypercomputers: humans create a test set of problems for an undecidable problem X, that is, a finite set Y with $Y \bigcap X \neq \emptyset$ and $Y \bigcap X^c \neq \emptyset$. The humans are in possession of the solutions, preferably of proofs φ_y of $y \in X$ or of $y \notin X$ for any $y \in Y$. This finite set may at least serve the purpose of *falsifying* hypercomputation (provided the hypercomputer is not stochastic and wrong answers are admitted). Beyond the possibility of falsification we might consider the following scenario: the hypercomputer answers all questions concerning the test set Y correctly, and its computing time is independent of the complexity of the proofs φ_y. Such a phenomenon would, of course, not yield a verification of the hypercomputer but at least indicate a behavior structurally differing from computable proof systems.

But the ultimate barrier of verifying a hypercomputer is that of verifying a black box, characterized by the attempt to induce a property of infinitely many input/output pairs by a finite test set.

5. Discussion and Summary

The considerations presented here may be viewed as special cases of a very general black box identification problem: is it possible to deduce certain features of a black box, without opening the box and without knowing the intrinsic working of the black box, from its input/output behavior alone? Several issues of this general problem have already been discussed. For instance, in an effort to formalize the uncertainty principle, Moore [42] considered initial state identification problems of (given) deterministic finite automata. Gold [43-47] considered a question related to induction: if one restricts black boxes to computable functions, then the rule inference problem, that is, finding out which function is implemented by the black box, is in general unsolvable. The halting problem [48-50] can be translated into a black box problem: given a black box with a known partial recursive function, then its future behavior is generally unpredictable. Even the problem of determining whether or not a black box system is polynomial in computation space and time appears to be far from trivial.

So, if presented with a hypercomputer or oracle, we could only assert heuristic information, nothing more. We have to accept the fact that more general assertions, or even proofs for computational capaci-

https://doi.org/10.25088/ComplexSystems.18.1.131

ties beyond very limited finite computational capacities remain impossible, and will possibly remain so forever.

The situation is not dissimilar from claims of absolute indeterminism and randomness on a microphysical scale [37], where a few, albeit subtle, tests of time series [38] generated by quantum randomness oracles such as *Quantis* [41] can be compared against advanced algorithmic random number generators such as the Rule30CA Wolfram rule 30 generator implemented by *Mathematica*. Beyond heuristic testing, any general statement about quantum randomness remains conjectural.

Acknowledgments

This manuscript grew out of discussions between computer scientists and physicists at the Vienna University of Technology, including, among others, Erman Acar, Bernhard Gramlich, Markus Moschner, and Gernot Salzer.

References

[1] S. Wolfram, *A New Kind Of Science*, Champaign, IL: Wolfram Media, Inc., 2002.

[2] K. Svozil, "Computational Universes," *Chaos, Solitons & Fractals*, **25**(4), 2006 pp. 845-859. dx.doi.org/10.1016/j.chaos.2004.11.055.

[3] P. W. Bridgman, "A Physicist's Second Reaction to Mengenlehre," *Scripta Mathematica*, **2**, 1934 pp. 101-117, 224-234; cf. R. Landauer [51].

[4] G. J. Chaitin, *Algorithmic Information Theory*, Cambridge: Cambridge University Press, 1987.

[5] C. S. Calude and M. J. Dinneen, "Exact Approximations of Omega Numbers," *International Journal of Bifurcation and Chaos (IJBC)*, **17**(6), 2007 pp. 1937-1954 (CDMTCS Research Report Series 293). dx.doi.org/10.1142/S0218127407018130.

[6] C. S. Calude, E. C. Calude, and M. J. Dinneen, "A New Measure of the Difficulty of Problems," *Journal for Multiple-Valued Logic and Soft Computing*, **12**, 2006 pp. 285-307 (CDMTCS Research Report Series 277). www.cs.auckland.ac.nz/CDMTCS//researchreports/277cris.pdf.

[7] H. Weyl, *Philosophy of Mathematics and Natural Science*, Princeton: Princeton University Press, 1949.

[8] A. Grünbaum, *Philosophical Problems of Space and Time (Boston Studies in the Philosophy of Science, Vol. 12*, 2nd ed., Dordrecht, The Netherlands: D. Reidel Publishing Co., 1973.

[9] J. F. Thomson, "Tasks and Supertasks," *Analysis*, **15**, 1954 pp. 1-13.

[10] P. Benacerraf, "Tasks and Supertasks, and the Modern Eleatics," *Journal of Philosophy*, **LIX**(24), 1962 pp. 765-784.

https://doi.org/10.25088/ComplexSystems.18.1.131

[11] R. Rucker, *Infinity and the Mind*, Boston: Birkhäuser, 1982; New York: Bantam Books, 1983 (reprint).

[12] I. Pitowsky, "The Physical Church-Turing Thesis and Physical Computational Complexity," *Iyyun*, **39**, 1990 pp. 81-99.

[13] J. Earman and J. D. Norton, "Forever Is a Day: Supertasks in Pitowsky and Malament-Hogart Spacetimes," *Philosophy of Science,* **60**(1), 1993 pp. 22-42.

[14] M. Hogarth, "Predicting the Future in Relativistic Spacetimes," *Studies in History and Philosophy of Modern Physics*, **24**, 1993 pp. 721-739.

[15] M. Hogarth, "Non-Turing Computers and Non-Turing Computability," *Proceedings of the Biennial Meeting of the Philosophy of Science Association (PSA)*, **1**, 1994 pp. 126-138.

[16] E. W. Beth, *The Foundations of Metamathematics*, Amsterdam: North-Holland, 1959.

[17] E. G. K. López-Escobar, "Zeno's Paradoxes: Pre Gödelian Incompleteness," *Yearbook 1991 of the Kurt-Gödel-Society*, **4**, 1991 pp. 49-63.

[18] K. Svozil, "The Church-Turing Thesis as a Guiding Principle for Physics," *Unconventional Models of Computation (Discrete Mathematics and Theoretical Computer Science)* (C. S. Calude, J. Casti, and M. J. Dinneen, eds.), New York: Springer, 1998 pp. 371-385.

[19] C. S. Calude and B. Pavlov, "Coins, Quantum Measurements, and Turing's Barrier," *Quantum Information Processing*, **1**(1-2), 2002 pp. 107-127. (Mar 1, 2002) arxiv.org/abs/quant-ph/0112087v3.

[20] V. A. Adamyan, C. S. Calude, and B. S. Pavlov. "Transcending the Limits of Turing Computability." (May 11, 2003) arxiv.org/abs/quant-ph/0304128v2.

[21] T. D. Kieu, "Quantum Algorithm for Hilbert's Tenth Problem," *International Journal of Theoretical Physics*, **42**, 2003 pp. 1461-1478. (Oct 8, 2003) arxiv.org/abs/quant-ph/0110136v3.

[22] T. D. Kieu, "Computing the Noncomputable," *Contemporary Physics*, **44**, 2003 pp. 51-71. (Oct 8, 2003) arxiv.org/abs/quant-ph/0203034v4.

[23] T. Ord, "The Many Forms of Hypercomputation," *Applied Mathematics and Computation*, **178**(1), 2006 pp. 143-153. dx.doi.org/10.1016/j.amc.2005.09.076.

[24] M. Davis, "The Myth of Hypercomputation," *Alan Turing: Life and Legacy of a Great Thinker* (C. Teuscher, ed.), Berlin: Springer, 2004 pp. 195-212.

[25] F. A. Doria and J. F. Costa, "Introduction to the Special Issue on Hypercomputation," *Applied Mathematics and Computation*, **178**(1), 2006 pp. 1-3. dx.doi.org/10.1016/j.amc.2005.09.065.

[26] M. Davis, "Why There Is No Such Discipline as Hypercomputation," *Applied Mathematics and Computation*, **178**, 2006 pp. 4-7. dx.doi.org/10.1016/j.amc.2005.09.066.

[27] A. Einstein and L. Infeld, *The Evolution of Physics*, Cambridge: Cambridge University Press, 1938.

[28] T. Y. Chow, "The Myth of Hypercomputation," (contribution to a discussion group on hypercomputation), 2004 cs.nyu.edu/pipermail/fom/2004-February/007883.html.

https://doi.org/10.25088/ComplexSystems.18.1.131

[29] M. Davis, *Computability and Unsolvability*, New York: McGraw-Hill, 1958.

[30] A. Shamir, "IP = PSPACE," *Journal of the ACM (JACM)*, **39**(4), 1992 pp. 869-877. dx.doi.org/10.1145/146585.146609.

[31] L. Babai, "Trading Group Theory for Randomness," in *Proceedings of the Seventeenth Annual ACM Symposium On Theory of Computing (STOC '85)*, Providence, RI (N. Beebe, ed.), 1985 pp. 421-429. dx.doi.org/10.1145/22145.22192.

[32] S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems," *SIAM Journal on Computing (SICOMP)*, **18**(1), 1989 pp. 186-208. dx.doi.org/10.1137/0218012.

[33] L. Babai and S. Moran, "Arthur-Merlin Games: A Randomized Proof System, and a Hierarchy of Complexity Classes," *Journal of Computer and System Sciences*, **36**(2), 1988 pp. 254-276.

[34] O. Goldreich, *Foundations of Cryptography: Basic Tools*, Vol. 1, Cambridge: Cambridge University Press, 2001.

[35] G. J. Chaitin, *Exploring Randomness (Discrete Mathematics and Theoretical Computer Science Series)*, London: Springer-Verlag, 2001.

[36] C. Calude, *Information and Randomness—An Algorithmic Perspective*, Berlin: Springer, 1994.

[37] K. Svozil, "The Quantum Coin Toss—Testing Microphysical Undecidability," *Physics Letters A*, **143**(9), 1990 pp. 433-437. dx.doi.org/10.1016/0375-9601(90)90408-G.

[38] C. S. Calude and M. J. Dinneen, "Is Quantum Randomness Algorithmic Random? A Preliminary Attack," in *Proceedings of the First International Conference on Algebraic Informatics*, Thessaloniki, Greece, (S. Bozapalidis, A. Kalampakas, and G. Rahonis, eds.), Thessaloniki, Greece: Aristotle University of Thessaloniki, 2005 pp. 195-196.

[39] T. Erber, "Testing the Randomness of Quantum Mechanics: Nature's Ultimate Cryptogram?" *Annals of the New York Academy of Sciences. Fundamental Problems in Quantum Theory: A Conference Held in Honor of Professor John A. Wheeler*, Vol. 755, (D. M. Greenberger and A. Zeilinger, eds.), Berlin, Heidelberg, New York: Springer, 1995 pp. 748-756. dx.doi.org/10.1111/j.1749-6632.1995.tb39016.x.

[40] D. J. Berkeland, D. A. Raymondson, and V. M. Tassin, "Tests for Nonrandomness in Quantum Jumps," *Physical Review A (Atomic, Molecular, and Optical Physics)*, **69**, 2004 pp. 052 103. dx.doi.org/10.1103/PhysRevA.69.052103.

[41] id Quantique, "Quantis - Quantum Random Number Generators (QRNG)," 2004. www.idquantique.com.

[42] E. F. Moore, "Gedanken-Experiments on Sequential Machines," *Automata Studies* (C. E. Shannon and J. McCarthy, eds.), Princeton: Princeton University Press, 1956 pp. 129-153.

[43] M. E. Gold, "Language Identification in the Limit," *Information and Control*, **10**(5), 1967 pp. 447-474. dx.doi.org/10.1016/S0019-9958(67)91165-5.

[44] L. Blum and M. Blum, "Toward a Mathematical Theory of Inductive Inference," *Information and Control*, **28**(2), 1975 pp. 125-155.

https://doi.org/10.25088/ComplexSystems.18.1.131

[45] D. Angluin and C. H. Smith, "A Survey of Inductive Inference: Theory and Methods," *Computing Surveys*, **15**, 1983 pp. 237-269.

[46] L. M. Adleman and M. Blum, "Inductive Inference and Unsolvability," *The Journal of Symbolic Logic*, **56**(3), 1991 pp. 891-900. dx.doi.org/10.2307/2275058.

[47] M. Li and P. M. B. Vitányi, "Inductive Reasoning and Kolmogorov Complexity," *Journal of Computer and System Science*, **44**(2), 1992 pp. 343-384. dx.doi.org/10.1016/0022-0000(92)90026-F.

[48] A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, Series 2 (**42** and **43**), 230-265 and 544-546 (1936 and 1937), reprinted in [52].

[49] H. Rogers, Jr., *Theory of Recursive Functions and Effective Computability*, New York: McGraw-Hill, 1967.

[50] P. Odifreddi, *Classical Recursion Theory, Vol. 1*, Amsterdam: North-Holland, 1989.

[51] R. Landauer, "Advertisement for a Paper I Like," *On Limits* (J. L. Casti and J. F. Traub, eds.), Report 94-10-056, Santa Fe, NM: Santa Fe Institute, 1994 p. 39. www.santafe.edu/research/publications/workingpapers/94-10-056.pdf.

[52] M. Davis, *The Undecidable*, New York: Raven Press, 1965.

https://doi.org/10.25088/ComplexSystems.18.1.131