

On More or Less Appropriate Notions of ‘Computation’

Andy M. Gravell amg@ecs.soton.ac.uk (CORRESPONDING),
Stefan Gruner sg@cs.up.ac.za (CORRESPONDING)

^a School of Electronic and Computer Science, University of Southampton, United Kingdom

^b Department of Computer Science, University of Pretoria, South Africa

1 INTRODUCTION

Half a century after the emergence of computer science (a.k.a. informatics) as an academic discipline, the notion of ‘computation’ is not yet ‘settled’. On the contrary: recent developments in the natural sciences, in mathematics, as well as in computer hardware engineering have also ‘shaken’ the belief in the sufficiency of the ‘classical’ notion of ‘computation’ from the tradition of the Church-Turing-Hypothesis.

In his seminal medicine-historic and science-philosophical book (1935), the medical scientist and philosopher Ludwik Fleck has demonstrated (inter alia) how the notion of ‘illness’ has undergone so profound changes during the past five centuries that a medieval medicus and a contemporary clinical practitioner—if brought together for conversation—would hardly be able to understand each other. After Fleck’s book (1935) had almost fallen into oblivion it was by mere chance re-discovered; then it strongly influenced Thomas Kuhn’s well-known notion of ‘incommensurability’ (1962). If we accept those science-historical and science-philosophical insights as valid—for comparison see also Kurt Hübner’s critique of scientific reason (1983)—then we can no longer naively ask in our field of inquiry: ‘what is’ a computer, or ‘what is’ computation? Instead we are obliged to ask: which (historically volatile) *notions* of ‘computer’ and ‘computation’ do we wish to accept for now, and which plausible reasons and rational philosophical arguments can we muster in defence of our choice of acceptance? The same type of consideration is also applicable in all the many recent *culture versus nature* disputes (Schiemann, 2014, 2008) in the context of which the question can arise whether computation is a cultural-technical artefact or whether computation can be empirically ‘found’ (like motion or forces) in the realm of nature; we are then immediately confronted with questions concerning the *historical semantics* of the *notion* of ‘nature’ as well as—again—the historical semantics of the notion of ‘computation’, whereby

Gravell, A.M. and Gruner, S. (2018). On More or Less Appropriate Notions of ‘Computation’ [Viewpoint]. *South African Computer Journal* 30(1), 161–181. <https://doi.org/10.18489/sacj.v30i1.580>

Copyright © the author(s); published under a [Creative Commons NonCommercial 4.0 License \(CC BY-NC 4.0\)](https://creativecommons.org/licenses/by-nc/4.0/). SACJ is a publication of the South African Institute of Computer Scientists and Information Technologists. ISSN 1015-7999 (print) ISSN 2313-7835 (online).

we know (at least since Kant) that we can only ‘find’ in nature of what our minds are already aware.

Recent computer-philosophical literature provides quantitatively ample evidence—the publications are far too numerous to be exhaustively cited here—in support of the conjecture that the very *notion of ‘computation’* (derived from the Latin verb ‘computare’) *has come under critical dispute and is no longer naively taken for granted* as something self-understanding or simply ‘given’. The same fate had happened already to scientific notions like ‘illness’ or ‘force’ during the histories of medicine (Fleck, 1935) and physics (Hübner, 1983), such that we need not feel particularly surprised about the emerging science-philosophical disputes on the notion of ‘computation’ in our time. Hence, we can in this paper only argue—hopefully with good and intersubjectively compelling reason—for *our* notion of ‘computation’, without being anyhow able to re-enter into the lost paradise of naive certainty (the entrance gate of which is guarded by the ‘angel of time’).

Before any kind of technical apparatus came to be understood as a ‘computer’, *a computer was a human being*, or—more precisely—a human being in a specific *profession* (Bauer, 1972). Thus wrote, for example, Gottfried Wilhelm Leibniz (1646–1716) as an expression of one of his ambitious-utopic aspirations:

Quo facto quando orientur controversiae, non magis disputatione opus erit inter duos philosophos quam inter duos computistas. Sufficient enim calamos in manus sumere, sederque ad abacos et sibi mutuo (accito si placet amico) dicere: ‘calculemus!’¹

There we can see that the ‘computers’ are the professional human experts who operate the Abacus device—*not the device itself* which was used merely as an auxiliary tool by those experts. Only much later, in the 20th century, a ‘computer’ came to be understood as a specific type of *apparatus*, an artificial-technical device (in service of its human operators)—that is: *outside the realm of nature* (Schiemann, 2014, 2008)—whereby even in Turing’s seminal paper of 1936 we can still find an ‘echo’ of this old notion of a computer as a ‘clerk’ by whom ‘symbols’ are ‘written’ (Mounier-Kuhn, 2015; Turing, 1937). Beheld in this historic perspective, the *ontic* question ‘what is a computer?’ (Piccinini, 2007) is closely linked to the *pragmatic* question: ‘what is, *with which purpose*, being computed?’ by such a device—i.e., what ‘goes in’, what ‘comes out’, how are input and output *interpreted and understood* (Penrose, 1997; Schwandt, 1999), and what *epistemic goals* are *intentionally pursued* by the operators (as the supervisory *agents*) of such computation?

By contrast: the ideology of *pancomputationalism* (Bozsahin, 2016), which started with Konrad Zuse’s famous *computing space* paper and book (Zuse, 1969a, 1969b) according to which the entire universe is one gigantic computer that computes itself, and which also resulted in the establishment of the new academic field of ‘natural computing’ as a sub-discipline of computer science—see, for example, (Paun, Rozenberg, & Salomaa, 2005)—does not want to know any such human-cultural-technological history and understands ‘computation’ *a-historically* as any

¹Thus, whenever controversies arise, no more effort of disputation will there be between two philosophers than between two computers. It suffices then namely to take a pen into the hand, to sit down at the Abacus, and (if they like, even in the presence of a friend) to say to each other: let us calculate! [Leibniz, *De arte characteristica ad perficiendas scientias ratione nitentes*]

‘natural’ process in which any chain of distinguishable events can somehow be *described* by means of some suitable algorithmic *language*—i.e., an artificial language that human wisdom has designed during the most recent history of the 20th century. Radical pancomputationalism (in its most extremist form), which is philosophically family-related to the metaphysics of pantheism, must be considered as ‘ideology’ from a Popperian point of view (K. R. Popper, 1935), because we cannot—not even in principle as a mere thought-experiment—design any experiment by means of which either H_0 (‘the entire universe is essentially a computer’) or its opposite \hat{H}_0 (‘it is not the case that the entire universe is essentially a computer’) could be empirically-effectively refuted. This impossibility of effective-empirical H -refutation is due to our immersion into this universe of which we can always only observe some limited part—even if we would be able to find non-computing parts inside our reach of vision we could still not jump to conclusions about the ‘essence’ of universe as a whole.² In addition to pancomputationalism’s a-historicity comes in some cases (as mentioned above) a philosophically un-reflected notion of ‘nature’ which is portrayed in hypostasis like an intentionally acting quasi-person.³ The ultimate consequences of such extreme pancomputationalism are meaningless epistemological trivialities—e.g., the entire universe ‘computes itself’—which no longer allow us to make any sensible distinction between what is ‘computation’ and what is not. Moreover: the implicit and un-acknowledged *technomorphism* (albeit in the guise of the above-mentioned a-historic naturicism) of pancomputationalism becomes obvious when we take into account what our entire universe used to ‘be’ before it ‘became’ a ‘computer’: in the era of early-modern mechanics our universe ‘was’ a huge mechanical ‘machine’ or a giant ‘clockwork’; now it supposedly ‘is’ a giant ‘computer’—thus it seems that it is always mankind’s own latest technical-artificial invention or achievement which determines what our universe supposedly ‘is’—in analogy to the notorious technomorphistic ‘transformation’ of the human mind from ‘being’ a mechanical mill via ‘being’ an electro-telephonic switching centre to ‘being’ an algorithm during the modern history of ideas. Closely related to this science-philosophical problem of pancomputationalism (and computation in general) are the questions:

- What is ‘information’ (Oberholzer & Gruner, 2016; Primiero, 2016)? Is ‘information’ anyhow ‘inherent’ (per se) in nature, such that we only need to ‘fetch’ it from there? Or is it the human mind that somehow ‘projects’ (in some creative act) a pre-known category of ‘information’ onto the natural objects so perceived (as per Kant)?
- Is computation (as ‘information-processing’) a happening for which (physical) energy is necessarily needed?⁴ Or can computation somehow be had ‘for free’, i.e., without the involvement of the *2nd Hauptsatz* (main law) of Thermodynamics?
- What are the necessary and/or sufficient epistemological conditions under which a given

²By analogy: none of our many biological cells is a human being.

³Such metaphysical naturicism in radical pancomputationalism is philosophically family-related to various forms of metaphysical historicism in which quasi-personhood and purposeful agency are ascribed to history itself.

⁴Implication: if Zuse’s radically pancomputationalist universe ‘computes itself’, in its entirety, then from where does the energy for such computation come—perhaps from any un-imaginable metaphysical source ‘outside’ the universe?

(or chosen) *notion* (meaning) of ‘computation’ can be (or become) *intellectually fruitful* (salient), without being too rigid (i.e., no progress in the history of ideas) as well as without being too vague (i.e., pseudo-science) in the ever-evolving web-work of science (Fleck, 1935)? In other words: which notion of ‘computation’ could fruitfully improve mankind’s reliable world-knowledge *and* technical [cap]abilities, or which notion of ‘computation’ would lead us astray into the voids of ignorance (perhaps disguised by a false facade of shallow superficial eloquence and vain terminological sophistication)?

After these introductory remarks of above, in which we have merely sketched (for the sake of illustration) a few of many further interesting problematic issues in this large ‘domain of discourse’, we cursorily (not exhaustively) recapitulate some relevant recent literature in the next following section; subsequently we describe and defend our own ideas about what we want to regard as ‘computers’ and ‘computation’ *w.r.t.* our *epistemic and pragmatic interests and purposes* as theoretical scientists (‘beholding and knowing’) and practical engineers (‘making and providing’).

2 BRIEF LITERATURE OVERVIEW

The literature of our discourse is already so vast that it can no longer be reviewed and summarised exhaustively in any one paper. Omissions—even of very important papers—in the subsequent paragraphs of this section are thus inevitable. Recently ‘trending’ topics in this discourse include (Acar, 2012): ‘fuzzy’ or non-deterministic computing with ‘fuzzy’ or non-deterministic machines (Li, 2008; Syropoulos, 2016, 2006), ‘infinite-’ or ‘continuous-time’ computing by ‘infinite-’ or ‘continuous-time Turing Machines’ (Bournez & Campagnolo, 2008; Hamkins, 2002; Hamkins & Lewis, 2000), ‘interactive’ (Goldin & Wegner, 2008) or ‘reactive’ Turing machines (also including: infinite alphabets) (Baeten, Luttik, & Van Tilburg, 2013; Luttik & Yang, 2014, 2016) that can communicate or receive ‘oracles’ (Franchette, 2015; Resconi & Licata, 2012) while running, structurally self-modifying Turing machines (Ramezani, 2014), ‘quantum computing’ in which many computations can be super-positioned into one, ‘analogue computation’ by means of various ‘classical’ (non-quantum and non-discrete) physical systems (Siegelmann & Sontag, 1994), ‘relativistic’ computing with the aim of somehow exploiting the space-time effects described by Einstein’s theories (Andréka, Németi, & Németi, 2009), ‘chemical’ and/or ‘biological’ and/or ‘bio-chemical’ computing with the aim of using sufficiently complex molecules or living cells or entire organisms as computing devices (Maldonado, Cruz, & Nelson, 2015), as well as ‘hybrid’ approaches in which any of the previously mentioned ideas can be anyhow mixed and combined: see for example (Syropoulos, 2009). Entire journals and series of conference proceedings are nowadays dedicated to the theme of ‘unconventional’ computing (International Journal of Unconventional Computing, Unconventional Models of Computation: International Conference Proceedings: Lecture Notes in Computer Science, Workshop Physics and Computation), whereby especially the possibility of ‘hyper-computing’ (da Costa & Doria, 2006; Hagar & Korolev, 2007; Loff & Costa, 2009; MacLennan, 2009; Teuscher & Sipper, 2002; Wegner & Goldin, 2003)—i.e., being able to decide the classically undecidable Halting Problem effectively and in general by new (whatever

‘new’ means) devices—has been (and is still being) much contested and debated (in some cases even by naming the opponents) (Cockshott, 2015; Cockshott & Michaelson, 2007; Cotogno, 2009; Kieu, 2006; Michaelson & Cockshott, 2006; Müller, 2011; Nayebi, 2014; Smith, 2006; Welch, 2004). Moreover, whenever any new ‘models’ of computation—for example: ‘membrane’ computing (Gheorghe & Stannett, 2012; Stannett, 2012, 2014)—were proposed, their own new complexity theories (Blakey, 2011) and/or ‘power’ of computability theories (Beggs & Tucker, 2007a, 2006; Costa, 2017; Shang, Lu, & Lu, 2015, 2012; Wüthrich, 2015; Ziegler, 2005) had to follow en-suite with many subtle technical details. Some papers have also appeared in which questions concerning the ‘scientific-ness’ (Bringsjord & Zenzen, 2002; Stannett, 2001; Ziegler, 2009) of all those various ideas have been asked from a higher-level science-philosophical vantage point—e.g., are the claims and hypotheses of this discourse empirically refutable?—etc. Related to all this are the well-known meta-mathematical disputes about ‘finitism’ and ‘ultra-finitism’, which arose already in the context of the mathematical *Grundlagenstreit* of the 1920s, as well as the even much older meta-physical and nature-philosophical disputes about whether the ultimate reality of the entire universe itself is ‘discrete’ (in which case our analog-continuous thought-models of it would be merely ‘convenient’ approximations) or ‘continuous’ (in which case, vice versa, our discrete-digital model-representations of it would be merely ‘convenient’ approximations). In summary, the entire recent debate on what is ‘computation’, and what are the limits thereof (Cockshott, Mackenzie, & Michaelson, 2010, 2008), must be seen in the contexts of

- Meta-physics and ontology of nature (what is ‘real’?),
- Meta-mathematics (can or shall we accept ‘infinity’ ∞ ?),
- Philosophy of science (w.r.t. ‘solid theory’, or ‘quackery’?).

In Turing’s seminal work (Turing, 1937) defining his eponymous machines, Turing also proved that Turing machine computability is equivalent to definability or reduction in the *pure lambda calculus*. Arguably this paper removed the human operator from the computation itself. Hence the Turing machine models a purely mechanical process. Turing later explored extensions to computation and definability using ‘oracles’ (Turing, 1939), which are machines or new logical means able to decide the halting problem for the original formal computational system. In particular, Gödel’s incompleteness proof gives a means of extending any system L logic sufficiently powerful to include arithmetic, and there is of course no end to such extensions. In Gödel’s well-known paper, the formula which states it is not provable can be used to extend any consistent logic, and such a formula can be constructed using Gödel’s arithmetisation technique; (there are also infinitely many other ways to extend an incomplete logic, namely by adding any unprovable sentence as a new axiom).

Arising out of this came the Church-Turing thesis (Kleene, 1943) which is primarily a theory about the real world rather than about formal mathematics, and the more refined definitions by Gandy (Gandy, 1980) which attempt to capture essential properties of a physically realisable computer. These restrictions were however challenged by several authors proposing ways that a physical device could in fact compute the Turing machine halting function. Such devices have

become known as ‘hypercomputers’, given that the more obvious nomenclature ‘supercomputer’ has already been taken to denote a more powerful version of a standard computer. Note that already in the year 1959, long before the ‘hypercomputing’ discourse arose, the almost-forgotten Lázló Kalmár had raised *an argument against the plausibility of Church’s thesis* from a *mathematical* point of view (Szabó, 2018). More recently, the formulation of the Church-Turing thesis was also criticised by Doukas Kapantaïs who claimed that there would exist a ‘human’ manner of calculating values of the notorious *Ackermann* function that cannot be ‘mimicked’ by the operations of Turing’s apparatus (Kapantaïs, 2016). However, whereas arguments such as Kalmár’s or Kapantaïs’ are mathematical-philosophical, the later hypercomputer proposals depend on aspects of *physics* which are currently not certain, in particular on whether the universe is infinite in some way (Kieu, 2004)—is space infinitely divisible, or time?—suggesting that Gandy’s assumptions might not hold. These proposals use features of theories such as general relativity (Pitowsky, 1990), continuous physics (Davies, 2001; Longo, 2003; Stannett, 1990), and/or quantum theory (Deutsch, 1985). These theories might be mutually incompatible, but nonetheless if sufficient features of just one of them are valid in the real world, then it seems that hypercomputation would be possible. Note that, of course, our understanding of physics remains incomplete; just as it has changed over the last 150 years, so too should we expect it to change further in the future. It is indeed not known whether real numbers actually exist in the real world or whether, as Landauer suggests (Landauer, 1996), the real world merely approximates these through its computational information processing. Indeed, Landauer argued that “*algorithms are significant only to the extent they are executable in the real world*” and that “*the real world is unlikely to supply us with unlimited memory or unlimited Turing machine tape*”, and hence that there is “*limited precision available for the execution of algorithms*”. According to Heisenberg, no physical measurement can be done with arbitrarily high precision. However, it would be unwise to infer from this deficiency that the universe itself does not afford such precision. Beyond physics is the question of *engineering* which has received comparatively *little attention in the hypercomputation disputes* seen so far: even if a design is theoretically feasible, it is of little interest if the hypercomputer cannot be built or requires excessive resources to run. It remains unclear whether any of the proposed hypercomputers is feasible to construct in future. A final concern remains our own ability to perceive the results of hypercomputation: what is the value of infinite precision computation if we ourselves can only sample a finite number of bits from the output?

One reason for the recent interest in hypercomputation stems from considerations by philosophers and cognitive scientists concerning the nature of *consciousness*. Searle has famously argued using his ‘Chinese Room’ argument (Searle, 1990), which was recently extended to more advanced computational devices (Nasuto, Bishop, Roesch, & Spencer, 2015). In a similar context, Penrose has identified four mind-philosophical positions ranging from (A) “*all thinking is computation*”, (B) “*awareness is a feature of the brain’s action*” which does not arise when the brain is simulated computationally, (C) the physical action of the brain that evokes awareness cannot be simulated computationally, to (D) “*awareness cannot be explained in physical, computational or any other scientific terms*” (Penrose, 1997). Following on from this it has been suggested that if the brain is more than a computer, then it is—perhaps—a hypercomputer (Copeland & Sylvan,

1999). In those discourses the topic of *hypercomputation* is used for trying to bridge the gap between *philosophy of nature* (physics) and *philosophy of mind* (psychology) which—in spite of many attempts—was never convincingly bridged since the days of Driesch (Driesch, 1923; Gruner, 2017).

3 OUR NOTION OF ‘COMPUTATION’

With the recapitulation of some relevant literature in the foregoing section we have prepared a frame of reference into which we can now position our own ideas about the notions of ‘computer’ and ‘computation’.

- We want to be able to distinguish clearly between processes which are ‘computations’ and other processes which are *not* ‘computations’, as well as between things or entities or devices which are ‘computers’ and other things or entities or devices which are *not* computers. This epistemological requirement *forbids* any extremist pancomputationalism in which *everything* (or the entire universe) is somehow a ‘computer’—otherwise the *word* ‘computer’ would merely become a fancy synonym of the word ‘everything’; for comparison see (MacLennan, 2007, sub-sect. 4.4).
- We do *not* dogmatically restrict our notion of ‘computation’ to the currently predominant idea of *discrete* (binary-digital) computation. Our notion of ‘computation’ ought to remain ‘semantically open’ for *analog* (MacLennan, 2007) as well as hybrid (analog-digital) implementations (Ambaram, Beggs, Costa, Poças, & Tucker, 2017), whereby the ‘classical’ *theories of ‘computability’, ‘programmability’ and ‘computational complexity’* might have to be reconsidered and re-researched for those ‘other’ implementational possibilities (Beggs, Costa, Poças, & Tucker, 2014; Beggs, Costa, & Tucker, 2014).
- If we accept with Landauer that ‘information’ is not only physical (Landauer et al., 1991; Landauer, 1996) but even *inevitably* physical (Landauer, 2002) (or, more precisely: that information must necessarily possess some physical substrate in forms of matter and/or energy), then computation as the processing of information is limited in its possibilities by the laws of nature (as studied in the natural sciences, particularly physics and chemistry). In other words: there cannot, by definition, be any super-natural ‘computation’.
- Further we require that ‘we’ (or any other competent observer) can distinguish clearly between some state of the world *before* and some different state of the world *after* a computation. In other words, our notion of ‘computation’ *semantically implies* the notions of ‘time’, ‘order’, and ‘history’. This, in turn, requires ontically the availability of some kind of memory without which we could not remember any before-state against which any after-state could be distinguished. Such a notion of ‘computation’ prevents us, for example, from dubbing an ideal gas in a bottle at constant temperature and constant pressure a ‘computer’ because ‘inside’ such an ideal gas (at its maximal entropy) we cannot identify any ‘history’

in terms of clearly distinguishable before- and after-states. Moreover, it is the observer who must possess such memory which enables the comparative distinction between those before- and after-states of computation. Hence there is *no computation without observer*, (which does, at this point, however, *not* preclude non-human ‘observers’).

- From a *pragmatic perspective* (beyond the realm of the pure natural sciences) we relate the afore-mentioned pre- and after-states of a computation to ‘input’ and ‘output’ (I/O), and we require that for any competent (educated) observer the *input and output must be understandable*⁵, in the *hermeneutic*-philosophical sense of the term ‘understanding’ (*verstehen*) (Penrose, 1997; Schwandt, 1999), along the lines of the tradition of the humanities (*Geisteswissenschaften*) since the 19th century (Schnädelbach, 1984, chpt. 4). This notion of ‘computation’ does not allow us to speak about ‘computation’ in any case of observable natural processes that yield after-states which are *in principle* incomprehensible for any- and everybody (i.e., not merely incomprehensible for un-educated non-experts)⁶.
- Also from a pragmatic perspective we require that any process of ‘computation’ must be *reliable*—or, expressed from a pure nature-scientific perspective: ‘repeatable’, such that the same input will predictably yield the same output under the same external environmental constraints and conditions. If ‘1+1’ is computed by any physical means as ‘= 2’ today, then we demand that the same physical means shall also yield ‘= 2’ tomorrow and next week and next year, etc., as long as our environment, into which all these activities embedded, has itself remained sufficiently ‘stable’. This requirement, however, is *not* intended to absolutely preclude ‘stochastic’ computations in a parochial attitude of pedantic determinism. However, even stochastic processes can be ‘reliable on the meta-level’ (Beggs, Cortez, Costa, & Tucker, 2016), whereby the distribution of random-outputs is known (e.g., a Gaussian bell-curve). Without this pragmatic-epistemic requirement we would not only not be able to distinguish generally between computers and non-computers in principle any more—we could not even distinguish ‘mal-functioning’ from ‘well-functioning’ devices (Floridi, Fresco, & Primiero, 2015; Fresco & Primiero, 2013) in our most mundane daily situations any more.
- Closely related to the above-mentioned reliability of a computational process (when beheld as a ‘black box’) is its internal *explainability* (when beheld as a ‘glass box’). The notion of ‘explanation’ at this point is as in Dilthey’s and Windelband’s classical distinctions between

⁵Some readers might perhaps remember the perplexing answer ‘42’ which the artificial super-intelligence ‘Deep Thought’ gave, after more than 7 million years of computation time, to ‘the ultimate question of life, the universe, and everything’ in a popular science fiction parody by Douglas Adams, whereby nobody could actually grasp the very question to which ‘42’ was the computed ‘answer’ (Adams, 1979). We may speculate whether Douglas Adams, when writing his parody novel, might perhaps have been aware of some esoteric Kabbalistic mysticism in which ‘42’ symbolises an unfathomable divine principle of cosmic creation.

⁶To Einstein’s well-known aphorism, according to which God “does not play dice”, we might thus add that (at least from our limited human vantage point) *God also does not compute*—if we want to accept the Biblical wisdom that “God’s ways are unfathomable” [Rom.11:33].

erklären versus *verstehen*, respectively *nomothetic* versus *idiographic* (Windelband, 1998)—i.e., a kind of Hempel-Oppenheim (or similar notion) of ‘lawful comprehension’ (Bunge, 1998, chpt. 9) which also includes *predictability*. This requirement stipulates that computational processes must be *amenable to proper scientific inquiry*—an argument that was raised some years ago against the concept of the ‘Turing Test’ (Kary & Mahner, 2002).

- W.r.t. what has traditionally been called ‘programming’ (by means of ‘software’) and what *distinguishes a genuine computer from a mere calculator*, we request that the above-mentioned reliable and explainable computational process must be effectively and purposefully ‘manipulatable’ (i.e., computation *by design*). ‘Purposeful’ is (once again) a pragmatic aspect of computation (outside the realm of ‘pure nature’), whilst ‘effective’ implies ‘with *finite resources*’ of matter, energy, and time. This finiteness requirement must not be [mis]understood as dogmatically precluding non-terminating ‘loops’ in computations—it only precludes the infeasible engineering (construction) of any infinitely large apparatus or the composition of ‘programs’ with infinitely large Kolmogorov complexity. This ‘finitism of the means’, which we postulated from a perspective of engineering (design and construction) as we know it, implies some careful skepticism w.r.t. the actual implementability of some recently published ‘science fiction’ about some hyper-computational devices hovering in an infinitesimally small ϵ -distance above the Schwarzschild horizon of rotating black holes (Durand-Lose, 2004, 2012; Németi & Andréka, 2006a, 2006b)⁷, and the like. At this point, however, somebody might perhaps raise the critical question whether (for example) a farmer could ‘program’ his crop field by ‘purposefully and effectively manipulating’ the growth process of his plants with the addition of fertiliser? A non-traditional and un-dogmatic answer to this vexing question might *perhaps* be ‘yes’ (i.e., the farmer’s plantation field *can* be ‘programmed’ by means of fertiliser), *if* it is anyhow possible to construe a hermeneutically plausible and intersubjectively acceptable *interpretation* of the seeds in the soil as ‘input’ and the harvested crops as ‘output’ (see above: hermeneutics of I/O) *without* merely playing an arbitrary private *Sprachspiel* in an arbitrary private language (Wittgenstein, 1953) in which seeds just so happen to be called ‘input’ and harvested crops ‘output’. In any case the practical applicability of such a fertiliser-‘programmed’ soil-‘computer’ would appear as disappointingly limited when compared to the huge number of application possibilities already offered to us even by the most un-fancy, conventional, traditional, classical electronic non-hyper von-Neumann/Zuse devices; for comparison see (MacLennan, 2007, sub-sect. 4.4) again.
- Though we have demanded that I/O of computations must be ‘hermeneutically meaningful’ (see above), we do *not* dogmatically define ‘computation’ purely semiotically (semilogically) as ‘symbol manipulation’ (Stender & Addis, 1990; Harnad, 1994; Kolers &

⁷A large-scale online discussion on this topic was recently (2018) launched on the web pages of <https://www.researchgate.net/>—however, after many critical replies by various scientists (including mathematicians and astro-physicists) had been posted, this online discussion topic suddenly vanished from the ResearchGate pages and was no longer reachable since then.

Smythe, 1984), because a computational process might well pass through various non-symbolic intermediate physical micro-states (which might not each and all be ‘meaningful’ per-se). Indeed, whether or not some observable material-energetic phenomenon P carries any ‘flavour’ of symbolicity is already a question of cultural hermeneutics: beheld from a purely Husserlian-‘phenomenological’ perspective (with ‘epoché’-abstaining from judgments) (Bochenski, 1954, part II) there is nothing inherent in P which could ‘tell’ us that P possesses any ‘symbolic power’ in and by itself. This question concerning the ‘symbolicity’ of natural phenomena arises especially in the field of DNA computing wherein complex bio-chemical molecule structures are abstractly interpreted as ‘texts’ composed of discrete ‘characters’ (Paun et al., 2005).

- Unlike purpose-less natural processes, which ‘just so happen’, we want to speak of ‘computation’ where the entire ‘happening’ $I \rightarrow P \rightarrow O$ from (hermeneutically meaningful) ‘input’ to (hermeneutically meaningful) output fulfills a *recognisable purpose*. ‘Recognisable’ means: from our own human vantage point (not from any mysterious divine vantage point which we humans cannot reach). Any kind of computer-‘thing’ which *implements* (enables, facilitates) such a recognisably purposeful ‘happening’ has thus the status of *Zeug* (equipment) in Heidegger’s hermeneutic fundamental-ontology (Heidegger, 1927). Whether or not ‘pure nature’ outside the human realm actually has (or can possibly have) any purposes is philosophically debatable (albeit not in the centre of the focus of this paper): whilst Aristotle distinguished and accepted the well-known 4 ‘classical’ notions of causation (*materialis, formalis, efficiens, finalis*), high-modern philosophers of nature continued to accept only the *efficiens* as ‘causa’ (thereby more or less equating ‘causation’ with no ‘measurable transmission of energy’), whereas late-modern philosophers of nature (especially philosophers of biology) have raised the question of natural *teleology* anew.
- The above-mentioned identifiable purpose of $I \rightarrow P \rightarrow O$ (which gives a genuine computer the fundamental-ontological status of *Zeug*) is closely related to a user’s *epistemic desire* for the output which a prospective or ongoing computation will or shall ‘hopefully’ yield. Such epistemic desire ‘felt’ by a user can only be fulfilled if a computation’s output, when it eventually appears, can genuinely *inform* the user in a non-trivial manner (Oberholzer & Gruner, 2016). The ‘epistemic desire’ (Piccinini, 2011) of the user is directed towards what is not yet known to him, or, in other words: the ‘amount of surprise’ which the user can ‘feel’ at the arrival of a computation’s output. This requirement, which is conceptually related to Shannon’s notion of information *quantity*, is intended to *prevent triviality* in our notion of ‘computation’, (such as in our artificially construed ‘soil computer’ example of above which ‘outputs’ a harvest for the farmer); for comparison see (MacLennan, 2007, sub-sect. 4.4) once more. On the other hand, this ‘principle of surprise’ (according to which no rational-reasonable user would feel any epistemic desire to ‘compute’ what is already known) brings ‘computations’ into some peculiar epistemological and methodological proximity to explorative scientific ‘experiments’ (Bunge, 1998, chpt. 14), because also scientific experiments are not conducted in situations of zero epistemic desire where everything is already known:

see (Beggs, Costa, & Tucker, 2012a, 2012b; Beggs & Tucker, 2007b) for comparison.

4 DISCUSSION AND ‘NON-CONCLUSION’

Current discussions about hyper-computers, and the nature of consciousness, awareness or mind seem to be grounded in the ‘real world’. By contrast, mathematical notions of computing exist in the domain of mathematics, which may instead be ‘platonic’.⁸ Here we encounter an interesting tension between Landauer (Penrose, 1997) who does not believe that any real number can be computed at all (thereby also quoting John Wheeler’s dictum *no continuum*) versus what children are taught in school, namely that real numbers are the way to measure quantities in the real world (and hence that π exists as a ratio of two real world measurements). By contrast, mathematicians developed the notion of ‘converging sequence’ and are thus able to define real numbers like π using infinite summations. Then, via the development of set theory, they came to realise some challenges such as the fact that the real numbers are uncountable, and introduced the still controversial ‘continuum hypothesis’. Finally, Turing distinguished between some real numbers which are ‘computable’ and others which are not—a distinction which rests on the definition of the actual *machine* (or class of machines) which is capable of outputting them in a format a human being can perceive. These different perspectives seem to encapsulate much of the arguments and issues inherent in the notion of ‘computing/computer’ and the relation of this notion with the notion of ‘infinity’ (∞). These points are considered further in the discussion that follows.

Computer scientists are familiar with *regular expressions* and *context-free grammars* which correspond to the first two levels of the Chomsky hierarchy. These are directly associated with finite-state and stack automata as the theoretical symbol processing machines able to recognise the ‘words’ of regular and context-free languages (Anderson, 2006). The third and fourth level of the Chomsky hierarchy contain the context-sensitive and recursively enumerable languages, associated with linear bounded and full Turing machines respectively. These mathematical automata can well be related to information processing devices in the real world. Although our current understanding is that the full power of a Turing machine is required in order to qualify as a ‘computer’, *less* powerful automata *also* have their practical uses. For example: Most *embedded systems* are finite state systems, albeit ‘transducers’ rather than ‘recognisers’ of ‘words’. The microprocessor inside a digitally controlled washing machine or microwave oven has a fixed and finite memory, corresponding to the first level of the Chomsky hierarchy (known as level 0). Rational numbers such as $\frac{1}{3}$ have a decimal representation that can be described approximately using a regular language and recognised by a finite state machine. An example of a language at level 1 of the Chomsky hierarchy would be bracket-free expressions which can be recognised and processed by a stack automaton or transducer. Beyond this, software of high practical utility, such as a weather simulator and forecaster, is merely at Chomsky level 2 because the size of memory used is in proportion to the size of the system being simulated. Yet these programs are routinely run on

⁸Above we put ‘real world’ and ‘platonic’ into quotation marks because when beheld from the opposite—Plato’s—point of view the real world appears merely as ‘shadow’ whilst the ideas—and only they—are regarded as fully ‘real’.

super-computers because an ordinary desktop computer would be unable to run the program and deliver the forecast quickly enough. It is indeed somewhat ironic that a machine with a grandiose title like ‘super-computer’ can be considered, from a meta-mathematical perspective, as inferior to a Turing machine with its humble tape.

The so-called ‘hyper-computers’ proposed in the literature would be able to *solve* the (Turing machine) *halting problem*, and would therefore be strictly more powerful than Turing machines. However, from a meta-mathematical perspective, there is an *infinite hierarchy* of super-Turing machines. For each level, adding or postulating a solution to its own halting problem yields a strictly more powerful machine. It is reasonable to ask therefore what advantage there is to being able to build a hyper-computer and thus to move just one step up in this infinite staircase. Would being able to solve the (Turing machine) halting problem have any great value? In practice, if a program does not halt after a reasonable period of time, such as an hour or a day, we will stop it and re-write the program, perhaps to solve a slightly different problem or to return an approximate solution. It is therefore *not clear what type of problems the hyper-computer could usefully solve* for us. Admittedly, such a machine would have the power to confirm some interesting theorems in arithmetics, such as the commutativity of addition or, more controversially, Goodstein’s theorem. There is, however, little room for doubt among mathematicians about these theorems, such that the hyper-computer would merely re-confirm what we already know. There would be a benefit if we could observe its workings so that we could check or at least survey these infinitely many confirming calculations, but that would require us to have infinite perception.

Speculatively, there might be an advantage in *using a hyper-computer to enhance or augment our senses*. If we can use an infinite computer to observe infinity then that might give us a kind of infinite perception. This thought-experiment raises an interesting and related question: *when is the combination of a human being and a computer more powerful than either, or a different thing altogether?*—see (Beggs et al., 2012a) for comparison. Anyway, it is notable that speculation about the power of computers and computation often makes *assumptions about finiteness*. Turing assumed that human memory is finite (Turing, 1937). Landauer stated that in a “*world with limited memory we simply cannot distinguish between π and a terribly close neighbour*” (Penrose, 1997). An analog process—such as drawing a circle, measuring the diameter and circumference, then taking their ratio—can only ever approximate π and even repeating this as often as realistically possible will only give us a slightly better approximation. A digital computer, by contrast, can calculate any digit or finite set of digits of π we can realistically ask for, assuming we are prepared to supply enough physical resources and wait for the answer. A hyper-computer with infinite memory could compute all digits of π in a finite time, but we would still only be able to access a finite set of those digits with our limited senses. If somehow we could be enhanced with infinite perception, then we might be able to perceive all those digits of π in a finite period of time. Alternatively, if we were granted infinite memory and infinite time, then we could still perceive all these digits eventually even if our perception remained finite. Similar considerations seem to motivate some of the recent ‘black hole machine’ speculations (Németi & Andr eka, 2006a, 2006b) which also aim at ‘compressing’ the infinity until it becomes amenable to finite perception.

Mathematicians however assume that π is well-defined, and that real numbers ‘exist’ (some-

where, even if not in our real world). A rational number such as $\frac{1}{3}$ is hardly controversial, however, being the ratio of two integer quantities—its decimal expansion can be output by a finite state transducer. By contrast, π is either defined to be the ratio of two physical measurements, the circumference and radius of a circle, which in practice we are only able to approximate. Alternatively π is considered to be the limit of an infinite summation, hence the result of a computation (without time-limit), hence Turing-computable (without time-limit). The situation is somewhat worse with a *Chaitin constant* or halting probability.⁹ As Chaitin has shown in (Chaitin, 2006), no matter how large a number (bits) of this we are able to discover, there are still infinitely many mathematical truths that cannot be proved. There may be limited benefit, therefore, in creating a hyper-computer which only allows us to determine this constant one bit at a time. As Turing has shown in (Turing, 1939), mathematical knowledge can be extended indefinitely: in the context of Turing machines, simply add another oracle machine. It is interesting to consider the *difference between an oracle*, for example in the classical world, *and a forecast*. These days we expect rich graphics and animations to explain why we should expect rain at 4pm tomorrow. By contrast, a classical oracle such as the one at Delphi was mysterious in its workings, and could be ambiguous in its predictions. With our current notion of a ‘computer’, it has a program whose behaviour can be explained by its author[s] and other programming experts. Moreover, popular programs can have extremely large numbers of users so that those programs’ behaviours are commonly experienced and predictable even by non-programmers. Even if a hyper-computer is feasible, however, it may be very expensive to construct, or to run, or both. In this case, very few people will ever be able to use the hyper-computer and it may have limited impact on most people’s lives, much like an early supercomputer. Most people will tend to identify the notion of a computer with the devices they are familiar with, much as we think of a bus station or a train station rather than a space station when asked to describe what a station is. Given the comments earlier about π , Chaitin’s constant, and oracles, a *hyper-computer could appear as mysterious and untrustworthy* compared to current day computers.¹⁰ Indeed, it may be that we would also need infinite perception, or infinite memory and time, to gain the full benefit of an infinite memory computer.

While speculating about the development of super-human powers to allow us to benefit from hyper-computation, it is worth observing that such powers could allow us to go beyond the limits of Gödel’s incompleteness theorem. In most meta-mathematics it is assumed that logical statements are written using a finite alphabet of symbols, and that the number of axioms and inference rules is also finite. If it is possible to use an infinite alphabet and/or an infinite number of axioms—for comparison see (Gasquet, Goranko, & Schwarzentruher, 2016), wherein axioms can be parameterised by real numbers—then a formal system, even one for the reals, could be defined by ‘listing’ all facts. This means that no inference rules would be needed, and the logic would be complete. Of course, if no inference rules are needed, then the idea of a ‘decision procedure’, and a computational device to carry it out, would become redundant altogether, and the epistemic

⁹The exact value of this probability depends on the program encoding being used, so there is more than one.

¹⁰Such a machine would thus no longer possess the status of ‘device’ in the philosophy of engineering (Maibaum, 2014).

distinction between ‘computing’ and ‘not-computing’ would become obsolete.

At this point, where we would normally ‘conclude’ a paper, we end it in this particular case with a ‘non-conclusion’ in order to emphasise once again the *historic-semantic openness* of all theoretical, philosophical and scientific notions, including the notions of ‘computer’ and ‘computing’, in the long-term history of ideas (Fleck, 1935; Hübner, 1983; Kuhn, 1962; K. Popper, 1944). Hence we cannot reasonably aim at ‘freezing’ the meaning of the words ‘computer’ and ‘computing’ once and forever. However, for our own time of nowadays, we are well able to point out the various ‘degrees’ of ‘appropriateness’ (or inappropriateness) which various (mutually competing) notions of ‘computation’ can have *in relation to* what we currently know (or believe to know) about our external physical nature as well as about ourselves as ‘soulful’ (i.e., purpose-driven) human beings who are—one the one hand—*part of* this external physical nature but who also—on the other hand—*transcend* this external physical nature without being able to be completely ‘reduced into’ it (in a way in which naive 19th century ontologies of mechanism / physicalism / materialism wanted to have it.) Such ‘pointing-out’ was the intended purpose—and hopefully also the actual achievement—of this paper.

References

- Acar, E. (2012). *Models of hypercomputation* (Master’s thesis, Fakultät für Informatik der Technischen Universität Wien, Vienna, Austria).
- Adams, D. (1979). *The Hitchhiker’s Guide to the Galaxy*. Pan.
- Ambaram, T., Beggs, E., Costa, J. F., Poças, D., & Tucker, J. V. (2017). An analogue-digital model of computation: Turing machines with physical oracles. In *Advances in unconventional computing* (pp. 73–115). Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-33924-5_4
- Anderson, J. A. (2006). *Automata theory with modern applications*. Cambridge University Press. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-33924-5_4
- Andréka, H., Németi, I., & Németi, P. (2009). General relativistic hypercomputing and foundation of mathematics. *Natural Computing*, 8(3), 499–516. Retrieved from <https://link.springer.com/article/10.1007/s11047-009-9114-3>
- Baeten, J. C., Luttik, B., & Van Tilburg, P. (2013). Reactive Turing machines. *Information and Computation*, 231, 143–166. Retrieved from <https://dl.acm.org/citation.cfm?id=2542906>
- Bauer, F. (1972). From scientific computing to computer science. *The Skyline of Information Processing—Proceedings 10th Anniversary Celebration of the IFIP*, 57–71.
- Beggs, E., Cortez, P., Costa, J. F., & Tucker, J. V. (2016). Classifying the computational power of stochastic physical oracles. Retrieved from https://www.researchgate.net/publication/305709935_Classifying_the_computational_power_of_stochastic_physical_oracles
- Beggs, E., Costa, J. F., Poças, D., & Tucker, J. V. (2014). An analogue-digital Church-Turing thesis. *International Journal of Foundations of Computer Science*, 25(04), 373–389. Retrieved from <https://www.worldscientific.com/doi/abs/10.1142/S0129054114400012>

- Beggs, E., Costa, J. F., & Tucker, J. V. (2012a). Axiomatizing physical experiments as oracles to algorithms. *Phil. Trans. R. Soc. A*, 370(1971), 3359–3384. Retrieved from <https://www.jstor.org/stable/23250193>
- Beggs, E., Costa, J. F., & Tucker, J. V. (2012b). Unifying science through computation: Reflections on computability and physics. In *Special sciences and the unity of science* (pp. 53–80). Springer. Retrieved from citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.153.6854
- Beggs, E., Costa, J. F., & Tucker, J. V. (2014). Three forms of physical measurement and their computability. *The Review of Symbolic Logic*, 7(4), 618–646.
- Beggs, E. & Tucker, J. V. (2006). Embedding infinitely parallel computation in Newtonian kinematics. *Applied mathematics and computation*, 178(1), 25–43. Retrieved from <https://www.sciencedirect.com/science/article/abs/pii/S0096300305008313>
- Beggs, E. & Tucker, J. V. (2007a). Can Newtonian systems, bounded in space, time, mass and energy compute all functions? *Theoretical Computer Science*, 371(1-2), 4–19. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0304397506007481>
- Beggs, E. & Tucker, J. V. (2007b). Experimental computation of real numbers by Newtonian machines. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* (Vol. 463, pp. 1541–1561). The Royal Society. Retrieved from <http://rspa.royalsocietypublishing.org/content/463/2082/1541>
- Blakey, E. (2011). Computational complexity in non-Turing models of computation: The what, the why and the how. *Electronic Notes in Theoretical Computer Science*, 270(1), 17–28. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1571066111000041>
- Bochenski, I. M. (1954). *Die zeitgenössischen Denkmethode*. Francke Publ.
- Bournez, O. & Campagnolo, M. L. (2008). A survey on continuous time computations. In *New computational paradigms* (pp. 383–423). Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-0-387-68546-5_17
- Bozşahin, C. (2016). What is a computational constraint? In *Computing and philosophy* (pp. 3–16). Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-23291-1_1
- Bringsjord, S. & Zenzen, M. (2002). Toward a formal philosophy of hypercomputation. *Minds and Machines*, 12(2), 241–258. Retrieved from <https://link.springer.com/article/10.1023/A:1015651216328>
- Bunge, M. (1998). *Philosophy of Science, vol. 2: From Explanation to Justification* (Revised). Transaction Publ.
- Chaitin, G. (2006). The limits of reason. *Scientific American*, 294(3), 74–81.
- Cockshott, P. (2015). *No mysteries*. [10.13140/RG.2.1.1510.2567](https://doi.org/10.13140/RG.2.1.1510.2567)
- Cockshott, P., Mackenzie, L., & Michaelson, G. (2008). Physical constraints on hypercomputation. *Theoretical Computer Science*, 394(3), 159–174. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0304397507008778>
- Cockshott, P., Mackenzie, L., & Michaelson, G. (2010). Non-classical computing: Feasible versus infeasible. In *Proceedings of the 2010 ACM-BCS Visions of Computer Science Conference*

- (p. 4). British Computer Society. Retrieved from <https://dl.acm.org/citation.cfm?id=1811188>
- Cockshott, P. & Michaelson, G. (2007). Are there new models of computation? Reply to Wegner and Eberbach. *The Computer Journal*, 50(2), 232–247. Retrieved from <https://ieeexplore.ieee.org/document/11089503/>
- Copeland, B. J. & Sylvan, R. (1999). Beyond the universal Turing machine. *Australasian Journal of Philosophy*, 77(1), 46–66. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.1790>
- Costa, J. F. (2017). The power of analogue-digital machines. In International Conference on Unconventional Computation and Natural Computation (pp. 3–7). Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-58187-3_1
- Cotogno, P. (2009). A brief critique of pure hypercomputation. *Minds and Machines*, 19(3), 391. Retrieved from <https://link.springer.com/article/10.1007/s11023-009-9161-7>
- da Costa, N. C. & Doria, F. A. (2006). Some thoughts on hypercomputation. *Applied mathematics and computation*, 178(1), 83–92. Retrieved from <https://www.sciencedirect.com/science/article/abs/pii/S0096300305008362>
- Davies, E. B. (2001). Building infinite machines. *The British Journal for the Philosophy of Science*, 52(4), 671–682. Retrieved from <https://www.jstor.org/stable/3541913>
- Deutsch, D. (1985). Quantum theory, the Church–Turing principle and the universal quantum computer. *Proc. R. Soc. Lond. A*, 400(1818), 97–117.
- Driesch, H. (1923). *Leib and Seele: eine Untersuchung über das psychophysische Grundproblem*. E. Reinicke.
- Durand-Lose, J. (2004). Abstract geometrical computation for black hole computation. In International Conference on Machines, Computations, and Universality (pp. 176–187). Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-540-31834-7_14
- Durand-Lose, J. (2012). Abstract geometrical computation 6: A reversible, conservative and rational based model for black hole computation. *International Journal of Unconventional Computing*, 8(1), 33–46.
- Fleck, L. (1935). *Entstehung und Entwicklung einer wissenschaftlichen Tatsache: Einführung in die Lehre vom Denkstil und Denkkollektiv*. Benno Schwabe & Co. Publ.
- Floridi, L., Fresco, N., & Primiero, G. (2015). On malfunctioning software. *Synthese*, 192(4), 1199–1220. Retrieved from <https://link.springer.com/article/10.1007/s11229-014-0610-3>
- Franchette, F. (2015). Oracles in the hypercomputation debate. *IJUC*, 11(3-4), 299–323.
- Fresco, N. & Primiero, G. (2013). Miscomputation. *Philosophy & Technology*, 26(3), 253–272.
- Gandy, R. (1980). Church's thesis and principles for mechanisms. In *Studies in logic and the foundations of mathematics* (Vol. 101, pp. 123–148). Elsevier. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0049237X08712576>
- Gasquet, O., Goranko, V., & Schwarzentruher, F. (2016). Big Brother Logic: Visual-epistemic reasoning in stationary multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 30(5), 793–825. Retrieved from <https://link.springer.com/article/10.1007/s10458-015-9306-4>

- Gheorghe, M. & Stannett, M. (2012). Membrane system models for super-Turing paradigms. *Natural computing*, 11(2), 253–259. Retrieved from <https://link.springer.com/article/10.1007/s11047-011-9291-8>
- Goldin, D. & Wegner, P. (2008). The interactive nature of computing: Refuting the strong Church–Turing thesis. *Minds and Machines*, 18(1), 17–38. Retrieved from <https://link.springer.com/article/10.1007/s11023-007-9083-1>
- Gruner, S. (2017). Hans Driesch re-visited after a Century: on ‘Leib und Seele—Eine Untersuchung über das psychophysische Grundproblem’. *Cosmos and History*, 13(3), 401–424.
- Hagar, A. & Korolev, A. (2007). Quantum hypercomputation: Hype or computation? *Philosophy of Science*, 74(3), 347–363. Retrieved from <https://www.jstor.org/stable/10.1086/521969>
- Hamkins, J. D. (2002). Infinite time Turing machines. *Minds and Machines*, 12(4), 521–539. Retrieved from <https://link.springer.com/content/pdf/10.1023/A:1021180801870.pdf>
- Hamkins, J. D. & Lewis, A. (2000). Infinite time Turing machines. *The Journal of Symbolic Logic*, 65(2), 567–604.
- Harnad, S. (1994). Computation is just interpretable symbol manipulation; Cognition isn't. *Minds and Machines*, 4(4), 379–390. Retrieved from <https://link.springer.com/article/10.1007/BF00974165>
- Heidegger, M. (1927). *Sein und Zeit*. Max Niemeyer Publ.
- Hübner, K. (1983). *Critique of scientific reason*. University of Chicago Press.
- Kapantais, D. (2016). A refutation of the Church-Turing thesis according to some interpretation of what the thesis says. In *Computing and philosophy* (pp. 45–62). Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-23291-1_4
- Kary, M. & Mahner, M. (2002). How would you know if you synthesized a Thinking Thing? *Minds and Machines*, 12(1), 61–86. Retrieved from <https://link.springer.com/article/10.1023/A:1013738831686>
- Kieu, T. D. (2004). Finiteness of the universe and computation beyond Turing computability. *arXiv preprint quant-ph/0403045*.
- Kieu, T. D. (2006). Reply to Andrew Hodges. *arXiv preprint quant-ph/0602214*.
- Kleene, S. C. (1943). Recursive predicates and quantifiers. *Transactions of the American Mathematical Society*, 53(1), 41–73. Retrieved from <http://www.ams.org/journals/tran/1943-053-01/S0002-9947-1943-0007371-8/>
- Kolers, P. A. & Smythe, W. E. (1984). Symbol manipulation: Alternatives to the computational view of mind. *Journal of Verbal Learning and Verbal Behavior*, 23(3), 289–314. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0022537184901828>
- Kuhn, T. (1962). The structure of scientific revolutions. *University of Chicago Press*.
- Landauer, R. et al. (1991). Information is physical. *Physics Today*, 44(5), 23–29.
- Landauer, R. (1996). The physical nature of information. *Physics letters A*, 217(4-5), 188–193. Retrieved from <https://www.sciencedirect.com/science/article/pii/0375960196004537>

- Landauer, R. (2002). Information is inevitably physical. In *Feynman and computation* (pp. 77–92). Westview Press. Retrieved from <https://dl.acm.org/citation.cfm?id=305684>
- Li, Y. (2008). Fuzzy Turing machines: Variants and universality. *IEEE Transactions on Fuzzy Systems*, 16(6), 1491–1502. Retrieved from <https://ieeexplore.ieee.org/document/4607247/>
- Loff, B. & Costa, J. F. (2009). Five views of hypercomputation. *IJUC*, 5(3-4), 193–207.
- Longo, G. (2003). Computer modelling and natural phenomena. In ACM SIGSOFT Software Engineering Notes (Vol. 28, pp. 1–5). ACM.
- Luttik, B. & Yang, F. (2014). Executable behaviour and the pi-Calculus. arXiv preprint arXiv:1410.4512.
- Luttik, B. & Yang, F. (2016). Reactive Turing machines with infinite alphabets. *arXiv preprint arXiv:1610.06552*.
- MacLennan, B. J. (2007). A review of analog computing. *Department of Electrical Engineering & Computer Science, University of Tennessee, Technical Report UT-CS-07-601 (September)*.
- MacLennan, B. J. (2009). Super-Turing or non-Turing? Extending the concept of computation. *International Journal of Unconventional Computing*, 5(3-4), 369–387. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.76.433>
- Maibaum, T. (2014). What is a BOK? *Lecture Notes in Computer Science*, 8368, 184–188.
- Maldonado, C. E., Cruz, G., & Nelson, A. (2015). Biological hypercomputation: A new research problem in complexity theory. *Complexity*, 20(4), 8–18. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1002/cplx.21535>
- Michaelson, G. & Cockshott, P. (2006). Constraints on hypercomputation. In *Conference on Computability in Europe* (pp. 378–387). Springer. Retrieved from https://link.springer.com/chapter/10.1007/11780342_40
- Mounier-Kuhn, P. (2015). Logic, formal linguistics and computing in France: From non-reception to progressive convergence. In *International Conference on History and Philosophy of Computing* (pp. 24–41). Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-47286-7_2
- Müller, V. C. (2011). On the possibilities of hypercomputing supertasks. *Minds and Machines*, 21(1), 83–96. Retrieved from <https://link.springer.com/article/10.1007/s11023-011-9222-6>
- Nasuto, S. J., Bishop, J. M., Roesch, E. B., & Spencer, M. C. (2015). Zombie mouse in a Chinese room. *Philosophy & Technology*, 28(2), 209–223. Retrieved from <https://link.springer.com/article/10.1007/s13347-014-0150-2>
- Nayebi, A. (2014). Practical intractability: A critique of the hypercomputation movement. *Minds and Machines*, 24(3), 275–305. Retrieved from <https://link.springer.com/article/10.1007/s11023-013-9317-3>
- Németi, I. & Andréka, H. (2006a). Can general relativistic computers break the Turing barrier? In *Conference on Computability in Europe* (pp. 398–412). Springer. Retrieved from https://link.springer.com/chapter/10.1007/11780342_42

- Németi, I. & Andréka, H. (2006b). New physics and hypercomputation. In SOFSEM (pp. 63–63). Retrieved from https://link.springer.com/chapter/10.1007/11611257_6
- Oberholzer, F. & Gruner, S. (2016). The notion of 'information': Enlightening or forming? In Preproceedings IACAP16: Annual Meeting of the International Association for Computing and Philosophy, Ferrara.
- Paun, G., Rozenberg, G., & Salomaa, A. (2005). *DNA computing: New computing paradigms*. Springer.
- Penrose, R. (1997). On understanding understanding. *International Studies in the Philosophy of Science*, 11(1), 7–20. Retrieved from <https://www.tandfonline.com/doi/pdf/10.1080/02698599708573547>
- Piccinini, G. (2007). Computing mechanisms. *Philosophy of Science*, 74(4), 501–526. Retrieved from <https://www.journals.uchicago.edu/doi/10.1086/522851>
- Piccinini, G. (2011). The physical Church–Turing thesis: Modest or bold? *The British Journal for the Philosophy of Science*, 62(4), 733–769. Retrieved from <https://academic.oup.com/bjps/article/62/4/733/1419656>
- Pitowsky, I. (1990). The physical Church thesis and physical computational complexity. *Iyyun: The Jerusalem Philosophical Quarterly*, 81–99. Retrieved from <https://www.jstor.org/stable/23350656>
- Popper, K. (1944). The poverty of historicism, I. *Economica*, 11(42), 86–103. Retrieved from <https://www.jstor.org/stable/2549642>
- Popper, K. R. (1935). *Logik der Forschung: zur Erkenntnistheorie der moderner Naturwissenschaft*. Springer.
- Primiero, G. (2016). Information in the philosophy of computer science. *The Routledge handbook of philosophy of information*, 90–106.
- Ramezani, R. (2014). A hypercomputation in Brouwer's constructivism. arXiv preprint arXiv:1408.2930.
- Resconi, G. & Licata, I. (2012). Information as environment changings classical and quantum morphic computation. In *Methods, models, simulations and approaches towards a general theory of change* (pp. 47–81). World Scientific.
- Schiemann, G. (2008). *Natur, Technik, Geist: Kontexte der Natur nach Aristoteles und Descartes in lebensweltlicher und subjektiver Erfahrung*. Walter de Gruyter.
- Schiemann, G. (2014). Die Relevanz nicht-technischer Natur: Aristoteles' Natur-Technik-Differenz in der Moderne. In T. Hartung G. & Kirchhoff (Ed.), *Welche Natur brauchen wir? Analyse einer anthropologischen Grundproblematik des 21. Jahrhunderts* (pp. 67–96). Karl Alber Publ.
- Schnädelbach, H. (1984). *Philosophy in Germany 1831–1933*. Cambridge University Press.
- Schwandt, T. A. (1999). On understanding understanding. *Qualitative inquiry*, 5(4), 451–464. Retrieved from <http://journals.sagepub.com/doi/10.1177/107780049900500401>
- Searle, J. R. (1990). Is the brain's mind a computer program? *Scientific American*, 262(1), 25–31. Retrieved from <https://www.osti.gov/biblio/5305098>

- Shang, Y., Lu, X., & Lu, R. (2012). Turing machines based on unsharp quantum logic. *arXiv preprint arXiv:1210.1097*.
- Shang, Y., Lu, X., & Lu, R. (2015). Computing power of Turing machines in the framework of unsharp quantum logic. *Theoretical Computer Science*, 598, 2–14. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0304397514010147>
- Siegelmann, H. T. & Sontag, E. D. (1994). Analog computation via neural networks. *Theoretical Computer Science*, 131(2), 331–360. Retrieved from <https://www.sciencedirect.com/science/article/pii/0304397594901783>
- Smith, W. D. (2006). Three counterexamples refuting Kieu’s plan for ‘quantum adiabatic hyper-computation’; and some uncomputable quantum mechanical tasks. *Applied Mathematics and Computation*, 178(1), 184–193. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0096300305008507>
- Stannett, M. (1990). X-machines and the halting problem: Building a super-Turing machine. *Formal Aspects of Computing*, 2(1), 331–341. Retrieved from <https://link.springer.com/article/10.1007/BF01888233>
- Stannett, M. (2001). Hypercomputation is experimentally irrefutable. *Sheffield University Department of Computer Science Report TR CS-01-04*.
- Stannett, M. (2012). Membrane systems and hypercomputation. In International Conference on Membrane Computing (pp. 78–87). Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-3-642-36751-9_6
- Stannett, M. (2014). Specification, testing and verification of unconventional computations using generalized X-machines. *International Journal of General Systems*, 43(7), 713–721. Retrieved from <https://www.tandfonline.com/doi/abs/10.1080/03081079.2014.924203>
- Stender, J. & Addis, T. (1990). *Symbols versus neurons?* IOS Press. Retrieved from <https://doi.org/10.1016/j.tics.2013.06.004>
- Syropoulos, A. (2006). Fuzzifying P systems. *The Computer Journal*, 49(5), 619–628. Retrieved from <https://ieeexplore.ieee.org/document/8139208/?denied>
- Syropoulos, A. (2009). Fuzzy chemical abstract machines. *arXiv preprint arXiv:0903.3513*.
- Syropoulos, A. (2016). A (basis for a) philosophy of a theory of fuzzy computation. *arXiv preprint arXiv:1603.05162*.
- Szabó, M. (2018). Kalmár’s argument against the plausibility of Church’s thesis. *History and Philosophy of Logic*, 39(2), 140–157. Retrieved from <https://www.tandfonline.com/doi/abs/10.1080/01445340.2017.1396520>
- Teuscher, C. & Sipper, M. (2002). Hypercomputation: Hype or computation? *Communications of the ACM*, 45(8), 23–24.
- Turing, A. M. (1937). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1), 230–265.
- Turing, A. M. (1939). Systems of logic based on ordinals. *Proceedings of the London mathematical society*, 2(1), 161–228.

- Wegner, P. & Goldin, D. (2003). Computation beyond Turing machines. *Communications of the ACM*, 46(4), 100–102. Retrieved from <https://dl.acm.org/citation.cfm?id=641235>
- Welch, P. D. (2004). On the possibility, or otherwise, of hypercomputation. *The British Journal for the philosophy of Science*, 55(4), 739–746. Retrieved from <https://www.jstor.org/stable/3541626>
- Windelband, W. (1998). History and Natural Science: Rectorial Address, University of Strassburg, 1894. *Theory and Psychology*, 8(1), 5–22.
- Wittgenstein, L. (1953). *Philosophical investigations*. Oxford.
- Wüthrich, C. (2015). A quantum-information-theoretic complement to a general-relativistic implementation of a beyond-Turing computer. *Synthese*, 192(7), 1989–2008. Retrieved from <https://link.springer.com/article/10.1007/s11229-014-0502-6>
- Ziegler, M. (2005). Computational power of infinite quantum parallelism. *International Journal of Theoretical Physics*, 44(11), 2059–2071. Retrieved from <https://link.springer.com/article/10.1007/s10773-005-8984-0>
- Ziegler, M. (2009). Physically-relativized Church–Turing Hypotheses: Physical foundations of computing and complexity theory of computational physics. *Applied Mathematics and Computation*, 215(4), 1431–1447. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0096300309004226>
- Zuse, K. (1969a). *Rechnender Raum, Schriften zur Datenverarbeitung, vol. 1. Vieweg: Braunschweig.*
- Zuse, K. (1969b). *Rechnender Raum. Vieweg: Braunschweig. English translation: Calculating Space. MIT Technical Translation AZT-70-164-GEMIT, Massachusetts Institute of Technology (Proj. MAC), Cambridge, Mass, 2139, 1970.*