Does Quantum Mechanics allow for Infinite Parallelism?

Martin Ziegler, University of Paderborn, 33095 GERMANY

Abstract

Recent works have independently suggested that Quantum Mechanics might permit for procedures that transcend the power of Turing Machines as well as of 'standard' Quantum Computers. These approaches rely on and indicate that Quantum Mechanics seems to support some infinite variant of classical parallel computing.

We compare this new one with other attempts towards hypercomputation by separating 1) its principal computing capabilities from 2) realizability issues. The first are shown to coincide with recursive enumerability; the second are considered in analogy to 'existence' in mathematical logic.

1   Computability

In 1936, ALAN M. TURING gave an example of a well-founded (and thus mathematically solvable) problem which he showed to admit no computable solution. More precisely he first introduced what is now called the Turing Machine (TM) as an idealization ('*model*') of a digital computer and revealed it capable of solving a vast variety of practical problems like, for instance, answering whether a given integer is prime.

Definition 1.1 *A computational* problem *is a subset* $L \subseteq \mathbb{N}$.
It is decided *by* TM *M if, upon input of any* $x \in \mathbb{N}$,

- *M eventually outputs "0" (*rejects*) and halts in case* $x \notin L$.
- *M eventually outputs "1" (*accepts*) and halts in case* $x \in L$

Having thus indicated the fundamental power of this machine, TURING then proceeded to exhibit its limitation by formally proving that the Halting Problem H — the question whether a given TM M eventually halts or rather continues executing indefinitely — cannot be decided by any TMM_0. Notice that, according to Definition 1.1, this hypothetical M_0 is required to always give the correct answer and to terminate. More precisely, the difficulty inherent to the Halting Problem consists in telling within finite time whether M does *not* halt; for, simply simulating M step by step, M_0 can easily identify the case when M *does* terminate.

TURING's result initiated the flourishing field of Computability or Recursion Theory [28]. Its goal is to distinguish computable from uncomputable problems and to classify the latter according to their degree of uncomputability [31]. For example, the following celebrated result of MATIYA-SEVICH has settled HILBERT's Tenth Problem to the negative by proving it equivalent to H:

Theorem 1.2 ([26]) *On the one hand, a given description of a Diophantine Equation E (like* FERMAT*'s famous "$a^n + b^n = c^n$" for $a, b, c, n \in \mathbb{N}$) can computably be transformed into a* TM *M such that it holds: M terminates iff E admits an integral solution.*

On the other hand, a given TM *M can computably be converted into the description of a Diophantine Equation E in such a way that, again, M halts iff E admits an integral solution. In particular since, according to [32], the first cannot be decided algorithmically, neither can the latter.*

Apart from the TM, many further sensible notions of computability have been proposed: e.g., μ-Recursion (which gave the field its name), HERBRAND-GÖDEL-Computability (which led to the programming language `Prolog`), or λ-Calculus (which stipulated `Lisp`). But they were all shown equivalent to the TM by CHURCH, KLEENE, POST, TURING and others; cf. e.g. [28, `Chapter I`] or [2, `Section 26.3+4`]. The popular language `Java` for example is equivalent, too: internet provides many applets for simulating a TM; and conversely devising a `Java` interpreter on a TM. is merely tedious but not difficult. Even a Pentium® processor is basically just a TM— although a very fast one; recall that we are dealing with problems which cannot be solved computationally at all, neither quickly nor slowly. For the very same reason, (at least 'standard') Quantum Computers are still no more powerful than an ordinary TM [20, p.3 footnote 1].

1.1 Church-Turing Hypothesis

It should be emphasized that both Halting and Hilbert's Tenth Problem are desirable to be solved for very practical reasons. The first for instance arises in automatized software verification; indeed, correctness of some `Java` source code includes its termination which, by the above considerations, cannot be checked algorithmically. Similarly, a hypothetical algorithm for deciding feasibility of Diophantine Equations could be applied to computer-proving not only FERMAT's Last Theorem but also to settle many other still open questions for example in number theory.

Observing that [32, 26] ruled out the possibility of a *Turing Machine* to decide either of these problems, people have since long tried to devise other computing devices exceeding its principal power. However the perpetual failure to do so plus the aforementioned results of a TM being able to simulate many other notions of computability have eventually led to what has become known as the Church-Turing Hypothesis:

> *Anything that can be computed in practice, is also computable by a TM.*

We emphasize that, due to its informal nature, this hypothesis cannot be proven formally. Informal arguments in its favor usually point out that computation is a physical process which, by mathematically describing the physical laws it is governed by, can be simulated by a TM up to arbitrary finite numerical precision; and infinite accuracy were required only for 'chaotic' processes which are too sensitive to perturbations than being harnessable for practical computation anyway.

However it has later been pointed out that certain theories of quantum gravitation might actually *not* admit a simulation by a TM [18, p.546]; furthermore even Classical Mechanics seems to conceivably provide for processes whose simulation requires infinite precision during *intermediate* times only, whereas the resulting behavior is asymptotically stable and thus suited well to realize a non-Turing form of physical computation [34]. Moreover the laws of nature we know so far cannot be deduced to fundamentally restrict computation [5].

In fact neither CHURCH nor TURING themselves have put forward a claim as universal as the way 'their' hypothesis is often (mis-)interpreted [12]. Instead, literature contains and discusses a rich variety of related hypotheses [27, SECTION 2.2].

1.2 Hypercomputation

Anyway, the question remains open whether there might exist a computing device more powerful than the TM or not.

Remark 1.3 *Already* POST *wondered whether* any *super-TM has the ability to solve the Halting Problem. This was settled in the negative by* FRIEDBERG *and* MUCHNIK *independently showing that there exists an entire hierarchy of undecidable problems strictly 'easier' than H. More precisely they constructed problems $P \subseteq \mathbb{N}$ that a TM can*not *decide, yet access to whose solution (in terms of an oracle, that is, by permitting queries to some hypothetical external device answering questions "$y \in P$") provably still does not enable this* super-TM *to solve the Halting Problem. However, in contrast to H, such P seem to be artificial. See [31,* SECTIONS 5 *to 7] for a more detailed account of* POST*'s Problem.*

To get an idea towards how a *Hyper*computer might look like, Theoreticians have started considering super-TMs and their respective fundamental computing capabilities. This established the flourishing field of research called 'Hypercomputation' [14] which entire volumes of significant journals have become dedicated to [10, 8]. Devising such a formal model (i.e., an idealized abstraction) of a Hypercomputer proceeds in many cases less by adding extensions to but rather by removing restrictions from a TM.

Observation 1.4 *The TM is characterized by*

    a) a finite control (the 'program', so to speak);

    b) an initially blank, countable supply of memory cells

    c) storing a finite amount of information each (e.g., a bit or an integer);

    d) finite running time;

    e) possibly finite parallelism (as, e.g., for a nondeterministic *TM).*

Irregardless of the details of its precise definition, these finiteness conditions directly imply that there is an at most countable number of different TMs; whereas computational problems according to Definition 1.1 exist of cardinality of the continuum. Thus, most of them are undecidable.

Conditions a)-e) underlie the mourned limitations of the classical TM, and dropping one or more of them leads to several well-known models of hypercomputation; see, e.g. [27, SECTION 3] or [13, SECTION 2]. *Oracle* Machines for instance, subject of TURING's Dissertation in Princeton [33] and now core of Recursion Theory [31, 28], correspond to TMs with initial memory inscription, that is, they remove Condition b); BLUM, SHUB, AND SMALE's ℝ-Machine [7] abolishes Condition c) by allowing each cell to store a real number; while *Infinite Time* Machines due to [21] lift Condition d).

The proposal, consideration, and investigation of such enhanced abstract models of computation and their respective computational powers by Logicians and Theoretical Computer Scientists has proven particular seminal regarding related contributions from Theoretical Physics on their realizability. For example, [4] has indicated that a physical system breaking Condition a) might actually exist[1]; while [22, 30] pointed out that in General Relativity there might exist[1] space-time geometries allowing to watch within finite time a computer execute an infinite number of steps and thus to lift Condition d).

1.3 Quantum Mechanical Hypercomputation

Recently, several new approaches have been suggested for solving either the Halting Problem [9, 11, 1] or Hilbert's Tenth Problem [24, 25]. They exploit Quantum Mechanics and thus form a nice counterpart to previous approaches based on General Relativity [22, 30] as the other pillar of non-classical physics. Recalling that 'standard' Quantum Computing does not exceed Turing's Barrier, these approaches must be non-standard in some sense which closer inspection reveals to be infinite parallelism:

- *"Our quantum algorithm is based on [. . .] our ability to implement physically certain Hamiltonians having infinite numbers of energy levels"*   [24, top of SECTION 6.3];
- *"The key ingredients are the availability of a countably infinite number of Fock states, the ability to construct/simulate a suitable Hamiltonian"*   [25, end of SECTION 4];
- *"The new ingredients built in our 'device' include the use of an infinite superposition (in an infinite-dimensional Hilbert space) which creates an 'infinite type of quantum parallelism'"*   [11, p.123 SECTION 5].

It thus seems that Quantum Mechanics allows to drop Condition e) from Observation 1.4 and so to provide a new promising approach to the existence[1] of hypercomputers — an approach *not* covered by ORD's classification [27, SECTION 3]. The present work describes in Section 2 the theoretical consequences from lifting Condition e), that is the principal computing power of infinite parallelism.

We conclude this section with an already announced remark on the notion of existence.

---

[1]refer to Remark 1.5 below

Remark 1.5 *The question whether a physical device with certain properties* exists *bears logical similarity to the question whether a mathematical object with certain properties exists. In the latter case for a proof, only very few (namely constructive or intuitionistic) mathematicians will*

    A) insist that one actually constructs *this object*

whereas most are satisfied for instance with

    B) an indirect argument showing that its non-*existence leads to a contradiction.*

In fact a majority of contemporary mathematicians will even take it for granted if

    C) the object's existence *does* not *lead to a contradiction.*

For example the claim "To every vector space there exists a basis" *is of kind C) as well as many principles in* **Functional Analysis**: *Each of them is equivalent to the* **Axiom of Choice** *[6] and thus does not lead to a contradiction to conservative set theory (C) but cannot be deduced from it (B) as has first been proven by* K. Gödel *in [19] and later strengthened by* P.J. Cohen.

Similarly, the existence of a physical object can be proven in a strong way A) by actually constructing ist. But in most cases, showing it C) consistent with physical laws is accepted as well. Observe that this is how both Positrons as well as Black Holes came to existence: as solutions of (and thus consistent with) Dirac*'s Equation and* Einstein*'s General Relativity, respectively; only later have new experimental observations upgraded their existence to type B).*

2   Infinite Parallel Computing

The prospering field of Parallel Computing knows and has agreed upon a small collection of models as theoretical abstractions for devising and analyzing new algorithms for various actual parallel machines [2, Sections 45.2 and 47.2]. Of course with respect to their principal power, that is computability rather than complexity, they are all equivalent to the TM.

However when talking about *infinite* parallelism, seemingly no such agreement has been established, cf. e.g. [15, p.284]; and in fact no equivalence, either, as will turn out. For instance, of what kind are the countably infinitely many individual computers that are to operate concurrently — TMs or finite automata? In the first case, do they all execute the same program? When is the result to be read off? The answers to these questions fundamentally affect the capabilities of the resulting system.

2.1   Infinite Cellular Automata

Consider parallelism in an infinite cellular automaton in the plane. More specifically, we refer to Conway's famous **Game of Life** [3, Ch. 25] where in each step, any cell's successor state concurrently is determined by its present state as well as those of its eight adjacent ones' as follows (cf. Figure 1):

- A `dead` cell with exactly three neighbors `alive` becomes `alive`, too;
  otherwise it remains `dead`.

- A living cell with two or three neighbors `alive` stays `alive`;
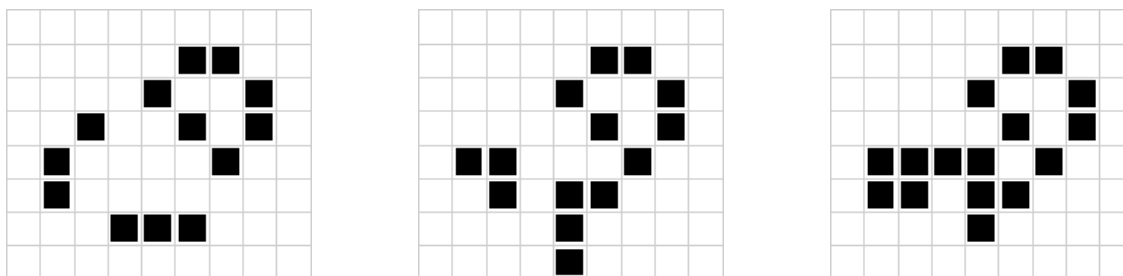 otherwise (0,1,4...8 living neighbors, that is) it dies.

Figure 1: Sample Initial and Two Successor Configurations of Life.

Definition 2.1 *Starting from a given initial configuration,* Life *terminates if the sequence of successor configurations eventually stabilizes. This resulting configuration is called* rejecting *if it is empty (every cell* dead*), otherwise* accepting.
An initial configuration is finite *if, out of the infinite number of cells, only finitely many are* alive.

Theorem 2.2 Life *with finite initial configuration is a system with infinite parallelism yet equivalent to the* TM. *More precisely:*

i) *Given a finite initial configuration, its evolution through* Life *can be simulated by a* TM.

ii) *There is a finite initial configuration capable of simulating the Universal (and thus any)* TM.

Simulation *here means: The* TM *terminates/accepts/rejects iff* Life *terminates/accepts/rejects.*

Proof: The internet provides plenty implementations for Life written, e.g., in Java. Feeding one to the aforementioned Java interpreter for TMs yields i). Claim ii) is in fact a famous result based on an ingenious and complicated construction; refer to [3, CH. 25] for details. □

In particular, Life matches but does *not* exceed the computing capabilities of a TM; cf. [2, SECTION 26.4(1)].

Here, finiteness of the initial configuration enters crucially of course. One can indeed show that *in*finite initial configurations in ii) correspond to non-blank memory contents and thus to dropping in Observation 1.4 both Conditions e) *and* b).

2.2 Infinite Turing Concurrency

In order to focus on the power obtained from infinite parallelism *only* (that is, by removing just Condition e), now consider the finite automata replaced by TMs. Indeed the three citations in Section 1.3 indicate that, whereas Quantum Computing — and in particular simulating a single classical TM — requires only a finite (or at most countably infinite) number of dimensions from quantum mechanical Hilbert Space, its *in*finitely many dimensions provide room for an infinite number of TMs: cf. Hilbert's Hotel.

Strictness of CHOMSKY's Hierarchy implies that a single TM is provably more powerful than a single automaton [2, SECTION 25.3]. One may therefore expect that the capabilities of an infinite number of TMs exceed those of an infinite number of automata (and thus actually lead to hypercomputation); by how much, however, turns out to depend.

In analogy to Definition 1.1, consider first the following notion of solving a problem by means of infinite parallelism.

Definition 2.3 *Fix a problem $L \subseteq \mathbb{N}$ and a countably infinite family $(M_k)_{k \in \mathbb{N}}$ of* TMs. *This family solves L if, for each $x \in \mathbb{N}$,*

i) *each M_k, upon input x, eventually terminates and*

ii) *at least one M_k outputs "1" (accepts) iff $x \in L$.*

However observe that, whereas each individual M_k halts, the time required to do so may depend on k so that it takes infinitely long for the entire family $(M_k)_k$ to terminate. (We point out that this behavior resembles the *fair infinite nondeterminism* of [16].) In order to know the result within finite time, the following additional requirement is therefore important:

Definition 2.3 (continued)

iii) *if upon input of any $x \in \mathbb{N}$, all M_k terminate within finite time bounded independently of k.*

While seeming sensible at first glance, a second thought reveals that, even with this restriction, the resulting notion of 'infinitely parallel computability' is still unreasonable: simply because *any* problem $L \subseteq \mathbb{N}$ becomes trivially solvable by an appropriate family $(M_k)_k$. To this end let the program executed by M_k store the constant "1" if $k \in L$ and the constant "0" otherwise. Let its main part then operate as follows: Upon input of $x \in \mathbb{N}$ test whether $x = k$; if so, output the stored constant, otherwise output "0"; then terminate.

The point is of course that the according family $(M_k)_k$ solving L is only shown to *exist*. More precisely there is in general no means of *computing*, given $k \in \mathbb{N}$, a description of M_k and its constants. This insight suggests to finally add the following further requirement, in Theoretical Computer Science known as a *uniformity* condition.

Definition 2.3 (concluded)

iv) if a TM M_0 *is capable of generating, upon input of k, (the encoding of)* M_k.

Here, *encoding* refers to a sort of 'blueprint' of M_k or, more formally, its GÖDEL Number [23, SECTION 9.1.2].

2.3 Computational Power of Infinite Turing Concurrency

This section reveals that the Definition 2.3(i-iv) indeed yields an interesting non-trivial way of hypercomputation. More precisely we show that, in this sense, infinite Turing-Parallelism can

- solve the Halting Problem H
- as well as Hilbert's Tenth Problem
- but *not* Totality.

While H refers to the question whether given TM M, started on a single given input x, eventually terminates, Totality asks whether M halts on *all* possible inputs. So in contrast to the first, this even more important property of correct software still remains intractable to automated checking even on this kind of hypercomputer.

Theorem 2.4 *The Halting Problem is solvable by an infinity of* TMs *working in parallel in the sense of Definition 2.3(i-iv).*

Proof: For each $k \in \mathbb{N}$, let M'_k proceed as follows: Given M and x, simulate the first k steps of M operating on x; if M halts within these steps, then output "1" and terminate; otherwise output "0" and terminate.
Observe that the family $(M'_k)_k$ satisfies i) and ii) from Definition 2.3. Moreover, one easily confirms (iv) that an appropriate TM M'_0 can indeed generate from k an encoding of this M'_k. Based on the Universal TM U, M'_k can be achieved to have running time $t(n) \leq c \cdot (n \cdot k)^2$ for some constant c; combine for example [17, the LEMMA in SECTION 4] with [23, THEOREM 8.10]. Here, $n = |x| + |M|$ denotes the joint length of the binary encoding of x and M.
Now let M_k be the TM obtained from applying the below *Linear Speed-Up Lemma* 2.5 to M'_k with $C := k^3$. It follows that M_k has running time independent of k, that is, it does comply with iii) while still satisfying i), ii), and iv). □

In order to achieve Property iii) in the above proof, the crucial ingredient is the below classical construction. It basically says that any TM can be speed up by a *constant* factor.

Lemma 2.5 (Linear Speed-Up) *For each $C \in \mathbb{N}$ and any* TM M' *of time complexity $t(n)$, there exists another* TM M *simulating M' within running time $n + t(n)/C$.*
M *can be obtained computationally from M'; i.e., there is a fixed further* TM *which, given an encoding of any M' and C, outputs an encoding of M as above.*

Proof: See for instance [2, THEOREM 24.5(b)]. □

The solvability of Hilbert's Tenth Problem now follows from Theorem 1.2 and Theorem 2.4. More generally, the latter implies that infinite Turing-Parallelism can solve *any* semi-decidable Problem $L \subseteq \mathbb{N}$. In fact, the converse holds as well:

Theorem 2.6 *A Problem $L \subseteq \mathbb{N}$ is solvable in the sense of Definition 2.3(i-iv) iff semi-decidable.*

Recall that '*semi*-decidability' (also called *recursive enumerability*) weakens 'decidability' from Definition 1.1 in that, here, the TM is allowed in case $x \notin L$ to not halt but to loop endlessly [23, SECTION 825].

Proof: By the above remark, it remains to consider the case that L is solvable by some parallel family $(M_k)_k$ according to Definition 2.3.

Upon input of $x \in \mathbb{N}$, sequentially simulate the $(M_k)_k$ as follows: For each $k \in \mathbb{N}$,

- obtain from M_0 a description of M_k by virtue of iv)

- and simulate M_k on input x.   (Observe its termination according to Property i)

- If output is "1", halt; otherwise proceed with next k.

This algorithm indeed terminates iff at least one M_k outputs "1", that is (ii), iff $x \in L$.   □

Corollary 2.7 *Even infinite Turing concurrency in the sense of Definition 2.3(i-iv) cannot solve* Totality.

Proof: Totality is well-known to *not* be semi-decidable. More specifically, we refer to [31, THEOREM IV.3.2] where this problem is shown to be Π_2-complete, that is, [31, DEFINITION IV.2.1 and COROLLARY IV.2.2] reducible to $\overline{\emptyset^{(2)}} \notin \Sigma_2$, and therefore does not belong to the class $\Sigma_1 \subseteq \Sigma_2$ of recursively enumerable problems.   □

3   Conclusion

Section 1 has pointed out that recent and independent approaches due to KIEU, CALUDE, and PAVLOV to hypercomputation via quantum mechanics rely on some sort of infinite parallelism. Regarding the respective complicated intertwined quantum mechanical constructions, procedures, and analyses, we suggest to bring more clarity into this subject by considering algorithmic/ computational issues separately from physical ones. This leads to the following two questions to be treated individually:

1) Does Quantum Mechanics allow for infinite parallelism; and, if so, of what kind?

2) What kinds of idealized infinite parallelism yield which principal computational power; that is, does it and by how far exceed the fundamental capabilities of a TM?

Section 2 contains answers to the second question. It reveals that in fact infinite *classical* (i.e., Turing-) parallelism is sufficient for solving both the Halting Problem as well as Hilbert's Tenth Problem. This leaves open whether the infinite dimensions of quantum mechanical Hilbert Space do indeed allow for this kind of infinite classical parallelism. Specifically,

- preparation of a certain initial state,
- its maintenance (in particular coherence) through-out the computational evolution,   and
- read-out of the final result

are likely to raise here even more difficulties than already in the finite-dimensional case of 'standard' Quantum Computing [20, SECTION 7.2]. For example only recently has it become possible to read out a single spin [29]. However (im-)practicality of hypercomputation should not be confused with (un-)existence, particularly in the light of Remark 1.5.

References

[1] V.A. ADAMYAN, C.S. CALUDE, B.S. PAVLOV: "Transcending the limits of Turing computability", to appear in *International Journal of Theoretical Physics*;   `http://arXiv.org/quant-ph/0304128`

[2] M.J. ATALLAH (Edt): "*Algorithms and Theory of Computation Handbook*", CRC Press (1999).

[3] E.R. BERLEKAMP, J.H. CONWAY, R.K. GUY: "*Winning Ways for Your Mathematical Plays*" vol.**4**, Academic Press 2nd Edition (2004).

[4] E.J. BEGGS, J.V. TUCKER: "Computations via Experiments with Kinematic Systems", *Technical Report* 5–2004, Department of Computer Science, University of Wales Swansea.

[5] C.H. BENNETT, R. LANDAUER: "The Fundamental Physical Limits of Computation", pp.48–56 in *Scientific American*, vol.**253(1)** (1985).

[6] A. Blass: "Existence of bases implies the axiom of choice", pp.31–33 in *Axiomatic Set Theory, Contemporary Mathematics* vol.**31** (1984).

[7] L. Blum, M. Shub, S. Smale: "On a Theory of Computation and Complexity over the Real Numbers: $\mathcal{NP}$-Completeness, Recursive Functions, and Universal Machines", pp.1–46 in *Bulletin of the American Mathematical Society* (AMS Bulletin) vol.**21** (1989).

[8] M. Burgin, A. Klinger (Eds.): "Super-Recursive Algorithms and Hypercomputation", vol.**317** in *Theoretical Computer Science*, Elsevier (2004).

[9] C.S. Calude, M.J. Dinneen, K. Svozil: "Reflections on Quantum Computing", pp.35–37 in *Complexity* vol.**6(1)**, Wiley (2001).

[10] C.S. Calude, M.J. Dinneen, F. Peper (Eds.): "Unconventional Models of Computation", vol.**2509** in *Lecture Notes in Computer Science*, Springer (2002).

[11] C.S. Calude, B. Pavlov: "Coins, Quantum Measurements, and Turing's Barrier", pp.107–127 in *Quantum Information Processing* vol.**1**, Plenum (2002).

[12] J. Copeland: "The Broad Conception of Computation", pp.690–716 in *American Behavioural Scientist* vol.**40** (1997).

[13] J. Copeland: "Hypercomputation", pp.461–502 in *Minds and Machines* vol.**12**, Kluwer (2002).

[14] B.J. Copeland, D. Proudfoot: "Alan Turing's Forgotten Ideas in Computer Science", pp.98–103 in *Scientific American* vol.**280(4)** (1999).

[15] E. Eberbach, P. Wegner: "Beyond Turing Machines", pp.279–304 in *Bulletin of the European Association for Theoretical Computer Science* (EATCS Bulletin) vol.**81** (2003).

[16] P. van Emde Boas, E. Spaan, L. Torenvliet: "Nondeterminism, Fairness and a Fundamental Analogy", pp.186–193 in *The Bulletin of the European Association for Theoretical Computer Science* (EATCS Bulletin) vol.**37** (1989).

[17] M. Fürer: "The tight deterministic time hierarchy", pp.8-16 in *Proc. 14th ACM Symposium on Theory of Computing* (1982).

[18] R. Geroch, J.B. Hartle: "Computability and Physical Theories", pp.533–550 in *Foundations of Physics* vol.**16(6)** (1986).

[19] K. Gödel: "*The Consistency of the Axiom of Choice and of the Generalized Continuum-Hypothesis With the Axioms of Set Theory*", Princeton Univ Press (1940).

[20] J. Gruska: "Quantum Computing", McGraw-Hill (1999).

[21] J.D. Hamkins, A. Lewis: "Infinite Time Turing machines", pp.567–604 in *Journal of Symbolic Logic* vol.**65(2)** (2000).

[22] M.L. Hogarth: "Does General Relativity Allow an Observer to View an Eternity in a Finite Time?", pp.173–181 in *Foundations of Physics Letters* vol.**5(2)** (1992).

[23] J.E. Hopcroft, R. Motwani, J.D. Ullman: "*Introduction to Automata Theory, Languages, and Computation*", Addison-Wesley (2001).

[24] T. Kieu: "Computing the non-computable", pp.51–71 in *Contemporary Physics* vol.**44(1)** (2003).

[25] T. Kieu: "Quantum Algorithm for Hilbert's Tenth Problem", pp.1461–1478 in *International Journal of Theoretical Physics* **42**, Springer (2003).

[26] Y.V. Matiyasevich: "Enumerable Sets are Diophantine", pp.354–357 in *Soviet Math.Dokl.* vol.**11** (1970).

[27] T. Ord: "*Hypercomputation: computing more than the Turing machine*", Honours Thesis, University of Melbourne (2002); `http://arXiv.org/math.LO/0209332`

[28] P. Odifreddi: "*Classical Recursion Theory*", North-Holland (1989).

[29] D. Rugar, R. Budakian, H.J. Mamin, B.W. Chui: "Single spin detection by magnetic resonance force microscopy", pp.329–332 in *Nature* vol.**430** (2004).

[30] O. Shagrir, I. Pitowsky: "Physical Hypercomputation and the Church-Turing Hypothesis", pp.87–101 in *Minds and Machines* vol.**13** (2003).

[31] R.I. Soare: "*Recursively Enumerable Sets and Degrees*", Springer (1987).

[32] A.M. Turing: "On Computable Numbers, with an Application to the Entscheidungsproblem", pp.230–265 in *Proc. London Math. Soc.* vol.**42(2)** (1936).

[33] A.M. Turing: "Systems of Logic Based on Ordinals", pp.161–228 in *Proc. London Math. Soc.* vol.**45** (1939).

[34] A.C.-C. Yao: "Classical Physics and the Church-Turing Thesis", pp.100–105 in *Journal of the Association for Computing Machinery* vol.**50(1)** (2003).