CDMTCS
Research
Report
Series

Pre-proceedings of the Workshop "Physics and Computation" 2008

C. S. Calude[1], J. F. Costa[2] (eds.)
[1]University of Auckland, NZ,
[2]University of Wales Swansea, UK

CDMTCS-327
July 2008

Centre for Discrete Mathematics and
Theoretical Computer Science

Cristian S. Calude and José Félix Costa (eds.)

PHYSICS AND COMPUTATION

(Renaissance) International Worshop

Vienna, Austria, August 25–28, 2008

Pre-proceedings

Preface

In the 1940s, two different views of the brain and the computer were equally important. One was the analog technology and theory that had emerged before the war. The other was the digital technology and theory that was to become the main paradigm of computation.[1] The outcome of the contest between these two competing views derived from technological and epistemological arguments. While digital technology was improving dramatically, the technology of analog machines had already reached a significant level of development. In particular, digital technology offered a more effective way to control the precision of calculations. But the epistemological discussion was, at the time, equally relevant. For the supporters of the analog computer, the digital model — which can only process information transformed and coded in binary — wouldn't be suitable to represent certain kinds of continuous variation that help determine brain functions. With analog machines, on the contrary, there would be few or no layers between natural objects and the work and structure of computation (cf. [4, 1]). The 1942–52 Macy Conferences in cybernetics helped to validate digital theory and logic as legitimate ways to think about the brain and the machine [4]. In particular, those conferences helped made McCulloch-Pitts' digital model of the brain [3] a very influential paradigm. The descriptive strength of McCulloch-Pitts model led von Neumann, among others, to seek identities between the brain and specific kinds of electrical circuitry [1].

This was perhaps the first big event that brought together physicists and the (fathers of) computation theory.

Physics and computation theory have interacted from the early days of computing. After a joint start we witnessed the famous late 1950s divorce (fuelled by the hope of doing machine-independent computation) only to realise in the 1980s that, ultimately, physics laws permit computation. As a consequence the first important group of questions have gravitated around the constraints (known) physical laws put on (realisable) computers and computations. As a typical example we cite Lloyd's paper [2].

Quantum computing, relativistic computing, and, more recently, wireless sensor networks (sensornets) are three examples of different types of computations which differ from classical computation because of physical constraints. While the first two paradigms don't deserve any special introduction the third one does. Sensornets is a computing platform that blends computation, sensing and communication with a physics environment (see [5]). While classical complexity theory deals with time and space resources and their generalisations, sensornets have pointed to a new computational resource: *energy*. These ideas lead to the urgent need of a *theory of computational energy complexity*, a subject some people are already thinking about.

[1] For example, students at MIT could at that time learn about differential analysers *and* electronic circuits for binary arithmetic [4].

Secondly, but not less important, is the flow of ideas coming from computability theory to physics. Looking at physics with a guided computation/information eye we can ask: What, if anything, do the theories of computation and information can say about physics, what physical laws can be deduced using Wheeler's dictum "it from bit"? Computational physics has emerged, along with experiment and theory, as the third, new and complementary, approach to discovery in physics.

There is a long tradition of workshops on "Physics and Computation" inaugurated by the famous 1982 meeting whose proceedings have been published in a special issue of the *Int. J. Theor. Phys.* Volume 21, Numbers 3–4, April (1982) which starts with Toffoli's programatic article "Physics and computation" (pp. 165–175).

In a first organisational act of re-inaugurating the series of workshops on "Physics and Computation", we decided to invite twenty eight reputable researchers from the borderlines between computation theory and physics, but also from those sciences that interact strongly with physics, such like chemistry (reaction-diffusion model of computation), biology (physical-chemical driven organisms), and economic theory (macro-economic models), covering as much as possible all active fields on the subject. Nineteen researchers answer yes to our call and the pre-proceedings of this workshop is the product of their work. Our second act will be to organise next year a second workshop with invited lectures and contributed talks, now moving towards a more standard workshop or even a small conference.

The main fields covered by this event are (a) analog computation, (b) experimental computation, (c) Church-Turing thesis, (d) general dynamical systems computation, (e) general relativistic computation, (f) optical computation, (g) physarum computation, (h) quantum computation, (i) reaction-diffusion computation, (j) undecidability results for physics and economic theory.

∇

The organisers of this event are grateful for the highly appreciated work done by the reviewers of the papers submitted to the workshop. These experts were: Samson Abramsky, Andrew Adamatzki, Edwin Beggs, Udi Boker, Olivier Bournez, Časlav Brukner, Manuel Lameiras Campagnolo, S. Barry Cooper, Ben De Lacy Costello, Jean-Charles Delvenne, Francisco Dória, Fernando Ferreira, Jérôme Durand-Lose, Luís M. Gomes, Jerzy Górecki, Daniel Graça, Emmanuel Hainry, Andrew Hodges, Mark Hogarth, Bruno Loff, Aono Masashi, Cris Moore, Jerzy Mycka, István Németi, James M. Nyce, Kerry Ojakian, Oron Shagrir, Andrea Sorbi, Mike Stannett, Karl Svozil, Cristof Teuscher, John V. Tucker, Kumaraswamy Velupillai, Philip Welch, Damien Woods, Martin Ziegler, Jeffery Zucker.

We extend our thanks to all members of the local Conference Committee of the Conference UC'2008, particularly to Aneta Binder, Rudolf Freund (Chair of UC'2008), Franziska Gusel, and Marion Oswald of the Vienna University of Technology for their invaluable organisational work.

The venue for the conference was the Parkhotel Schönbrunn in immediate vicinity of Schönbrunn Palace, which, together with its ancillary buildings and extensive park, is by virtue of its long and colourful history one of the most important cultural monuments in Austria. Vienna, located in the heart of central Europe, is an old city whose historical role as the capital of a great empire and the residence of the Habsburgs is reflected in its architectural monuments, its famous art collections and its rich cultural life, in which music has always played an important part.

The workshop was partially supported by the Institute of Computer Languages of the Vienna University of Technology, the Centre for Discrete Mathematics and Theoretical Computer Science of the University of Auckland, the Kurt Gödel Society, and the OCG (Austrian Computer Society); we extend to all our gratitude.

C. S. Calude, J. F. Costa
Auckland, NZ, Swansea, UK

References

[1] S. J. Heims. *John von Neumann and Norbert Wiener: From Mathematics to the Technologies of Life and Death*, MIT Press, 1980.

[2] S. Lloyd. Ultimate physical limits of computation, *Nature*, 406:1047–1054, 2000.

[3] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[4] J. M. Nyce. Nature's machine: mimesis, the analog computer and the rhetoric of technology. In R. Paton (ed), *Computing with Biological Metaphors*, 414-423, Chapman & Hall, 1994.

[5] F. Zhao. The technical perspective, *Comm. ACM*, 51, $7:98$, July (2008).

Contents

From reaction-diffusion to Physarum computing

Andrew Adamatzky

University of the West of England Bristol BS16 1QY United Kingdom
andrew.adamatzky@uwe.ac.uk

Abstract

We experimentally demonstrate that computation of spanning trees and implementation of general purpose storage-modification machines can be executed by a vegetative state of the slime mold *Physarum polycephalum*. We advance theory and practice of reaction-diffusion computing by studying a biological model of reaction-diffusion encapsulated in a membrane.

Key words: reaction-diffusion computing, biological computing, spanning trees, computational universality
PACS: 87.17.Ee, 87.18.Hf, 87.18.Pj, 89.20.Ff, 89.75.-k, 05.45.-a,

1 Introduction: deficiencies of reaction-diffusion computers

In reaction-diffusion computers [2,6], data is presented by an initial concentration profile or a configuration of disturbance (e.g., sites of stimulation of excitable media). The information is transfered by spreading wave patterns, computation is implemented in collisions of wave-fronts, and final concentration profile represents results of the computation. Reaction-diffusion computers have been proved theoretically and experimentally capable for quite sophisticated computational tasks, including image processing and computational geometry, logics and arithmetics, and robot control (see [6] for detailed references, and overview of theoretical and experimental results). There is a particular feature of reaction-diffusion chemical computers: their classical, and so far commonly accepted form, the media are 'fully conductive' for chemical or excitation waves. Every point of a two- or three-dimensional medium can be involved in the propagation of chemical waves and reactions between diffusing chemical species. Once a reaction is initiated in a point, it spreads all over the computing space by target and spiral waves. Such phenomena of wave-propagation, analogues to one-to-all broadcasting in massive-parallel systems, are employed to solve problems ranging from the Voronoi diagram construction to robot navigation [2,6]. We could not, however, quantize information (e.g., assign logical values to certain waves) or implement one-to-one transmission in fully reactive media.

The field of reaction-diffusion computing was started by Kuhnert, Agladze and Krinsky [40,41], who, over twenty years ago, published their pioneering results on memory implementation and basic image processing in light-sensitive excitable chemical systems. Their ideas were further developed by Rambidi and colleagues [57,58]; and in Showalter and Yoshikawa's laboratories, who designed a range of chemical logical gates [71,42]. The computation of the shortest path, one of the classical optimization problems, has also been implemented in these laboratories using Belousov-Zhabotinsky media [66,12,6]. Untill quite recently, the only way to direct and quantize information in a chemical medium was to geometrically constrain the medium. Thus, only reactive or excitable channels are made, along which waves can travel. The waves collide with other waves at the junctions between the channels and implement certain logical gates in result of the collision (see an overview in Chapter 1 of [6]). Designs based on the geometrical constraining of the reaction-diffusion media are somewhat restricted by the conventionality of their architectures. This is because they simply re-enact standard computing architectures in non-standard 'conductive' materials.

Using sub-excitable media may be a successful way to quantize information. In sub-excitable media a local disturbance leads to the generation of mobile localization, where wave-fragments travel for a reasonably long distance without changing its shape [61]. The presence of a wave-fragment in a given domain of space signifies logical truth, the absence of the fragment logical falsity [5]. Despite being really promising candidates for collision-based computers [5], sub-excitable media are highly sensitive to experimental conditions and compact traveling wave-fragments are unstable and difficult to control.

In terms of well-established computing architectures, the following characteristics can be attributed to reaction-diffusion computers:

- massive-parallelism: there are thousands of elementary processing units, or microvolumes, in a standard Petri dish [6];
- local connections: microvolumes of a non-stirred chemical medium change their states (due to diffusion and reaction) depending on states of (concentration of reactants in) their closest neighbours;
- parallel input and output: in chemical reactions with indicators concentration profiles of the reagents, one can allow for parallel optical output. There is also a range of light-sensitive chemical reactions where data can be inputted via local disturbances of illumination [6].
- fault-tolerance: being in liquid phase, chemical reaction-diffusion computers

do restore their architecture even after substantial part of the medium is removed, however, the topology and the dynamics of diffusive and, particularly, phase waves (e.g., excitation waves in Belousov-Zhabotinsky system) may be affected.

Reaction-diffusion computers — when implemented in chemical medium — are much slower than silicon-based massively-parallel processors. When nano-scale materials are employed, e.g., networks of single-electron circuits [6], reaction-diffusion computers can however outperform even the most advanced silicon analogues.

There still remains a range of problems where chemical reaction-diffusion processors could not cope with without the external support from conventional silicon-based computing devices. The shortest path computation is one of such problems.

One can use excitable media to outline a set of all collision-free paths in a space with obstacles [4], but to select and visualize the shortest path amongst all possible paths, one needs to use an external cellular-automaton processor, or conceptually supply the excitable chemical media with some kind of field of the local pointers [4]. Experimental setups [66,12], which claim to directly compute a shortest path in chemical media, are indeed employing external computing resources to store time-lapsed snapshots of propagating wave-fronts and to anylise the dynamics of the wave-front propagation. Such usage of external resources dramatically reduces the fundamental values of the computing with propagating patterns.

Graph-theoretical computations pose even more difficulties for spatially-extended non-linear computers. For example, one can compute the Voronoi diagram of a planar set, but can not invert this diagram [6]. Let us consider a spanning tree, most graph famous of classical proximity graphs. Given a set of planar points one wants to connect the points with edges, such that the resultant graph has no cycles and there is a path between any two points of the set. So far, no algorithms of spanning tree construction were experimentally implemented in spatially extended non-linear systems. This is caused mainly by uniformity of spreading wave-fronts, their inability to sharply select directions toward locations of data points, and also because excitable systems usually do not form stationary structures.

Essentially, to compute a spanning tree over a given planar set, a system must first explore the date space, then cover the data points and physically representing edges of the tree by the system's structure. This is not possible in excitable chemical systems because they are essentially memoryless, and no stationary structure can be formed. Precipitating reaction-diffusion systems are also uncapable of constructing the trees because not only they operate with uniformly expanding diffusive fronts, but the systems are incapable of altering concentration profile of the precipitate once precipitation occurred.

To overcome these difficulties, we should allow reaction-diffusion computers to be geometrically self-constrained while still capable to operate in geometrically unconstrained (architectureless or ‘free’) space. Encapsulating reaction-diffusion processes in membranes would a possible solution. The idea is explored in the present paper. Based on our previous results [7–9], we speculate that vegetative state, or plasmodium, of *Physarum polycephalum* is a reaction-diffusion system constrained by a membrane that capable for solving graph-theoretical problems (not solvable by ‘classical’ reaction-diffusion computers) and which is also computationally universal.

A brief introduction to computing with *Physarum polycephalum* is presented in Sect. 2. Section 3 introduces our experimental findings on constructing spanning trees of finite planar sets by plasmodium of *Physarum polycephalum*. In Sect. 4 we demonstrate that plasmodium of *Physarum polycephalum* is an ideal biological substrate for the implementation of Kolmogorov-Uspensky machines [9]. Directions of further studies are outlined in Sect. 5.

2 Physarum computing

There is a real-world system which strongly resembles encapsulated reaction-diffusion system. *Physarum polycephalum* is a single cell with many nuclei which behave like amoeba. In its vegetative phase, called plasmodium, slime mold actively searches for nutrients. When another source of food is located, plasmodium forms a vein of protoplasm between previous and current food-sources.

Why the plasmodium of *Physarum* is an analog of an excitable reaction-diffusion system enclosed in a membrane? Growing and feeding plasmodium exhibits characteristic rhytmic contractions with articulated sources. The contraction waves are associated with waves of potential change, and the waves observed in plasmodium [43,44,77] are similar to the waves found in excitable chemical systems, like Belousov-Zhabotinsky medium. The following wave phenomena were discovered experimentally [77]: undisturbed propagation of contraction wave inside the cell body, collision and annihilation of contraction waves, splitting of the waves by inhomogeneity, and the formation of spiral waves of contraction (see Fig. 6c–f in [77]). These are closely matching dynamics of pattern propagation in excitable reaction-diffusion chemical systems.

Yamada and colleagues [77] indicate a possibility for interaction between the contraction force generating system with oscillating chemical reactions of calcium, ATP and associated pH [56,51,53]. Chemical oscillations can be seen as primary oscillations — contraction waves are guided by chemical oscillation waves — because chemical oscillations can be recorded in absence of contractions [46,52,54].

Nakagaki, Aono, Tsuda [15,16,73,49,50,74] and others have been exploring a power of Physarum computing from 2000 [48]. They proved experimentally that the plasmodium is a unique fruitful object to design various schemes of non-classical computation [15,16,73], including Voronoi diagram [63] and shortest path [49,50,63], and even design of robot controllers [74]. In present we paper we focus on one specialized instance of Physarum computing — approximation of spanning trees, and also implementation of a general purpose storage-modification machine.

The scoping experiments were designed as follows. We either covered container's bottom with a piece of wet filter paper and placed a piece of living plasmodium [1] on it, or we just planted plasmodium on a bottom of a bare container and fixed wet paper on the container's cover to keep the humidity high. Oat flakes were distributed in the container to supply nutrients and represent set of nodes to be spanned by a tree (Sect. 3) or to represent data-nodes of Physarum machine (Sect. 4). The containers were stored in the dark except during periods of observation. To color oat flakes, where required, we used SuperCook Food Colorings: [2] blue (colors E133, E122), yellow (E102, E110, E124), red (E110, E122), and green (E102, E142). The flakes were saturated with the colorings, then dried.

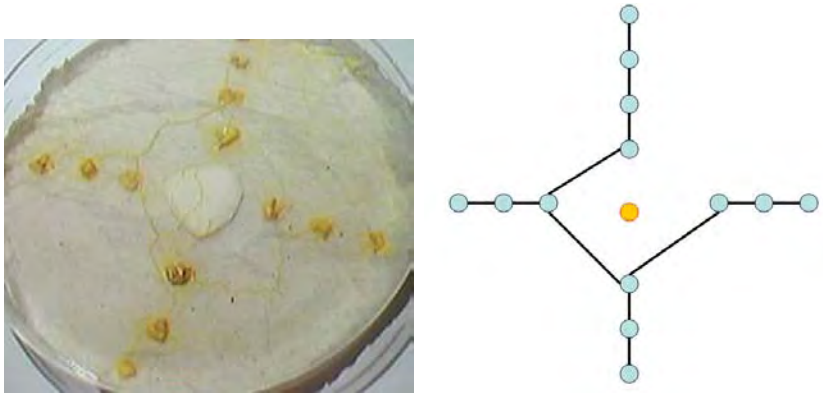
3 Approximation of spanning trees

The spanning tree of a finite planar set is a connected, undirected, acyclic planar graph, which vertices are points of the planar set; every point of the given planar set is connected to the tree (but no cycles or loops are formed). The tree is a minimal spanning tree where the sum of edges' lengths is minimal. Original algorithms for computing minimum spanning trees are described in [39,55,26]. Hundreds if not thousands of papers were published in last 50 years, mostly improving the original algorithms, or adapting them to multi-processor computing systems [28,13,32].

Non-classical and nature-inspired computing models brought their own solutions to the spanning tree problem. Spanning tree can be approximated by random walks, electrical fields, and even social insects [72,23,2,3]. However

[1] Thanks to Dr. Soichiro Tsuda for providing me with *P. polycephalum* culture.

[2] `www.supercook.co.uk`



(a) (b)

Fig. 1. Approximating spanning tree by plasmodium: (a) photograph of living plasmodium in a container, where oat flakes represent the nodes of the tree, (b) scheme of the tree computed by plasmodium.

neither of these non-classical algorithms offer an experimental realization.

In 1991 we proposed an algorithm of computing the spanning tree of a finite planar set based on the formation of a neurite tree in a development of a single neuron [1]. Our idea was to place a neuroblast somewhere on the plane amongst drops of chemical attractants, positions of which represent points of a given planar set. Then, the neurite tree starts to grow and spans the given planar set of chemo-attractants with acyclic graph of axonal and dendritic branches. Due to lateral circumstances experimental implementation of the algorithm was not possible at the time of its theoretical investigation [1]. Recent experimental developments in foraging behaviour of *P. Polycephalum* [49,50,74,15,16,73] convinced us that our original algorithm for morphological growing of spanning trees can be implemented by living plasmodium.

When computing the spanning tree, the plasmodium acts as follows: once placed in the container, where oat flakes represent given planar set to be spanned by a tree, and recovered, the plasmodium starts to explore the surrounding space. Numerous pseudopodia emerge, frequently branch and proceed. The plasmodium grows from its initial position by protoplasmic pseudopodia detecting, by chemotaxis, relative locations of closest sources of nutrients. When another source of nutrients, element of the given planar set, is reached, the relevant part of the plasmodium reshapes and shrinks to a protoplasmic strand, or a tube. This tube connects the initial and the newly acquired sites. This protoplasmic strand represents an edge of the computed spanning tree. Planar points distributed in a Petri dish are usually spanned by a protoplasmic vein tree in 1-3 days, depending on the diameter of the planar set, substrate and other conditions. An example of a spanning tree approximated by plasmodium is shown in Fig. 1.

The tree computed by plasmodium in our experiments [8] satisfactory match

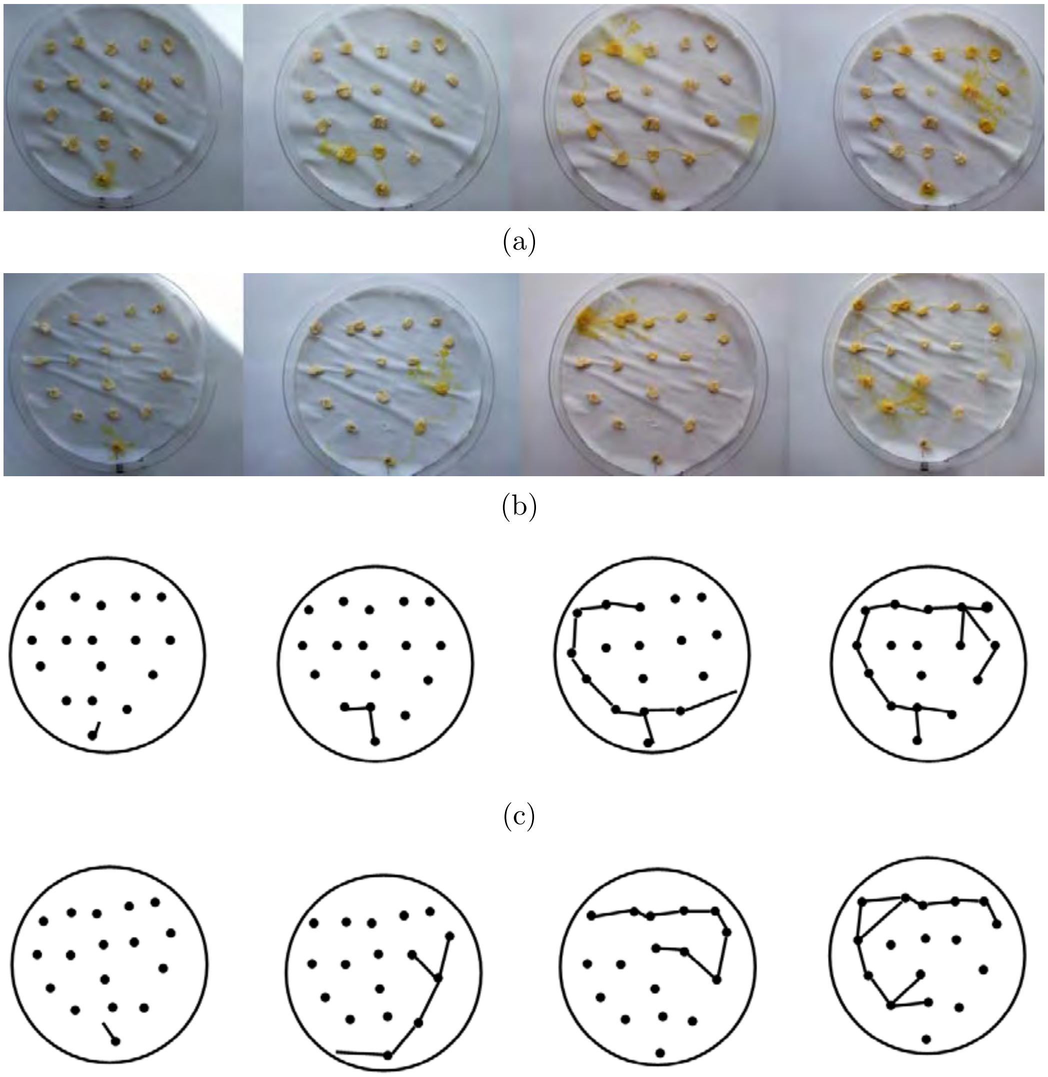


Fig. 2. Two scenarios of computing the spanning tree from the same planar data–points: (a) and (b) show photographs of the living plasmodium spanning oat flakes, which represent data nodes; (c) and (d) are schemes of the trees approximated. At the beginning of both experiments, plasmodium was placed at the Southmost oat flake.

trees computed by classical techniques, e.g., by Jaromczyk-Supowit method [33,67], see [8]. Even when represented in simulation, the algorithm works pretty well on large data sets [8].

We would like to refer those eager for details to our previous papers [8,7], where the advantages of computing the spanning tree by pladmodium are discussed. In the present paper we will mention two speculative points of the approximation.

Plasmodium almost never computes the same (including exact location of

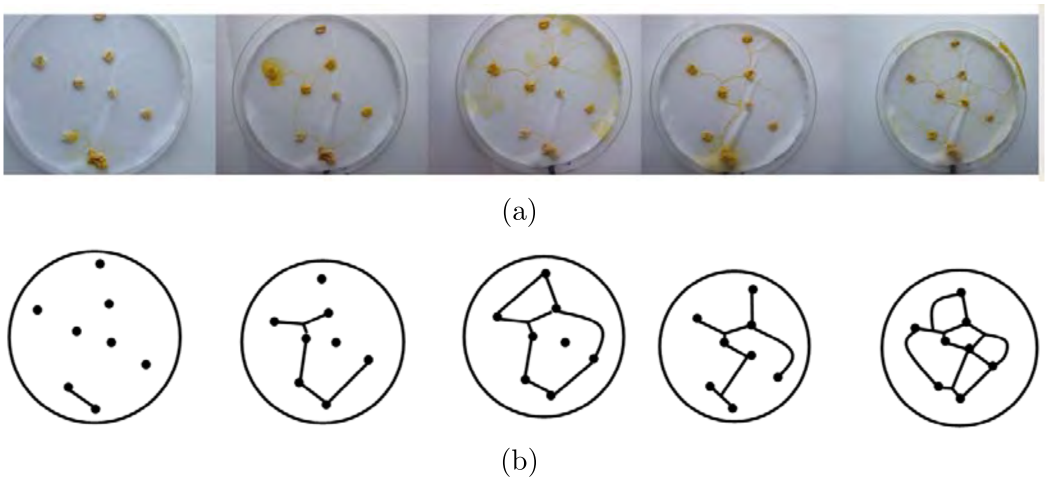


Fig. 3. Particular results of spanning planar data points by plasmodium (from the left to the right): first incomplete spanning tree is formed, then planar graph with cycles, then complete spanning tree; plasmodium continues its development after the tree is computed by transforming the tree again to a cyclic planar graph; (a) photographs of living plasmodium; (b) schemes of the graphs constructed.

protoplasmic edges) trees from the same data planar points. Not only the locations and configurations of the edges can be different, but also the topologies of the trees. An example is provided in Fig. 2. In one experiment plasmodium spans eastern and western data points while spreading North (Fig. 2ac), while in another experiment plasmodium relocates to the northern part of the data set and then spreads back South (Fig. 2bd).

Experimental results shown in Fig. 3 demonstrate that (1) tree can be constructed via other kinds of proximity graphs, or k-skeletons, and (2) plasmodium never stops 'computing', at some stage a tree is built, but then it is transformed to a planar graph with cycles. This experimental finding amazingly similar to how are spanning trees constructed on conventional computers – first a relative neighbourhood graph is computed, then some edges are deleted, and thus the graph is transformed to a minimum spanning tree [33,67].

4 Universal Physarum machines

In the late 1940s and early 1950s, while developing his ideas on recursive functions and recursively enumerable sets (which are fundamentals of algorithm theory) [76], Kolmogorov [37,38] established a formalism for algorithmic process realizable in physical time and space. He proposed that each state of an algorithm process is comprised of a finite number of elements and connections amongst them. Elements and connections belong to some types, and total number of types are bounded. Each connection has a fixed number of ele-

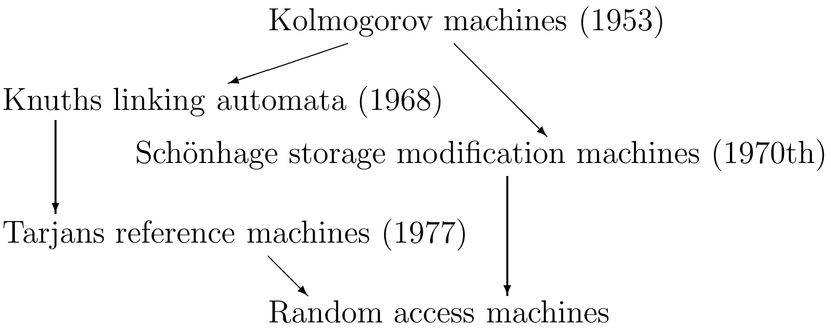

Kolmogorov machines (1953)
Knuths linking automata (1968)
Schönhage storage modification machines (1970th)
Tarjans reference machines (1977)
Random access machines

Fig. 4. Development of storage modification machines.

ments, and every element has a restricted number of connections. A restricted number of connections means locality in a sense that graphs connectivity is several orders less then the size of the graph. The state has a local active zone (i.e., specified elements) and connections amongst the elements can be updated dynamically. In computer science, Kolmogorov machine is treated as a computational device whose storage can change its topology. Later Kolmogorov's formalism was enriched by Uspensky, thus the name of the final abstract computational device.

A Kolmogorov-Uspensky machine (KUM) [37,38] is defined on a colored/labeled undirected graph with bounded degrees of nodes and bounded number of colors/labels. As Uspenski poetically said an algorithmic process "…can be regarded as a finite first-order structure of a finite signature, the signature being fixed for every particular algorithm"[76].

KUM operates, and modifies its storage as following: select an active node in the storage graph. Specify local active zone, the node's neighborhood. Modify the active zone, i.e. add a new node with the pair of edges, then connect the new node with the active node; delete a node with the pair of incident edges; add or delete edges between the nodes.

A program for KUM specifies how to replace the neighborhood of an active node with new neighborhood, depending on labels of edges connected to the active node and labels of the nodes in the proximity of the active node [19]. All previous and modern models of real-world computation are heirs of KUM: Knuth's linking automata [36], Tarjan's Reference Machines [68], Schönhage's storage modification machines [59,60] (Fig. 4. When the restrictions on bounded in- and out-degrees of the machine's storage graph are lifted, the machine becomes Random Access Machine.

Functions computable on Turing machines (TM) are also computed on KUM, and any sequential device are simulated by KUM [31]. KUM can simulate TM in real time, but not *vice versa* [30]. KUM's topology is much more flexible than that of TM, and KUM is also stronger than any 'tree-machine' [65].

In 1988 Gurevich [31], suggested that an edge of KUM is not only an informational but also a physical entity and reflects the physical proximity of the nodes (thus e.g. even in three-dimensional space number of neighbors of each node is polynomially bounded). A TM formalizes computation as performed by *humans*[75], whereas KUM formalizes computation as performed by *physical process* [19].

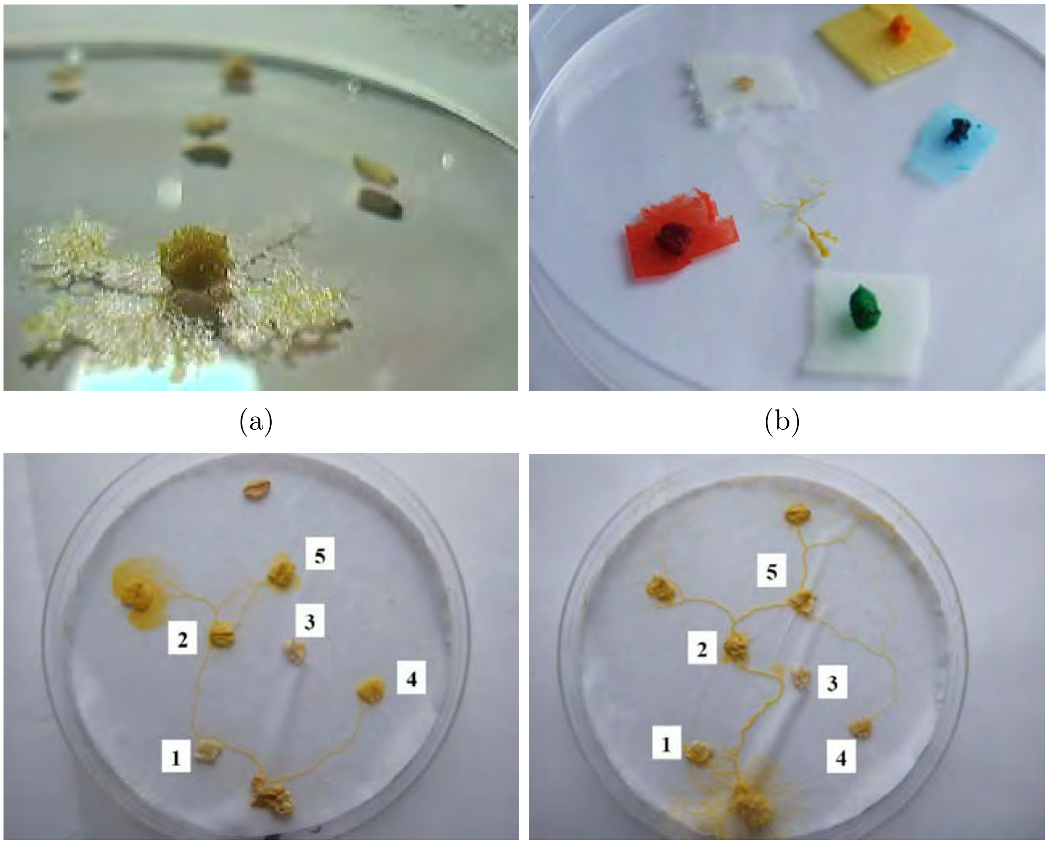
What would be the best natural implementation of KUM? A potential candidate should be capable for growing, unfolding, graph-like storage structure, dynamically manipulating nodes and edges, and should have a wide range of functioning parameters. Vegetative stage, i.e., plasmodium, of a true slime mold *Physarum polycephalum* satisfies all these requirements.

Physarum machine has two types of nodes: stationary nodes, presented by sources of nutrients (oat flakes), and dynamic nodes, sites where two or more protoplasmic veins originate. At the beginning of the computation, the stationary nodes are distributed in the computational space, and the plasmodium is placed at one point of the space. Starting in the initial conditions, the plasmodium exhibits foraging behavior, and occupies stationary nodes.

An edge of Physarum machine is a strand, or vein, of a protoplasm connecting stationary and/or dynamic nodes. KUM machine is an undirected graph, i.e., if nodes x and y are connected, then they are connected by two edges (xy) and (yx). In Physarum machine this is implemented by a single edge but with periodically reversing flow of protoplasm [34,47].

Program and data are represented by a spatial configuration of stationary nodes. Result of the computation over stationary data-node is presented by a configuration of dynamical nodes and edges. The initial state of a Physarum machine, includes part of input string (the part which represents position of plasmodium relatively to stationary nodes), an empty output string, a current instruction in the program, and a storage structure consists of one isolated node. That is, the whole graph structure developed by plasmodium is the result of its computation, "if S is a terminal state, then the connected component of the initial vertex is considered to be the 'solution"' [38]. Physarum machine halts when all data-nodes are utilized.

In KUM, a storage graph must have at least one active node. This is an inherent feature of Physarum machines. When the plasmodium resides on a substrate with poor or no nutrients, then just one or few nodes generate actively spreading protoplasmic waves. In these cases, the protoplasm spreads as mobile localizations similar to wave-fragments in sub-excitable Belousov-Zhabotinsky media [61]. An example of a single active node, which has just started to develop its active zone, is shown in Fig. 5. At every step of computation there is an active node and an active zone, usually nodes neighboring to



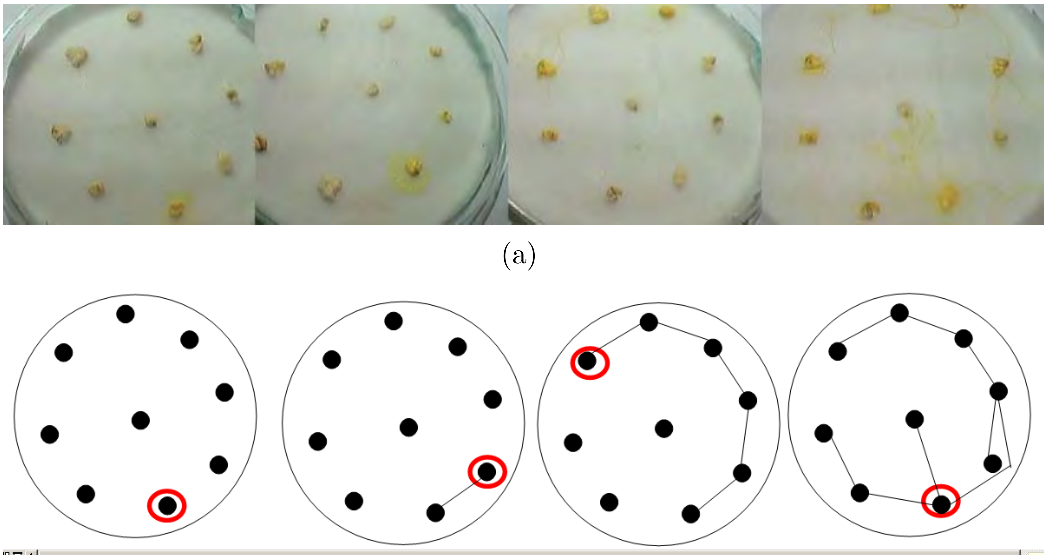
(a) (b)

(c) (d)

Fig. 5. Basic operations of Physarum machine: (a) a single active node generates an active zone at the beginning of computation, (b) addressing of a green-coloured data-node, (c) and (d) implementation of ADD NODE (nodes 3), ADD EDGE (edge (5, 4)), REMOVE EDGE (edge (*route*, 4)) operations.

active node. The active zone has limited complexity, in a sense that all elements of the zone are connected by some chain of edges to the initial node. In general, the size of an active zone may vary depending on the computational task. In Physarum machine an active node is a trigger of contraction/excitation waves, which spread all over the plasmodium tree and cause pseudopodia to propagate, change their shape, and even protoplasmic veins to annihilate. An active zone is comprised of stationary or dynamic nodes connected to an active node with veins of protoplasm.

KUM, in its original form, have a single control device [31,19]. Plasmodium acts as a unit in a long-term, i.e., it can change position, or retract some processes in one place to form new ones in another place. However, periodic contractions of the protoplasm are usually initiated from a single source of contraction waves [47,69]. The source of the waves can be interpreted as a single control unit. In some cases we experimentally observed (Fig. 6) presence of a single active zone in the growing plasmodium. However, during foraging behavior, several branches or processes of plasmodium can act independently



(a)

(b)

Fig. 6. Serial approximation of spanning tree by plasmodium: (a) snapshots of the living plasmodium, from the left to the right, made with 6 hours intervals; (b) scheme of the graph. The active zone at each step of computation is encircled.

and almost autonomously.

In contrast to Schönhage machine, KUM has bounded in- and out-degree of the storage graph. Graphs developed by Physarum are predominantly planar graphs. Moreover, if we put a piece of vein of protoplasm on top of another vein of protoplasm, the veins fuse [62]. Usually, not more than three protoplasmic strands join each other in one given point of space. Therefore we can assume that the average degree of the storage graph in Physarum machines is slightly higher then the degree of a spanning tree (average degree of 1.9 as reported in [22]) but smaller than the average degree of a random planar graph (degree 4 [14]).

Every node of KUM must be uniquely addressable and nodes and edges must be labeled [38]. There is no direct implementation of such addressing in Physarum machine. With stationary nodes this can be implemented, for example, by coloring the oat flakes. An example of such experimental implementation of a unique node addressing is shown in Fig. 5b.

A possible set of instructions for Physarum machine could be as follows: common instructions would include INPUT, OUTPUT, GO, HALT, and internal instructions: NEW, SET, IF [25]. At the present state of the experimental implementation, we assume that INPUT is done via distribution of sources of nutrients, while OUTPUT is recorded optically. The instruction SET causes pointers redirection, and can be realized by a placing a fresh source of nutrients in the experimental container, preferably on top of one of the old sources of nutrients. When a new node is created, all pointers can be redirected from the old node to the new node. Let us look at the experimental implementation of the core instructions.

To add a stationary node b to node a's neighborhood, the plasmodium must propagate from a to b, and form a protoplasmic vein representing the edge (ab). To form a dynamic node, propagating the pseudopodia must branch into two or more pseudopodia, and the site of branching will represent the newly formed node. We have also obtained experimental evidence that dynamic nodes can be formed when a tip of growing pseudopodia collides with existing protoplasmic strande. In some cases merging of protoplasmic veins occur.

To remove the stationary node from Physarum machine, the plasmodium leaves the node. Annihilating protoplasmic strands, which form a dynamic node at their intersection, remove the dynamic node from the storage structure of Physarum machine.

To add an edge to a neighborhood, an active node generates propagating processes, which establish a protoplasm vein with one or more neighboring nodes.

When a protoplasmic vein annihilates, e.g., depending on the global state or when source of nutrients exhausted, the edge represented by the vein is removed from Physarum machine (Fig. 5cd). The following sequence of operations is demonstrated in Fig. 5cd: node 3 is added to the structure by removing edge (12) and forming two new edges (13) and (23).

Let us consider an example of a task solvable by Physarum machine. Given nodes labeled Red, Green, Blue, Yellow, connect only Green, Blue and Yellow nodes in a single chain. Physarum machine solves the task by first exploring the whole data space, then connecting required nodes (Fig. 7).

A possible compromise between the original theoretical framework and the partly parallel execution in experiments could be reached by proposing two levels of 'biological' commands executed by Physarum machine's elements. There would have to be high-level commands, e.g. search for nutrients, escape light, form sclerotium, fructify, and low-level commands, e.g., form process, propagate in direction of, occupy source of nutrients, retract process, branch. Global commands are executed by plasmodium as a whole at once, i.e., in a given time step the plasmodium executes only one high-level command. Local commands are executed by local parts of the plasmodium. Two spatially distant sites can execute different low-level commands at the same time.

One of the referees questioned what would be the implementation of the halt command. So far we do not have any. The plasmodium continues its developed

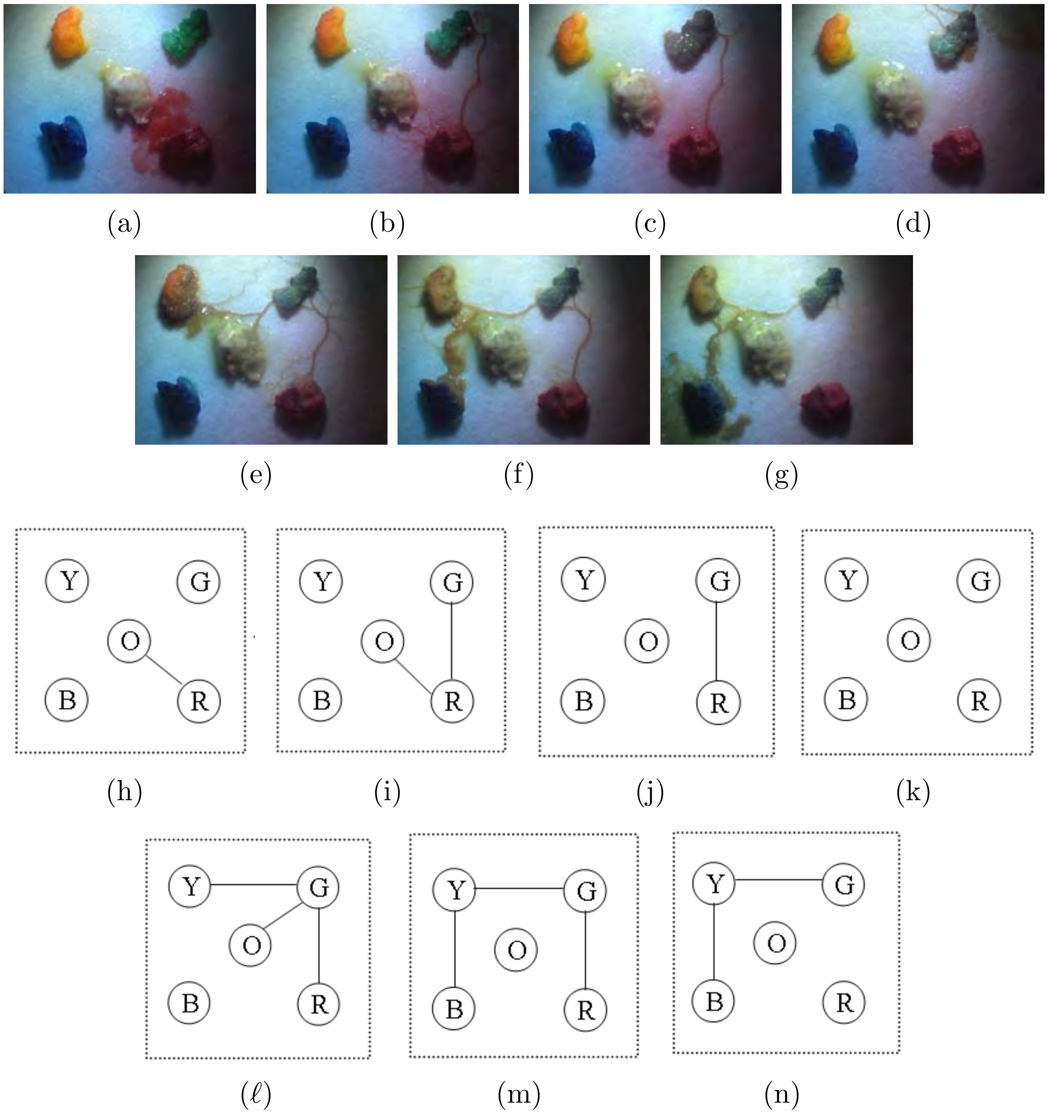


Fig. 7. Implementation of a simple task of connecting coloured nodes by Physarum machine: (a)–(g) shows a sequence of photographs of plasmodium (magnification ×10) placed in a small container in the centre of a rectangle, which corners represented by oat flakes that are coloured in yellow, green, red, and blue; the scheme of the computation is shown in (h)–(n).

and colonizes spaces even when e.g. a spanning tree is completed. We can however 'froze' the computation by depriving plasmodium from water. In a low humidity conditions the plasmodiums stops its foraging behaviour and forms a sclerotium, a compact mass of hardened protoplasmic mass, see Fig. 8. Results of the computation are not destroyed and remain detectable as 'empty/dead' protoplasmic tubes. In the state of sclerotium the Physarum machine is ready for further deployment.

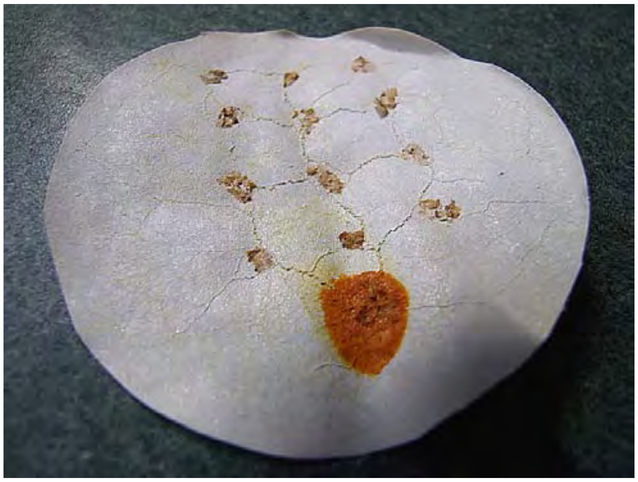


Fig. 8. Computation in Physarum is canceled by lowering humidity. Dark-brown coloured sclerotium is formed. You can also see empty and dead protoplasmic tubes, which formed previously active proximity graph spanning food sources.

5 Discussions

Up to date, there were three types of reaction-diffusion computers with respect to the geometry of reactor space and reaction wave propagation.

First, unconstrained reaction-diffusion computers: once locally perturbed target and spiral waves initiated, and they propagate everywhere. Computation can be performed at any point of the space, where travelling waves interact with each other. Such systems are massive parallel and successfully solve NP-complete problems of computational geometry, e.g., plane tessellation (Voronoi diagram), as well as robot guidance and navigation. The designs were also implemented in large-scale integrated circuits and possible nano-scale implementation in networks of single-electron oscillators were studied, see overview in [6]. The drawback of unconstrained reaction-diffusion computers is that they can not complete graph optimization tasks, such as spanning tree or shortest path, without help of external computing devices.

Second, geometrically constrained reaction-diffusion computers: the chemical medium resides only in channels and logical circuits are made of the channels, where computation happens at junctions between two or more channels, see e.g. [71,64,29,45]. The geometrically constrained systems can implement Boolean and multiple-valued logical operations, as well as count and process sequences of signals. There are two deficiencies: (1) geometrically constrained reaction-diffusion computers are essentially only implementations of conventional computing architectures with wires and logical switches in novel chemical materials; and, (2) the intrinsic parallelism of the medium is not utilized properly.

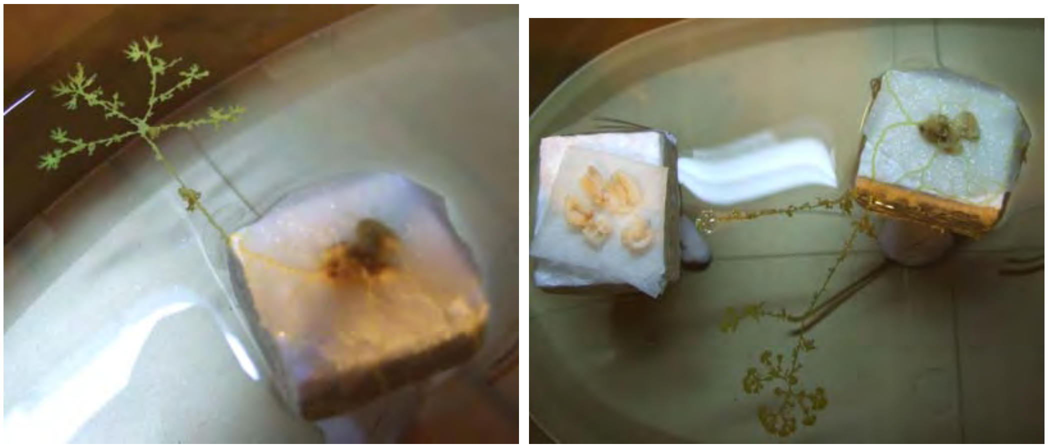
Third, reaction constrained reaction-diffusion computers: travelling waves can be initiated and then propagate anywhere in the reaction space. However due to low reactivity of the system, no classical (e.g., target) waves are formed. This is typical for sub-excitable Belousov-Zhabotinsky systems. Local perturbation leads to formation of compact wave-fragments which can travel for reasonable distance preserving its shape, see e.g. [61]. The system is an ideal implementation of collision-based computing schemes [5]. The deficiency of the system is that travelling compact wave-fragment are very sensitive to conditions of the medium, and therefore are cumbersome to control.

To overcome all these deficiencies of reaction-diffusion computers, we suggested to encapsulate the reaction-diffusion systems in a membrane because this seems to be a good combination of unconstrained physical space and membrane-constrained geometry of propagating patterns[3]. Also, a system encapsulated in an elastic or contractable membrane would be capable of reversible shape changing, a feature unavailable in reaction-diffusion chemical systems. We demonstrated that the vegetative state, the plasmodium, of slime mold *Physarum polycephalum*, is an ideal biological medium for suggested implementation. We have provided first experimental evidences that plasmodium can compute spanning trees of finite planar sets and implement Kolmogorov-Uspensky machines. Thus, Physarum computers can solve graph-theoretic problems and are capable for universal computation.

Physarum computers will be particularly efficient in solving large-scale graph and network (including telecommunication and road traffic networks) optimization tasks, and can also be used as embedded controllers for non-silicon (e.g., gel-based) reconfigurable robots and manipulators. They are reasonably robust (they live on almost any non-aggressive substrate including plastic, glass and metal foil, in a wide range of temperatures, they do not require special substrates or sophisticated equipment for maintenance) and are programmable (plasmodium exhibits negative phototaxis, and can follow gradients of humidity and some chemo-attractants) spatially extended and, distributed, computing devices.

In contrast to ‘classical’ chemical reaction-diffusion computers [6], Physarym machines can function on virtually any biologically non-agressive substrate, include metal and glass. Moreover, the substrate does not have to be static. For example, to implement Physarum machines with mobile data nodes, one can use a container with water, place the plasmodium of Physarum on one floating object, and the oat flakes (data) on several other floating objects.

[3] Recently we have demonstrated in chemical and biological laboratory experiments that plasmodoium of *Physarun polycephalum* behaves almost exactly the same, apart of leaving a ’trace’, as excitation patterns in *sub-excitable* Belousov-Zhabotinsky medium, see details in [10]. Thus plasmodium is also proved to be capable for collision-based universal computation [5].



(a)

(b)

Fig. 9. A floating Physarum machine: (a) an active zone of Physarum machine travelling on water surface, (b) a connection is formed between the active zone and one of the data points.

Plasmodium will then explore the physical space, travelling on the surface of the water, and eventually set up connections between data points (Fig. 9). First steps towards Physarum robots are reported at [11].

Future research will concentrate on expanding the domain of graph-theoretic tasks solvable by Physarum computers, developing programming language for Physarum machines, design and experimental implementation of plasmodium-based intelligent manipulators, and general purpose logical and arithmetical circuits.

Acknowledgment

Many thanks to Dr. Christof Teuscher (Los Alamos Labs, US) for editing the manuscript. I am grateful to Dr. Soichiro Tsuda (Southmapton Univ, UK) for providing me with the culture of *P*hysarum polycephalum and subsequent fruitful discussions.

References

[1] Adamatzky A., Neural algorithm for constructing minimal spanning tree. Neural Network World, 6 (1991) 335–339.

[2] Adamatzky A., Computing in non-linear media and automata collectives, IoP Publishing, 2001, 401 pp.

[3] Adamatzky A. and Holland O. Reaction-diffusion and ant-based load balancing of communication networks. Kybernetes 31 (2002) 667–681.

[4] Adamatzky A. and De Lacy Costello B.P.J., Collision-free path planning in the Belousov-Zhabotinsky medium assisted by a cellular automaton. Naturwissenschaften 89 (2002) 474–478.

[5] Adamatzky A. (Ed.) Collision-Based Computing, Springer, London, 2003.

[6] Adamatzky A., De Lacy Costello B., Asai T. Reaction-Diffusion Computers, Elsevier, Amsterdam, 2005.

[7] Adamatzky A. Physarum machines: encapsulating reaction-diffusion to compute spanning tree. Naturwisseschaften 94 (2007) 975–980.

[8] Adamatzky A. Growing spanning trees in plasmodium machines, Kybernetes: The International Journal of Systems & Cybernetics 37 (2008) 258–264.

[9] Adamatzky A. Physarum machine: implementation of a Kolmogorov-Uspensky machine on a biological substrate. Parallel Processing Letters 17 (2007) 455-467.

[10] Adamatzky A., De Lacy Costello B., Shirakawa T. Universal computation with limited resources: Belousov-Zhabotinsky and Physarum computers. Int. J. Bifurcation & Chaos (2008), in press.

[11] Adamatzky A., Towards Physarum robots: computing and manipulating on water surface. `arXiv:0804.2036v1` [cs.RO]

[12] Agladze K., Magome N., Aliev R., Yamaguchi T. and Yoshikawa K. Finding the optimal path with the aid of chemical wave. Physica D 106 (1997) 247–254.

[13] Ahuja M. and Zhu Y. A distributed algorithm for minimum weight spanning tree based on echo algorithms In: *Proc. Int. Conf. Distr. Computing Syst.*, 1989, 2–8.

[14] Alber J., Dorn F., Niedermeier R., Experiments on Optimally Solving NP-complete Problems on Planar Graphs, Manuscript (2001), `http://www.ii.uib.no/~frederic/ADN01.ps`

[15] Aono, M., and Gunji, Y.-P., Resolution of infinite-loop in hyperincursive and nonlocal cellular automata: Introduction to slime mold computing. Computing Anticiaptory Systems, AIP Conference Proceedings, 718 (2001) 177–187.

[16] Aono, M., and Gunji, Y.-P., Material implementation of hyper-incursive field on slime mold computer. Computing Anticipatory Systems, AIP Conference Proceedings, 718 (2004) 188–203.

[17] Bardzin's J. M. On universality problems in theory of growing automata, Doklady Akademii Nauk SSSR 157 (1964) 542–545.

[18] Barzdin' J.M. and Kalnins J. A universal automaton with variable structure, Automatic Control and Computing Sciences 8 (1974) 6–12.

[19] Blass A. and Gurevich Y. Algorithms: a quest for absolute definitions, Bull. Europ. Assoc. TCS 81 (2003) 195–225.

[20] van Emde Boas, P. Space measures for storage modification machines. Information Process. Lett. 30 (1989) 103–110.

[21] Calude C.S., Dinneen M.J., Paun G., Rozenberg G., Stepney S. Unconventional Computation: 5th International Conference, Springer, 2006.

[22] Cartigny J., Ingelrest F., Simplot-Ryl D., Stojmenovic I., Localized LMST and RNG based minimum-energy broadcast protocols in ad hoc networks. Ad Hoc Networks 3 (2005) 1-16.

[23] Chong F. Analog techniques for adaptive routing on interconnection networks *M.I.T. Transit Note No. 14*, 1993.

[24] Cloteaux B. and Rajan D. Some separation results between classes of pointer algorithms, In DCFS '06: Proceedings of the Eighth Workshop on Descriptional Complexity of Formal Systems. 2006, 232–240.

[25] Dexter S., Doyle P. and Gurevich Yu. Gurevich abstract state machines and Schönhage storage modification machines, J Universal Computer Science 3 (1997) 279–303.

[26] Dijkstra E.A. A note on two problems in connection with graphs *Numer. Math.* 1 (1959) 269–271.

[27] Gacs P. and Leving L. A. Casual nets or what is a deterministic computation, STAN-CS-80-768, 1980.

[28] Gallager R.G., Humblet P.A. and Spira P.M. A distributed algorithm for minimum–weight spanning tree *ACM Tranbs. Programming Languages and Systems* 5 (1983) 66–77.

[29] Górecki J., Yoshikawa K. and Igarashi Y., On chemical reactors that can count, J. Phys. Chem., A 107 (2003) 1664–1669.

[30] Grigoriev D. Kolmogorov algorithms are stronger than Turing machines. Notes of Scientific Seminars of LOMI 60 (1976) 29–37, in Russian. English translation in J. Soviet Math. 14 (1980) 1445–1450.

[31] Gurevich Y., On Kolmogorov machines and related issues, in Bull. EATCS 35 (1988) 71–82.

[32] Huang S.–T. A fully pipelined minimum spanning tree constructor. J. Parall. Distr. Computing 9 (1990) 55–62.

[33] Jaromczyk J.W. and Kowaluk M. A note on relative neighborhood graphs Proc. 3rd Ann. Symp. Computational Geometry, 1987, 233–241.

[34] Kamiya N. The protoplasmic flow in the myxomycete plasmodium as revealed by a volumetric analysis, Protoplasma 39 (1950) 3.

[35] Kirkpatrick D.G. and Radke J.D. A framework for computational morphology. In: Toussaint G. T., Ed., Computational Geometry (North-Holland, 1985) 217-248.

[36] Knuth D. E. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, Addison-Wesley, Reading, Mass. 1968.

[37] Kolmogorov A. N., On the concept of algorithm, Uspekhi Mat. Nauk 8 (1953) 175–176.

[38] Kolmogorov A. N., Uspensky V. A. On the definition of an algorithm. Uspekhi Mat. Nauk, 13 (1958) 3–28, in Russian. English translation: ASM Translations 21 (1963) 217–245.

[39] Kruskal J.B. On the shortest subtree of a graph and the traveling problem. Proc. Amer. Math. Sec. 7 (1956) 48–50.

[40] Kuhnert L. A new photochemical memory device in a light sensitive active medium. Nature 319 (1986) 393.

[41] Kuhnert L., Agladze K. L., Krinsky V. I. Image processing using light-sensitive chemical waves. Nature 337 (1989) 244–247.

[42] Kusumi T., Yamaguchi T., Aliev R., Amemiya T., Ohmori T., Hashimoto H. Yoshikawa K. Numerical study on time delay for chemical wave transmission via an inactive gap. Chem. Phys. Lett. 271 (1997) 355–360.

[43] Matsumoto K., Ueda T., and Kobatake Y. Propagation of phase wave in relation to tactic responses by the plasmodium of *Physarum polycephalum*. J. of Theor. Biology 122 (1986) 339–345.

[44] Matsumoto K., Ueda T. and Kobatake Y. Reversal of thermotaxis with oscillatory stimulation in the plasmodium of *Physarum polycephalum*, J. Theor. Biology 131 (1988) 175–182.

[45] Motoike I., Adamatzky A. Three-valued logic gates in reaction-diffusion excitable media. Chaos, Solitons & Fractals 24 (2005) 107–114.

[46] Nakagaki T., Yamada H., Ito M., Reaction-diffusion advection model for pattern formation of rhythmic contraction in a giant amoeboid cell of the Physarum plasmodium. J. Theor. Biol. 197 (1999) 497–506.

[47] Nakagakia T., Yamada H., Ueda T. Interaction between cell shape and contraction pattern in the *Physarum plasmodium*, Biophysical Chemistry 84 (2000) 195–204.

[48] Nakagaki T., Yamada H., and Toth A., Maze-solving by an amoeboid organism. Nature 407 (2000) 470-470.

[49] Nakagaki T., Smart behavior of true slime mold in a labyrinth. Research in Microbiology 152 (2001) 767-770.

[50] Nakagaki T., Yamada H., and Toth A., Path finding by tube morphogenesis in an amoeboid organism. Biophysical Chemistry 92 (2001) 47-52.

[51] Nakamura S., Yoshimoto Y., Kamiya N. Oscillation in surface pH of the Physarum plasmodium. Proc. Jpn. Acad. 58 (1982) 270–273.

[52] Nakamura S. and Kamiya N. Regional difference in oscillatory characteristics of Physarum plasmodium as revealed by surface pH. Cell Struct. Funct. 10 (1985) 173–176.

[53] Ogihara S. Calcium and ATP regulation of the oscillatory torsional movement in a triton model of Physarum plasmodial strands. Exp. Cell Res. 138 (1982) 377–384.

[54] Oster G. F. and Odel G. M. Mechanics of cytogels I: oscillations in Physarum. Cell Motil. 4 (1984) 469–503.

[55] Prim R.C. Shortest connection networks and some generalizations Bell Syst. Tech. J. 36 (1957) 1389–1401.

[56] Ridgway E. B., Durham A.C.H. Oscillations of calcium ion concentration in Physarum plasmodia, Protoplasma 100 (1976) 167–177.

[57] Rambidi N. G. Neural network devices based on reaction-duffusion media: an approach to artificial retina. Supramolecular Science 5 (1998) 765–767.

[58] Rambidi N.G., Shamayaev K. R., Peshkov G. Yu. Image processing using light-sensitive chemical waves. Phys. Lett. A 298 (2002) 375–382.

[59] Schönhage A. Real-time simulation of multi-dimensional Turing machines by storage modification machines, Project MAC Technical Memorandum 37, MIT (1973).

[60] Schönhage, A. Storage modification machines. SIAM J. Comp. 9 (1980) 490–508.

[61] Sedina-Nadal I, Mihaliuk E, Wang J, Perez-Munuzuri V, Showalter K. Wave propagation in subexcitable media with periodically modulated excitability. Phys Rev Lett 86 (2001) 1646-9.

[62] Shirakawa T. Private communication, Feb 2007.

[63] Shirakawa T. And Gunji Y. -P. Computation of Voronoi diagram and collision-free path using the Plasmodium of Physarum polycephalum. Int. J. Unconventional Computing (2008), in press.

[64] Sielewiesiuk J. and Górecki J., Logical Functions of a Cross Junction of Excitable Chemical Media, J. Phys. Chem., A105, 8189 (2001).

[65] Shvachko K.V. Different modifications of pointer machines and their computational power. In: Proc. Symp. Mathematical Foundations of Computer Science MFCS. Lect. Notec Comput. Sci. 520 (1991) 426–435.

[66] Steinbock O., Toth A., Showalter K. Navigating complex labyrinths: optimal paths from chemical waves. Science 267 (1995) 868–871.

[67] Supowit K.J. The relative neighbourhood graph, with application to minimum spanning tree. J. ACM 3 (1988) 428–448.

[68] Tarjan R. E. Reference machines require non-linear time to maintain disjoint sets, STAN-CS-77-603, March 1977.

[69] Tero A., Kobayashi R., Nakagaki T. A coupled-oscillator model with a conservation law for the rhythmic amoeboid movements of plasmodial slime molds. Physica D 205 (2005) 125-135.

[70] Tirosh R., Oplatka A., Chet I. Motility in a "cell sap" of the slime mold *Physarum Polycephalum*, FEBS Letters 34 (1973) 40–42.

[71] Tóth A., Showalter K. Logic gates in excitable media. J. Chem. Phys. 103 (1995) 2058-2066.

[72] Lyons R. and Peres Y. *Probability on Trees and Networks*, 1997. `http://mypage.iu.edu/~rdlyons/prbtree/prbtree.html`

[73] Tsuda, S., Aono, M., and Gunji, Y.-P., Robust and emergent Physarum-computing. BioSystems 73 (2004) 45–55.

[74] Tsuda, S., Zauner, K. P. and Gunji, Y. P., Robot control: From silicon circuitry to cells, In: Ijspeert, A. J., Masuzawa, T. and Kusumoto, S., Eds. Biologically Inspired Approaches to Advanced Information Technology, Springer, 2006, 20–32.

[75] Turing A., On computable numbers, with an application to the Entscheidungsproblem, Proc. London Mathematical Society, 42 (1936) 230–265.

[76] Uspensky V.A. Kolmogorov and mathematical logic, The Journal of Symbolic Logic 57 (1992) 385–412.

[77] Yamada H., Nakagaki T., Baker R.E., Maini P.K. Dispersion relation in oscillatory reaction-diffusion systems with self-consistent flow in true slime mold. J. Math. Biol. 54 (2007) 745–760.

Computations via Newtonian and relativistic kinematic systems

E.J. Beggs[1] and J.V. Tucker[2]

Swansea University,
Singleton Park,
Swansea, SA2 8PP,
United Kingdom

Abstract

We are developing a rigorous methodology to analyse *experimental computation*, by which we mean the idea of computing a set or function by experimenting with some physical equipment. Here we consider experimental computation by kinematic systems under *both* Newtonian and relativisitic kinematics. An experimental procedure, expressed in a language similar to imperative programming languages, is applied to equipment having a common form, that of a bagatelle, and is interpreted using the two theories. We prove that for any set A of natural numbers there exists a 2-dimensional kinematic system B_A with a single particle P whose observable behaviour decides $n \in A$ for all $n \in \mathbb{N}$. The procedure can operate under (a) Newtonian mechanics or (b) relativistic mechanics. The proofs show how *any* information (coded by some A) can be embedded in the structure of a simple kinematic system and retrieved by simple observations of its behaviour. We reflect on the methodology, which seeks a formal theory for performing experiments that can put physical restrictions on the construction of systems. We conclude with some open problems.

Keywords: foundations of computation; computable functions and sets; Newtonian kinematic systems; Relativistic kinematic systems; foundations of mechanics; theory of Gedanken experiments; non-computable physical systems.

1 Introduction

Consider the idea of computing functions by means of physical systems. Suppose each computation by a physical system is based on *running an experiment* with three stages:

(i) input data x are used to determine initial conditions of the physical system;

(ii) the system operates or evolves for a finite or infinite time; and

(iii) output data y are obtained by measuring the observable behaviour of a system.

The function f computed by a series of such experiments is simply the relation $y = f(x)$. The function may be partial or multivalued, and the data may be continuous or

[1] Department of Mathematics. Email: e.j.beggs@swansea.ac.uk

[2] Department of Computer Science. Email: j.v.tucker@swansea.ac.uk

discrete. We call the idea of using experiments with physical systems to define functions *experimental computation.* This concept of experimental computation is both old and general. It can be found in ideas about (a) technologies for making calculating instruments and machines and (b) modelling physical and biological systems. The concept is also complicated and in need of systematic theoretical investigation.

In contrast, the concept of algorithmic computation is well understood. Computability theory, founded by Church, Turing and Kleene in 1936, is a deep theory for the functions computable by algorithms on discrete and continuous data. The obvious questions arise:

What are the functions computable by experiments with a class of physical systems?

How do they compare with the functions computable by algorithms?

Related questions arise, about novel technologies for computing and about the computability of physical systems.

There is no shortage of results, discussion and debate on particular types of experimental computation. Many examples of non-computable systems are difficult to interpret physically [26, 27, 28, 18, 38]. Some are technically incomplete, and, strictly speaking, have the status of conjectures (e.g., [20]). Some theorems encode non-computability in general classes of mathematical systems (e.g., ODEs in [25]) rather than models of specific physical systems (e.g., wave machines, bagatelles, pendula). Different approaches have led to a diverse literature but the questions are, we believe, open for essentially *all* classes of system.

We are developing a methodology that aims to answer such basic questions in a definitive way [3, 4, 6]. Based on five general principles (summarised later in Section 2), a particular feature of our methodology is the detailed analysis of particular examples, in which we seek the precise *physical* concepts and laws that permit or prevent non-computable functions and behaviours. To do this we choose a precisely defined fragment T of a physical theory to specify, rather formally, the experimental procedure, equipment and reason about its behaviour.

We have illustrated and refined our methodology largely by analysing examples of experimental computations with idealised kinematic systems. Here we will show that there exist simple kinematic procedures and equipment, whose computational behaviour, according to both Newtonian and Relativistic mechanics, can decide the membership of *any* subset A of the set $\mathbb{N} = \{0, 1, 2, \ldots\}$ of natural numbers. The systems are infinite bagatelles that are based on simple energy and momentum conservation principles. They each require unbounded space, time and energy to decide $n \in \mathbb{N}$ for all n. The Newtonian case is simple. The relativistic case might be considered to be more realistic and it also has a useful theoretical property, a *maximum propagation speed* for objects or information, the speed of light c. Instead of unbounded velocity in the Newtonian case, in the relativistic case we exploit the fact that the mass of a particle is unbounded as its speed approaches c.

Theorem 1.1. *Let $A \subseteq \mathbb{N}$. There exists a 2-dimensional kinematic system with a single particle P whose observable behaviour decides A. More specifically, the system is an infinite bagatelle for which the following are equivalent: given any $n \in \mathbb{N}$*

(i) $n \in A$.

(ii) In an experiment, given initial velocity V_n the particle P leaves and returns to the origin within a known time T_n.

The system can be set up to operate under a class of kinematic theories, including
(a) Newtonian kinematics or
(b) Relativistic kinematics.

The velocity V_n and the time T_n are calculated from n and so by simply projecting the particle and watching the clock while waiting for its return, we can decide A. The fact that any conceivable discrete information can be represented in the discrete observable behaviour of a ball rolling along a line suggests that these elementary theories of kinematics are undesirably strong. What should be done with these examples?

The bagatelle uncovers an interesting uniformity or generality. The experimental procedure is essentially the same for any bagatelle and is physically sound. The observation and operation of the bagatelles require rather general assumptions that hold of several kinematic theories. However, it is through the *specification* or *description* of the system that the computation of any A is possible. If the analysis of the experiment concerned not just the observation of an existing system but the process of *assembly* or *construction* of the bagatelle then further conditions on the system would be needed that would restrict the subsets of $\mathbb{N}$. Thus, the bagatelles also reveal that something is missing: they show that a formal account of experimentation must include the specification and construction of mechanical equipment to answer the questions above. A critique of the examples is the subject of Section 6.

In the case of the bagatelle there are certain natural assumptions on experiments that would allow them to compute only the semicomputable and computable subsets of $\mathbb{N}$. Indeed, by choosing $A \subseteq \mathbb{N}$ to be a complete semicomputable set then the construction yields a new universal computer:

Corollary 1.2. *There exists a 2-dimensional kinematic system with a single particle P that is a universal machine for the computable partial functions on $\mathbb{N}$, i.e. the bagatelle computes by experiment all and only the computable partial functions on $\mathbb{N}$.*

The structure of the paper is this. In Section 2 we summarise our methodology. In Section 3 we describe the construction of a general type of infinite bagatelle. In Section 4 we apply the description to make a bagatelle that decides the membership relation for A under Newtonian mechanics, and in Section 5 we apply it to make a bagatelle to decide the membership relation for A under Relativistic mechanics. In Section 6 we reflect on the examples and argue for formal theory of experimentation to answer the questions. Finally, in Section 7 some open problems are discussed.

The reader should be familiar with theory for the functions computable by algorithms on discrete data (Rogers [29], Odifreddi [22], Griffor [16], Stoltenberg-Hansen and Tucker [32]) and continuous data (Pour-El and Richards [28], Tucker and Zucker [35, 36], Weihrauch [37]).

2 Methodological principles

With the idea of experimental computation, we can unify a disparate set of physical models of computation and seek properties they have in common. In particular, we can attempt to analyse physical models of computation *independently of the theory of algorithms.*

Physical theories play a fundamental role in understanding experimental computation; this we have discussed at length elsewhere [3, 4]. To seek conceptual clarity, and mathematical precision and detail, we have proposed, in [3, 4, 6], the following five principles for an investigation of any class of experimental computations:

Principle 1. Defining a physical subtheory: *Define precisely a subtheory T of a physical theory and examine experimental computation by the systems that are valid models of the subtheory T.*

Principle 2. Classifying computers in a physical theory: *Find systems that are models of T that can through experimental computation implement specific algorithms, calculators, computers, universal computers and hyper-computers.*

Principle 3. Mapping the border between computer and hyper-computer in physical theory: *Analyse what properties of the subtheory T are the source of computable and non-computable behaviour and seek necessary and sufficient conditions for the systems to implement precisely the algorithmically computable functions.*

Principle 4. Reviewing and refining the physical theory: *Determine the physical relevance of the systems of interest by reviewing the truth or valid scope of the subtheory. Criticism of the system might require strengthening the subtheory T in different ways, leading to a portfolio of theories and examples.*

Principle 5. Combining experiments and algorithms: *Use a physical system as an oracle in a model of algorithmic computation, such as Turing machines. Determine whether the subtheory T, the experimental computation, and the protocol extends the power and efficiency of the algorithmic model.*

Principles 1-4 were introduced in [3] and Principle 5 was introduced in [6].

To study experimental computation and seek answers to basic questions, the key idea is to lay bare all the concepts and technicalities of examples by putting them under a mathematical microscope using the theory T and, furthermore, to look at the computational behaviour of classes of systems that obey the laws of T. Our methodology requires a careful formulation of a physical theory T, which can best be done by axiomatisations a fragment of the physical theory.

3 Experiments with an infinite bagatelle

We describe the structure of our bagatelle, and the steps involved in using the bagatelle to compute. An important point is that the structural form of the bagatelle, and the experimental procedure to operate it, is common to both the Newtonian and relativistic machines.

3.1 Experiment procedure for the bagatelle

We consider a bagatelle game. A ball is fired into the bagatelle machine with a specified velocity, and the ball may or may not return in a given time period. *Nothing else about the bagatelle is externally observable.*

Each bagatelle machine is designed to define a subset A of the natural numbers $\mathbb{N}$ as follows:

"Given some number $n \in \mathbb{N}$ the operator of B_A chooses a point particle P, positions it at an origin 0 and projects the particle with a velocity V_n. Then the operator waits for a time T_n and if the particle returns before this time then declares that $n \in A$ and otherwise that $n \notin A$."

We can express the experimental procedure in the following experimental pseudocode:

```
exp-pseudocode Bagatelle;
place particle P mass ? radius 0 at point $0$;
start clock $t$;
project particle at $0$ with velocity $V_n$;
wait $T_n$ units and do nothing;
if particle in tray at $0$ then return "$n \in A$" else return "$n \notin A$"
end Bagatelle.
```

In the pseudocode, the particle P is a point particle, with radius 0, but may be of any mass.

Procedures of this kind are an example of "design pattern" called *project and wait*. They can be applied to a number of kinematic systems, since the equipment is not specified; see the bagatelle machines, marble runs, etc. in Beggs and Tucker [2, 3, 4, 5].

To analyse such a procedure further, we can

(a) express the experimental procedure precisely, turning the pseudocode into code;

(b) calculate parameters for the mass and size of particles, and velocities and times;

(c) determine the accuracy of measurements.

Methods for performing (a) are given in [8], where project and wait programs can be found.

The instructions for operating the bagatelle are independent of the mass and size of the particles but are base upon velocities and times. We must calculate a table of velocities

$$V_1, V_2, V_3, \ldots$$

and a table of times

$$T_1, T_2, T_3, \ldots$$

We will see that these tables of numbers are precisely the same for *all* the Newtonian bagatelle machines B_A. Similarly, the lists of velocities and times are the same for all relativistic machines.

The gap between T_n and $T_n + 1$ ensures that we only have to ensure measurement of time to a certain accuracy.

To prove that the experimental procedure works we have to show that each machine can define a subset A of the natural numbers $\mathbb{N}$ as follows: Given $n \in \mathbb{N}$, if a ball is fired into the machine at initial velocity V_n, and the ball returns in a time $\text{Return}(V_n)$. Then

$$\begin{aligned} n \in A \quad &\text{if and only if} \quad \text{Return}(V_n) \le T_n \ , \\ n \notin A \quad &\text{if and only if} \quad \text{Return}(V_n) \ge T_n + 1 \ . \end{aligned} \tag{1}$$

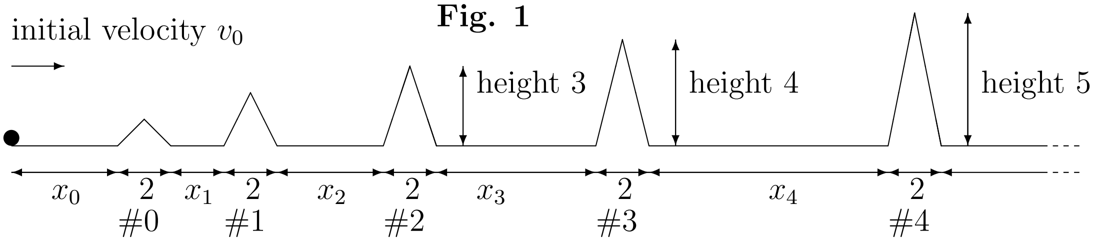
Note that the result can be determined in a finite time $T_n + 1$, even though the ball might never return.

We will do this in a rather generic way that enables us to prove theorems for a class of kinematic theories. Later we apply our analysis to calculating the lists of numbers for the Newtonian and relativistic models.

3.2 Equipment

Structure of the bagatelle If we were to lift the lid on the bagatelle, we would see something like this:

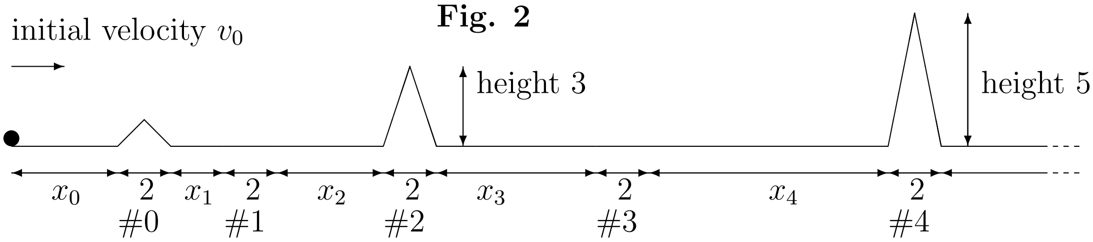
Fig. 1



The machine continues indefinitely off the right hand side. At time $t = 0$ the ball starts from position $x = 0$ with initial velocity v_0. It then crosses, or fails to cross, potential barriers placed in the way along the x-axis. For integer $n \ge 0$ the barrier #n has height $n + 1$ and width 2. For simplicity we assume that it is has the shape of an isoceles triangle. The reader who is anxious about the sharp corners should compute the arbitarily small corrections in the formulae given by introducing arbitrarily small smoothings of the corners. There is a flat gap (at height 0) between #n and #$n + 1$ of length x_{n+1}. We will give the value of the numbers x_n later.

To specify the internal workings of a bagatelle we need a subset A of $\mathbb{N}$. The bagatelle has a potential barrier of height $n + 1$ at position #n if $n \in A$, and a flat track if $n \notin A$. For example, the subset of even natural numbers would correspond to a machine looking like figure 2:

Fig. 2



The reader should note that we suppose that there is no friction or external force acting on the ball. We also assume that the ball is not spinning (or at least that, if it is spinning, that its moment of inertia is zero).

Operation of the bagatelle When a ball hits a potential barrier of height H at velocity v_0, there are three possibilities:

1) It has sufficient energy to cross the barrier, and crosses it in time $C(v_0, H)$ from one base to the other. We assume that $C(v_0, H) \geq 2/v_0$, i.e. that it takes the ball at least as much time to cross the barrier as to travel on a flat track if there is no barrier.

2) It has insufficient energy to cross the barrier, and rolls up and back down in time $B(v_0, H)$ from base to base.

3) It has exactly the right amount of energy to reach the top. We shall take care to avoid this case, as it gives rise to discontinuities in the return time, and the behaviour is critically dependent on the shape of the top of the barrier.

Take V_n to be an initial velocity which ensures that the ball has enough energy to cross all barriers $\#j$ for $j < n$, but that the ball will not cross, but roll back down from $\#n$. Suppose the ball is fired at this velocity on the bagatelle specified by the subset A.

If $n \in A$, then the time of return to the initial point would be

$$\mathrm{Return}(V_n, A) \;=\; \frac{2}{V_n}\Big(\sum_{j\leq n} x_j + \sum_{j<n,\ j\notin A} 2\Big) \;+\; 2\sum_{j<n,\ j\in A} C(V_n, j+1) \;+\; B(V_n, n+1)\ . \quad (2)$$

The first term is given by the ball traversing the flat track at height zero, and the second by the ball crossing over the barriers of height less than n. Remember that both these are done twice, once in either direction. The last term is the time taken for the ball to be reflected from the barrier $\#n$.

However if $n \notin A$, the time of return would be

$$\mathrm{Return}(V_n, A) \;\geq\; \frac{2}{V_n}\Big(\sum_{j\leq n} x_j + \sum_{j<n,\ j\notin A} 2\Big) \;+\; 2\sum_{j<n,\ j\in A} C(V_n, j+1) \;+\; \frac{2\,x_{n+1}}{V_n}\ . \qquad (3)$$

This time is based on the fact that if the ball did return, it would have to travel twice over a flat track of length x_{n+1}. Of course the ball might never return, as there might be no more barriers for it to cross, but this case is included in the inequality.

Choice of the displacements x_n We want an experiment to determine if $n \in A$, and do not want the result confused by other elements of A. However our results (2) and (3) depend on elements in A which are less than n. We deal with this by considering the values taken as we vary A, and choose x_n and T_n to be independent of A: First we choose the sequence $x_n \geq 0$ satisfying the inequalities

$$x_{n+1} \;\geq\; \sum_{j<n}\Big(V_n\, C(V_n, j+1) - 2\Big) \;+\; \frac{V_n(B(V_n, n+1) + 1)}{2}\ . \qquad (4)$$

Definition of the time bounds T_n Then we set T_n by

$$T_n \;=\; \frac{2}{V_n}\sum_{j\leq n} x_j \;+\; 2\sum_{j<n} C(V_n, j+1) \;+\; B(V_n, n+1)\ . \qquad (5)$$

If $n \in A$, remembering that $C(v_0, H) \geq 2/v_0$ we have from (2):

$$\text{Return}(V_n, A) \;\leq\; \frac{2}{V_n}\sum_{j\leq n} x_j \;+\; 2\sum_{j<n} C(V_n, j+1) \;+\; B(V_n, n+1) \;=\; T_n. \qquad (6)$$

Correspondingly for $n \notin A$, from (3) we have

$$\text{Return}(V_n, A) \geq \frac{2}{V_n}\Big(\sum_{j\leq n} x_j + \sum_{j<n} 2\Big) \;+\; \frac{2\,x_{n+1}}{V_n} \;\geq\; T_n + 1\ . \qquad (7)$$

Let us summarise these general calculations.

Theorem 3.1. *For any set A of numbers, let B_A be a bagatelle machine specified above. Let P the experimental procedure for its operation. Let T be a kinematic theory in which*

1. particles follow deterministic paths, traversing, or else being reflected by, the barriers of the bagatelle;

2. conservation of energy ensures the velocity before and after meeting each barrier is the same;

3. formulae can be given for the time of crossing and rolling back barriers, and for initial velocities

$$C(v_0, H) \geq 2/v_0,\ B(v_0, H) \text{ and } V_n.$$

Then one can prove in the kinematic theory T that the procedure P decides membership of the set A.

Condition 1 is not true of quantum kinematics where particles may tunnel through the barrier. the effects of friction are forbidden by Condition 2; the calculations would need to be altered to allow for friction.

It remains to find formulae for C, B and V_n in the Newtonian and relativistic theories, which satisfy the conditions.

3.3 Corollaries

Corollary 3.2. *Any function $f\colon \mathbb{N} \to \mathbb{N}$ can be computed by a Newtonian or Relativistic bagatelle*

Proof. Let G_f be the graph of f. Choose an injective function $c\colon \mathbb{N}^2 \to \mathbb{N}$ such as $(x, y) \mapsto 2^x.3^y$ and code the graph G_f as the set $c(G_f)$. A bagatelle B_A based on $A = c(G_f)$ would enable f to be computed experimentally by the mechanical system. □

Corollary 3.3. *There exist Newtonian and Relativistic bagatelles that are universal machines for the computable partial functions on $\mathbb{N}$, i.e. the bagatelles compute by experiment all and only the computable partial functions on $\mathbb{N}$.*

Proof. Choose a bagatelle B_A based on $A = c(G_U)$, the coded graph of a universal partial recursive function U. This would enable U to be computed experimentally by the mechanical system. □

4 Newtonian kinematics with constant gravitational field

The initial kinetic energy of the ball of mass m with any initial velocity v_0 is $\frac{1}{2}mv_0^2$. The potential energy of the ball at height h above the initial point is mgh, where g is the acceleration due to gravity (on the Earth's surface, this is about $9{\cdot}8$ meters/second^2). The principle of conservation of energy then gives the velocity v of the ball at a height h using $\frac{1}{2}mv_0^2 = \frac{1}{2}mv^2 + mgh$. It follows that the maximum height H that the ball can attain is given by $\frac{1}{2}mv_0^2 = mgH$, i.e. $H = \frac{1}{2}v_0^2/g$. We set V_n to be the initial velocity for which the maximum attainable height is $n + \frac{1}{2}$, i.e.

$$V_n \;=\; \sqrt{g(2n+1)}\;. \tag{8}$$

Proposition 4.1. *The time taken for a ball with initial velocity v_0 to climb a slope of gradient n to a height h (less than the maximum height $\frac{1}{2}v_0^2/g$) is*

$$\frac{v_0 - \sqrt{v_0^2 - 2gh}}{g}\sqrt{1+\frac{1}{n^2}}\;.$$

Proof. We start the slope at the point $(x, y) = (0, 0)$, so the equation of the slope is $y = nx$. On rearranging the conservation of energy equation, we see that at height y the particle has velocity $v = \sqrt{v_0^2 - 2gy}$. The length of slope from height y to $y + dy$ is given by Pythagoras' theorem as $\sqrt{(dx)^2 + (dy)^2}$, or using the equation $y = nx$, as $dy\,\sqrt{1+n^2}/n$. The time taken to move from height y to $y + dy$ is the distance divided by the velocity, or $dy\,\sqrt{1+n^2}/(nv)$. This gives the total time to climb to height h as the integral

$$\int_{y=0}^{h}\frac{dy\,\sqrt{1+n^2}}{n\,\sqrt{v_0^2-2gy}} \;=\; \frac{v_0-\sqrt{v_0^2-2gh}}{g}\sqrt{1+\frac{1}{n^2}}\;. \quad \square$$

Corollary 4.2. *The time taken for a ball with initial velocity v_0 to climb a slope of gradient n to its maximum attainable height is*

$$\frac{v_0}{g}\sqrt{1+\frac{1}{n^2}}\;.$$

Corollary 4.3. *Using the definition of V_n in (8), we have, for $j \le n$,*

$$\begin{aligned} C(V_n, j) \;&=\; 2\,\frac{\sqrt{2n+1}-\sqrt{2n-2j+1}}{\sqrt{g}}\sqrt{1+\frac{1}{j^2}}\;, \\ B(V_n, n+1) \;&=\; 2\,\frac{\sqrt{2n+1}}{\sqrt{g}}\sqrt{1+\frac{1}{(n+1)^2}}\;. \end{aligned}$$

Proof. We use the formulae given in 4.1 and 4.2, remembering that it takes the same time to roll down as to climb up. $\square$

Remark 4.4. *Here we calculate asymptotic bounds on the time taken by the Newtonian bagatelle to decide if $n \in A$ or not. From (8) and 4.3 we see that V_n, $C(V_n, j)$ and $B(V_n, n+1)$ are all $O(\sqrt{n})$. From (4) we can choose x_n to be $O(n^2)$, and from (5) we have T_n to be $O(n^{5/2})$.*

5 Relativistic kinematics with constant gravitational field

The relativistic mass of a ball of rest mass m travelling at velocity v is $M = m/\sqrt{1-v^2/c^2}$, where c is the speed of light. The momentum of the ball is Mv, and we use the usual formula that force is the rate of change of momentum. On a slope inclined at an angle α to the horizontal, we have $\frac{d}{dt}(Mv) = -Mg\sin(\alpha)$. On rearranging and differentiating this yields $\frac{dv}{dt} = -g(c^2-v^2)\sin(\alpha)/c^2$. On integrating we get

$$v \;=\; c\tanh(g(b-t)\sin(\alpha)/c)\ , \tag{9}$$

where b is a constant. The initial velocity is

$$v_0 \;=\; c\tanh(gb\sin(\alpha)/c)\ , \tag{10}$$

which, using a hyperbolic trig identity, becomes the useful formula

$$\cosh(gb\sin(\alpha)/c) = 1/\sqrt{1-v_0^2/c^2}\ . \tag{11}$$

The distance travelled along the slope as a function of time is given by integrating (9)

$$d \;=\; \frac{c^2}{g\,\sin(\alpha)}\,\log\Big(\frac{\cosh(bg\sin(\alpha)/c)}{\cosh((b-t)g\sin(\alpha)/c)}\Big)\ ,$$

so the height as a function of time is

$$h \;=\; \frac{c^2}{g}\,\log\Big(\frac{\cosh(bg\sin(\alpha)/c)}{\cosh((b-t)g\sin(\alpha)/c)}\Big)\ . \tag{12}$$

The maximum height achieveable occurs when $t = b$, and is

$$h_{max} \;=\; \frac{c^2}{g}\,\log\big(\cosh(bg\sin(\alpha)/c)\big)\ . \tag{13}$$

If the maximum height is set to $n+\frac{1}{2}$, then using (11) and (13) the corresponding initial velocity V_n is given by

$$V_n \;=\; c\sqrt{1-e^{-(2n+1)g/c^2}}\ . \tag{14}$$

Proposition 5.1. *The time taken for a ball with initial velocity v_0 to climb a slope of gradient $\sin\alpha$ to a height h (less than the maximum height) is*

$$\frac{c}{g\,\sin\alpha}\Big(\tanh^{-1}\Big(\frac{v_0}{c}\Big) - \cosh^{-1}\Big(\frac{e^{-gh/c^2}}{\sqrt{1-v_0^2/c^2}}\Big)\Big)\ .$$

Proof. If we rearrange (12) we get

$$\cosh((b-t)g\sin(\alpha)/c) \;=\; \cosh(bg\sin(\alpha)/c)\,e^{-gh/c^2}\ ,$$

so we get t as

$$t \;=\; b - \frac{c}{g\,\sin\alpha}\,\cosh^{-1}\Big(\cosh(bg\sin(\alpha)/c)\,e^{-gh/c^2}\Big)\ . \quad \square$$

Corollary 5.2. *The time taken for a ball with initial velocity v_0 to climb a slope of gradient* $\sin\alpha$ *to its maximum attainable height is*

$$\frac{c}{g\,\sin\alpha}\,\tanh^{-1}\left(\frac{v_0}{c}\right)\,.$$

Corollary 5.3. *Using the definition of V_n in (14), we have, for $j \le n$,*

$$\begin{aligned} C(V_n,j) &= \frac{2\,c\,\sqrt{1+j^2}}{g\,j}\left(\cosh^{-1}\left(e^{(2n+1)g/(2c^2)}\right)-\cosh^{-1}\left(e^{(2n+1-2j)g/(2c^2)}\right)\right)\,, \\ B(V_n,n+1) &= \frac{2\,c\,\sqrt{1+(n+1)^2}}{g\,(n+1)}\,\cosh^{-1}\left(e^{(2n+1)g/(2c^2)}\right)\,. \end{aligned}$$

Proof. We use 5.1 and 5.2, with (11) and (14) supplying the formula

$$\cosh(bg\sin(\alpha)/c) \;=\; e^{(2n+1)g/(2c^2)}\,. \quad \square$$

Remark 5.4. *Here we calculate asymptotic bounds on the time taken by the relativistic bagatelle to decide if $n \in A$ or not. From 5.3 we see that $C(V_n,j)$ and $B(V_n,n+1)$ are both $O(n)$. For n large, $V_n \cong c$. From (4) we can choose x_n to be $O(n^2)$, and from (5) we have T_n to be $O(n^3)$.*

6 Commentary on the Bagatelle

6.1 Genericity and extensions

Operationally, the experiments on a bagatelle B_A, needed to decide $n \in A$, can be carried out using the following *primitive experimental actions*:

(i) project a particle with arbitrary large energy (for arbitrary large natural numbers);
(ii) observe a fixed point in space;
(iii) measure arbitrarily large times on a clock; and
(iv) calculate with simple algebraic formulae.

Indeed, these actions are the starting point of almost any kinematic theory. In specifying the *forms* of both the experimental procedure and the equipment, we find we do *not* have to choose a *particular* kinematic theory. Indeed, the computation can be *verified* in a generic form for a class of theories, as in Theorem 3.1. Later, we choose Newtonian and relativistic theories.

From the point of view of formal languages to express experimental procedures, as in [8], one may liken the situation to formulating classes and reasoning about possible late bindings of semantics to syntax at runtime.

Technically, the experimental computations by bagatelles can be extended to further theories:

Friction. To include friction, we would have to have a particular equation for the force due to friction (probably as a function of velocity). Then we would have to alter the formulae given, possibly including the height of the barriers, increasing the the initial energy to compensate for the energy lost due to friction. The procedure would remain the same though we would need different tables of velocities and times.

Gravitational field. Varying the gravitational field as a function of height would alter the potential energy, and again would require altering the formulae. Also the potential should not decrease with height, otherwise the inequality $C(v_0, H) \geq 2/v_0$ might be compromised.

6.2 Constructible equipment

The computations by the bagatelles are valid in several kinematic theories. Clearly, the structure of the bagatelle B_A is based on the set A but there is nothing in mechanics that prevents, or even cautions, us from defining and reasoning about such systems for *any* set A: according to the theories, the bagatelle B_A is a legal kinematic system, a model of the theories.

However, suppose the account of the experiment is required *to explain how the mechanical system is constructed*, as well as what primitive experimental actions are needed to set initial states and observe behaviour. Then we find we have an interesting question:

What precise assumptions underly our idea of equipment?

Obviously, the sequence of primitive steps in the construction of the system B_A will involve knowledge of the set A. This knowledge is (probably) precisely the knowledge the system B_A is being designed to reveal, making the *purpose* of the experiment redundant. But since we are interested in the nature experimental computation, this point about redundancy is not relevant; the important point is:

What conditions will be required on A to allow experiments on B_A that are valid in a kinematic theory containing principles of constructible equipment?

Suppose we re-consider running the experiment, seeking to restrict A and narrow the class of models.

Suppose A is given by some increasing enumeration $A_0, A_1, A_2, \ldots$ of finite subsets with for each $i \in \mathbb{N}$, $A_i \subset A_{i+1}$ and $A = \bigcup_{i \in \mathbb{N}} A_i$. Suppose that A_i has i elements of A. Then to make an experiment to decide if $n \in A$ then we need an experimental procedure with a process to construct a finite part of the bagatelle. This finite part will have the form B_{A_k} for some $A_k \subset A$. It will have k potential barriers located by the k elements of A_k.

An independent laboratory clock will schedule the construction of the approximating bagatelle B_{A_k} and the experimental procedure to decide $n \in A_k$. If the experiment confirms that $n \in A_k$ then we know that $n \in A$.

However, if the experiment confirms that $n \notin A_k$ then we do *not* know that $n \notin A$. This result can change as k increases and more and more elements of A appear and the bagatelle grows. Each negative result must be repeated and so the experiment becomes a *search* for a positive result, secure in the knowledge that if $n \in A$ then an experiment with some part B_{A_k} of the bagatelle B_A will find it.

Thus, when we include the construction of the bagatelle in the primitive steps we have an outline proof that

A is decidable by experiment with constructible equipment if, and only if, the finite subsets of the enumeration of A can be generated by experiment.

How close are we to a proof that A is decidable by experiment with constructible equipment if, and only if, A is recursively enumerable subset of $\mathbb{N}$?

Allowing computability theory to be added to the physical theory would resolve the matter. Experimental computation by the bagatelles in kinetic theories with constructible equipment, where the equipment is specified by algorithms, does not lead to non-computable sets and functions.

6.3 General principles

The bagatelle, and several of our earlier examples [3, 4], show that the notion of equipment needs to be analysed. We have proposed

Experimental computation = Experimental procedure + Equipment.

A formal theory is needed that constrains the architecture and construction of mechanical systems and formally defines notions of constructible equipment. We have outlined in [4] the problems of designing formal languages for the specification of constructible equipment, and of combining them with languages for experimental procedures to make complete languages for experimental computation. In [8], we have proposed a set of constructs for a programming language for experimental procedures for Newtonian kinematics. The bagatelle adds gives us further insight into this approach using languages.

First, consider the role of languages in digital computation. We need the equation:

Digital computation language = Programming language + Hardware description language.

This is unfamiliar in programming language theory because the primary aim of programming is to formulate algorithms that are *independent of machines and devices.* Both languages have a syntax and a semantics. Theories of syntax are very general, and are widely understood and used. Semantic methods for programming languages are also well developed, though the semantics of hardware description languages is less so. This view of digital computation is complemented by this equation:

Experimental computation language = Experimental procedure language + Equipment description language.

Consider the distinction between syntax and semantics for experimental computation. The syntax of the languages can be handled by general methods. The bagatelle example shows how semantics is more complicated. Somewhat abstractly, we have a Gedanken experiment, in which we imagine an operator following the procedure in working with the equipment. What happens and, in particular, what is computed? To answer this we make some meta-assumptions and only later choose a theory - Newton's or Einstein's. The physical theories are used to define the semantics of the experimental procedure and equipment in order to reason about, predict etc., the behaviour of the system for the benefit of the operator. The bagatelle shows that various semantics are possible for the class of experimental procedures. Indeed, one is reminded of late binding and runtimes.

The importance of the theory cannot be underestimated. If the theories used for semantics can be axiomatically specified then we have sounder and sharper reasoning. We also have the phenomenon of unexpected models. It is just as important that we can reason about the bagatelle as we can satisfy ourselves that we can plausibly implement it. Clearly, both kinematic theories apply to experimental computations that we suspect are impossible. It is a problem to prove they are impossible.

One thinks of the first order theory of Peano arithmetic. It is a truly fundamental and useful theory for reasoning about natural numbers. It has infinitely many different (=non-isomorphic) models but only those isomorhic to the standard model $(\mathbb{N}|0, n+1, n+m, n.m, =)$ are algorithmically computable (Tennenbaum's Theorem).

We expect that research on languages for experimental computation will lead to new ideas and techniques in programming language theory.

7 Concluding remarks on kinematic systems

Experimental computation is not well understood and many basic questions are open. There is a paucity of examples that can be formulated and studied in *complete detail*, though plenty of informal ideas and speculations have been aired. The purpose of our methodology is explore experimental computation *systematically* and *precisely.* We are in no rush to make physically plausible pronouncements or critiques, rather we propose to study examples in forensic detail. Examples based on kinematics, possibly the simplest physical theory, offer insights into experimental computation, and pose interesting and difficult theoretical problems.

Our bagatelles are systems that each require unbounded space, time and energy to decide $n \in A$ for all $n \in \mathbb{N}$. Consider energy. In each of our Newtonian bagatelles mass is bounded (indeed, it can be an arbitrary constant) and velocity is unbounded. In each of our relativistic bagatelles velocity is bounded and mass is unbounded. One can ask if there are examples of kinematic systems that are bounded in space, time and energy?

In Newtonian mechanics we are allowed to shrink space and accelerate time. For example, the natural numbers $n = 0, 1, 2, \ldots$ that mark points in space or steps in time can be embedded into the interval $[0, 1]$ by $n \mapsto 1/2^n$. Shrinking space leads to mechanical systems that use arbitrarily small components. Of course, a mechanical system that exploits the infinite divisibility of space, with no *lower* bounds on units of space and time, violates any form of atomic theory. But such examples can be precision tools to investigate the theoretical foundations of computability and mechanics. For instance, it is possible to prove that for each set $A \subset \mathbb{N}$ there exists a valid Newtonian kinematic system S_A, which is embedded within a bounded 3-dimensional box, operates entirely within a fixed finite time interval using a fixed finite amount of energy, and can decide the membership of the subset A ([4]). The fact that *any* subset of the natural numbers can be recognised by a simple kinematic system raises an alarm because the theory of the subsets of natural numbers is so vastly complicated it depends on the foundations of set theory for its exploration.

An open problem is this:

Problem 7.1. *For all reasonable kinematic theories T, and all T-valid kinematic systems that possess both lower and upper bounds on space, time, mass, velocity and energy, are the sets and functions computable by experiment also computable by algorithms?*

We conjecture that the answer is "Yes". To attempt to prove this, one needs formal descriptions of equipment as mentioned in Section 6. Our bagatelle examples show that the notion of mechanical system - i.e., what qualifies as a valid or legal system in theoretical mechanics - must be sharpened. To the standard parameters of mass, velocity, distance, time we need to add formal theory that constrains the structure and construction of the equipment and explains how experiments are performed.

Theoretical intuitions about making experiments turn out to be strikingly similar to intuitions about algorithms and computers, although the primitive actions are different and are implicit in the physical theory. Indeed, we conjecture that a theory of Gedanken experiments for mechanics, if formalised, could be capable of underpinning the theory of the computable as follows:

Problem 7.2. *Extend theoretical kinematics by a mathematical theory of construction and the operation of mechanical equipment, and show that the sets and functions computable by experiment are precisely those computable by algorithms.*

This is a difficult problem which we have discussed in detail in [4]. More recent work, in [5], shows that a kinematic device called a *scatter machine* can compute any real number. The construction of the device is bounded in all respects however the theorem demonstrates that the existence of sharp corners in objects is fatal to computablity! One goal of this direction of research, from physical theory to computablity, is, roughly speaking, *To derive forms of Church-Turing Thesis as physical laws.* However, we can still ask the question:

Problem 7.3. *Does experimental computation in theories with constructible equipment, where the equipment is specified algorithmically, lead to algorithmically computable sets and functions?*

We conjecture yes. Finally, we should raise the special case of *efficient* computation by mechanical systems. New theory is needed to pose and answer a question such as:

Problem 7.4. *Are there sets that can be decided in polynomially bounded space and time by experimental computation with kinematic systems but cannot be decided by algorithms in polynomial space and time?*

We thank Brendan Larvor, David Miller, and J I Zucker for discussions on some of the areas mentioned in this paper.

References

[1] J Alper and M Bridger, Newtonian supertasks: A critical review, *Synthese* 114 (1998), 355 – 369.

[2] E J Beggs and J V Tucker, Computations via experiments with kinematic systems, Research Report 4.04, Department of Mathematics, University of Wales Swansea, March 2004 or Technical Report 5-2004, Department of Computer Science, University of Wales Swansea, March 2004.

[3] E J Beggs and J V Tucker, Embedding infinitely parallel computation in Newtonian kinematics, *Applied Mathematics and Computation*, 178 (2006) 25–43.

[4] E J Beggs and J V Tucker, Can Newtonian systems, bounded in space, time, mass and energy compute all functions?, *Theoretical Computer Science*, 371 (2007) 4–19.

[5] E J Beggs and J V Tucker, Experimental computation of real numbers by Newtonian machines, *Proceedings Royal Society Series A*, 463 (2007) 1541–1561.

[6] E J Beggs, J F Costa, B Loff and J V Tucker, Computational complexity with experiments as oracles, in preparation, 2008.

[7] E J Beggs, J F Costa, B Loff, and J V Tucker, The complexity of measurement in classical physics, in *Theory and Applications of Models of Computation*, Lecture Notes in Computer Science, Vol 4978, Springer, 2008, in press.

[8] E J Beggs and J V Tucker, Programming experimental procedures for Newtonian kinematic machines, A Beckmann et al (eds), —emphComputability in Europe, Athens, 2008, Springer Lecture Notes in Computer Science, 2008, in press.

[9] E J Beggs, J F Costa and J V Tucker, Oracles and advice as measurements, in preparation.

[10] V Bush, The differential analyser. A new machine for solving differential equations, *Journal of Franklin Institute* 212 (1931), 447-488.

[11] M Davis, The myth of hypercomputation, in *Turing Festschrift*, Springer, in preparation.

[12] R Geroch and J B Hartle, Computability and physical theories, *Foundations of Physics* 16 (1986), 533-550.

[13] D S Graça and J F Costa, Analog computers and recursive functions over the reals, 2003, submitted.

[14] D R Hartree, *Calculating Instruments and Machines*, Cambridge University Press, 1950.

[15] A V Holden, J V Tucker, H Zhang and M Poole, Coupled map lattices as computational systems, *American Institute of Physics - Chaos* 2 (1992), 367-376.

[16] E Griffor, (ed) *Handbook of Computability Theory*, Elsevier, 1999.

[17] G Kreisel, A notion of mechanistic theory, *Synthese* 29 (1974), 9-24.

[18] G Kreisel, Review of Pour El and Richards, *Journal of Symbolic Logic* 47 (1974), 900-902.

[19] L Lipshitz and L A Rubel, A differentially algebraic replacement theorem, *Proceedings of the American Mathematical Society* 99(2) (1987), 367 – 372.

[20] C Moore, Unpredictability and undecidability in dynamical systems, *Physical Review Letters* 64 (1990), 2354 – 2357.

[21] C Moore, Recursion theory on the reals and continuous time computation, *Theoretical Computer Science* 162 (1996), 23 – 44.

[22] P Odifreddi, *Classical Recursion Theory*, Studies in Logic and the Foundations of mathematics, Vol 129, North-Holland, Amsterdam, 1989.

[23] J Perez Laraudogoitia, A beautiful supertask, *Mind* 105 (1996), 81 – 83.

[24] J Perez Laraudogoitia, Infinity machines and creation ex nihilo, *Synthese* 114 (1998), 259 – 265.

[25] M B Pour-El and J. I. Richards, A computable ordinary differential equation which possesses no computable solution, *Annals of Mathematical Logic* 17 (1979), 61 – 90.

[26] M B Pour-El and J. I. Richards, The wave equation with computable initial data such that its unique solution is not computable, *Advances in Mathematics* 39 (1981), 215 – 239.

[27] M B Pour-El and J. I. Richards, Computability and noncomputability in classical analysis, *Transactions of the American Mathematical Society* 275 (1983), 539 – 560.

[28] M B Pour-El and J. I. Richards, *Computability in Analysis and Physics*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin, 1989.

[29] H Rogers, *Theory of recursive Functions and Effective Computability*, McGraw-Hill, New York, 1967.

[30] C Shannon, Mathematical theory of the differential analyser, *Journal of Mathematics and Physics* 20 (1941), 337-354.

[31] H T Siegelmann, *Neural networks and analog computation: Beyond the Turing limit*, Birkhäuser, Boston, 1999.

[32] V Stoltenberg-Hansen and J V Tucker, Effective algebras, in S Abramsky, D Gabbay and T Maibaum (eds.) *Handbook of Logic in Computer Science. Volume IV: Semantic Modelling*, Oxford University Press, 1995, pp.357-526.

[33] V Stoltenberg-Hansen and J V Tucker, Computable rings and fields, in E. R. Griffor (ed.), *Handbook of Computability Theory*, Elsevier, 1999, 363 – 447.

[34] V Stoltenberg-Hansen and J V Tucker, Computable and continuous partial homomorphisms on metric partial algebras, *Bulletin for Symbolic Logic* 9 (2003), 299 – 334.

[35] J V Tucker and J I Zucker, Computable functions and semicomputable sets on many sorted algebras, in S. Abramsky, D. Gabbay and T Maibaum (eds.), *Handbook of Logic for Computer Science* volume V, Oxford University Press, 2000, 317 – 523.

[36] J V Tucker and J I Zucker, Abstract versus concrete computation on metric partial algebras, *ACM Transactions on Computational Logic*, 5 (2004) 611–668.

[37] K Weihrauch, *Computable Analysis, An introduction*, Springer-Verlag, Heidelberg, 2000.

[38] K Weihrauch and N Zhong, Is wave propogation computable or can wave computers beat the Turing machine?, *Proceedings of London Mathematical Society* 85 (2002), 312-332.

The Influence of Domain Interpretations on Computational Models[⋆]

Udi Boker and Nachum Dershowitz

School of Computer Science, Tel Aviv University
Ramat Aviv, Tel Aviv 69978, Israel

Abstract

Computational models are usually defined over specific domains. For example, Turing machines are defined over strings, and the recursive functions over the natural numbers. Nevertheless, one often uses one computational model to compute functions over another domain, in which case, one is obliged to employ a representation, mapping elements of one domain into the other. For instance, Turing machines (or modern computers) are understood as computing numerical functions, by interpreting strings as numbers, via a binary or decimal representation, say.

We ask: Is the choice of the domain interpretation important? Clearly, complexity is influenced, but does the representation also affect computability? Can it be that the same model computes strictly more functions via one representation than another? We show that the answer is "yes", and further analyze the influence of domain interpretation on the extensionality of computational models (that is, on the set of functions computed by the model).

We introduce the notion of *interpretation-completeness* for computational models that are basically unaffected by the choice of domain interpretation, and prove that Turing machines and the recursive functions are interpretation-complete, while two-counter machines are incomplete. We continue by examining issues based on model extensionality that are influenced by the domain interpretation. We suggest a notion for comparing computational power of models operating over arbitrary domains, as well as an interpretation of the Church-Turing Thesis over arbitrary domains.

Key words: domain interpretation, domain representation, hypercomputation, Turing machine, computability, computational power, computational models, computational comparison

1. Introduction

We explore the problem of the sensitivity of models to domain interpretation, and the way we propose to handle it. This introductory section parallels the structure of the paper, as illustrated in Figure 1.

Sensitivity to domain interpretation. A computational model is defined over a specific domain. However, we may often use it to compute functions over a different domain. For example, using Turing machines (or modern computers) to compute functions over natural numbers requires a string representation by numbers. Another example becomes apparent when comparing the computational power of two models that operate

[⋆] This work was carried out in partial fulfillment of the requirements for the Ph.D. degree of the first author. Research was supported in part by the Israel Science Foundation under grant no. 250/05.

Email address: {udiboker,nachumd}@tau.ac.il (Udi Boker and Nachum Dershowitz).

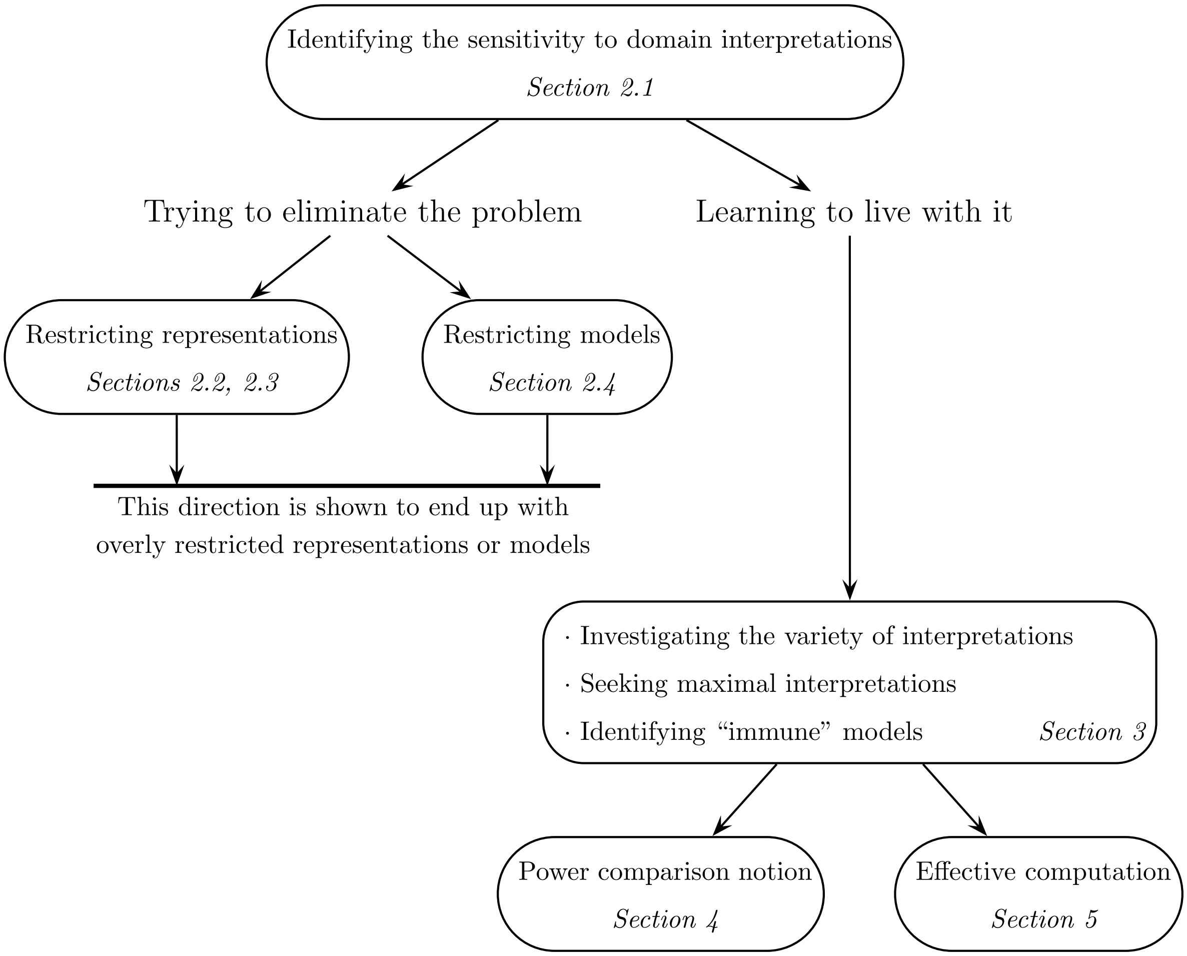

Identifying the sensitivity to domain interpretations
Section 2.1
Trying to eliminate the problem
Learning to live with it
Restricting representations
Sections 2.2, 2.3
Restricting models
Section 2.4
This direction is shown to end up with overly restricted representations or models
· Investigating the variety of interpretations
· Seeking maximal interpretations
· Identifying "immune" models
Section 3
Power comparison notion
Section 4
Effective computation
Section 5

Fig. 1. Structure of this paper

over different domains – we are obliged to represent the domain of one model by the domain of the other. Accordingly, the elements of the original domain are interpreted as elements of the target domain (see illustration in Figure 3). A "representation" is usually allowed to be any mapping from one domain into another, as long as it is injective. That is, every original domain element is mapped to a unique element of the second domain.[1]

It turns out that interpreting the domain allows for the possibility that a model be identified with one of its (strict) supermodels. The interpretation might allow one to "enlarge" the extensionality of a model, adding some "new" functions to it. A study of the sensitivity of models to interpretations via injective representations is undertaken in Section 2.1.

A reasonable response might be to restrict representations to bijections between domains. However, it turns out that there are models that can be identified with a supermodel even with bijective representations. That is, their extensionality is isomorphic to the extensionality of some of their strict supermodels. The case of bijective representations is explored in Section 2.2.

If bijective representations are not stringent enough, which representations are guaranteed not to influence the extensionality of computational models? It turns out that only very limited representations are, namely, "narrow permutations" (Definition 7). A sufficient and necessary criterion for these "harmless representations" is given in Section 2.3. Not only are narrow permutations a very limited family of representations,

[1] A representation may also be a relation (rather than a function), as long as two entities representing the same element behave the same in the relevant context (see, for example, [26]). The generalization of representations to relations does not eliminate the sensitivity of models to the domain interpretation.

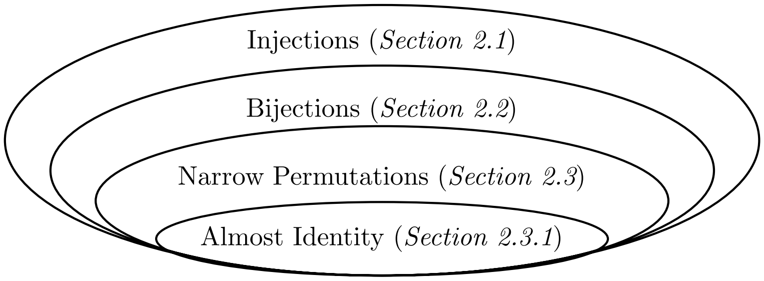

Injections (*Section 2.1*)
Bijections (*Section 2.2*)
Narrow Permutations (*Section 2.3*)
Almost Identity (*Section 2.3.1*)

Fig. 2. The hierarchy of mappings involved in seeking harmless representations.

they are also not closed under composition. Thus, seeking harmless representations for comparative purposes would lead to an even more limited family of permutations that are almost the identity (Definition 10). Hence, sticking to harmless representations is not a viable option, as it almost completely evaporates the concept of interpretations. A scheme of the families of representations involved is given in Figure 2.

Another direction for avoiding the influence of representations could be narrowing down the definition of a computational model, insisting – for example – that the set of computed functions is closed under functional composition. It turns out that these standard computational properties are insufficient, as shown in Section 2.4.

The implications. This sensitivity to the domain interpretation places a question mark on some of the main issues that concern model extensionality: How can we compare models over different domains? How should one define effective computation over arbitrary domains? How should one properly represent the natural numbers? Is there always an optimal representation? Are some models immune to the influence of the domain interpretation? These problems are briefly answered below, and more comprehensively addressed in Sections 3, 4 and 5.

Organizing model interpretations. Generally, the various interpretations of a model may be highly varied: they may be larger or smaller than the original; for some models there are maximal interpretations, while for others there are none; and there are models that are already maximal. This variety of possibilities is explored in Section 3.1.

The last property, that a model is already maximally interpreted, is the one that interests us most. We call such a model "interpretation-complete". We also define a weaker property, denoted "interpretation-stable", saying that a model is maximal with respect to bijective representations. When allowing only bijective representations, there are exactly two ways in which the interpretation may influence the model's extensionality: stable models are totally immune, in the sense that they have no better or worse interpretations via bijective representations, while unstable models have no maximum, nor minimum, interpretation via bijective representations (see the illustration in Figure 7). When allowing non-bijective representations, the picture is different – there might be a complete model with interpretations that are strictly contained in its original extensionality (see Figure 8). Interpretation-completeness and interpretation-stability, as well as some means for getting maximal interpretations, are investigated in Section 3.2.

In Section 3.3, we check for the completeness of some standard models. Turing machines and the recursive functions are shown to be complete, while two-counter machines and the untyped lambda calculus (over all lambda terms) are incomplete. As for hypercomputational models, they might be incomplete, though those preserving the closure properties of the recursive functions are ensured to be interpretation-complete.

Comparing computational power. It is common practice to compare the computational power of different models of computation. For example, the recursive functions are considered to be strictly more powerful than the primitive recursive functions, because the latter are a proper subset of the former, which includes Ackermann's function (see, for example, [22, p. 92]). Side-by-side with this "containment" method of measuring power, it is also standard to base comparisons on interpretations over different domains [7,18,21].

For example, one says that the (untyped) lambda calculus is as powerful—computationally speaking—as the partial recursive functions, because the lambda calculus can compute all partial recursive functions by encoding the natural numbers as Church numerals.

The problem is that unbridled use of these two distinct ways of comparing power allows one to show that some computational models are *strictly* stronger than themselves!

We define a comparison notion over arbitrary domains based on model interpretations. With this notion, model B is strictly stronger than A if B has an interpretation that contains A, whereas A cannot contain B under *any* interpretation. We provide, in Section 4.1, three variants of the comparison notion, depending on the allowed interpretations. In Sections 4.1.1 we extend the notion to non-deterministic models. We continue, in Section 4.2, with some results on the relations between power comparison, isomorphism and completeness. In Section 4.3, we use the notion to compare some standard models.

We deal here only with the mathematical aspects of power comparison. Some conceptual discussions and justifications can be found in [2,4].

Effective computation. Let f be some decision function (a Boolean-valued function) over an arbitrary countable domain D. What does one mean by saying that "f is computable"? One most likely means that there is a Turing machine M, such that M computes f, *using some string representation of the domain* D. But what are the allowed string representations? Obviously, allowing an arbitrary representation (any injection from D to Σ^*) is problematic – it will make any decision function "computable". For example, by permuting the domain of machine codes, the halting function can morph into the simple parity function, which returns true when the input number is even, representing a halting machine, and false otherwise. Thus, under a "strange" representation the function becomes eminently "computable" (see Section 5.1). Another approach is to allow only "natural" or "effective" representations. However, in the context of defining computability, one is obliged to resort to a vague and undefined notion of "naturalness" or of "effectiveness", thereby defeating the very purpose of characterizing computability.

Our approach to overcoming the representation problem is to ask about effectiveness of a set of functions over the domain of interest, rather than of a single function (Section 5.1). As Myhill observed [15], undecidability is a property of *classes* of problems, not of individual problems. In this sense, the halting function is undecidable in conjunction with an interpreter (universal machine) for Turing machine programs that uses the same representation. The Church-Turing Thesis, interpreted accordingly, asserts that there is no effective computational model that is more inclusive than Turing machines.

Nonetheless, there might have been a serious problem due the sensitivity of models to the domain interpretation (see Section 2). Fortunately, this cannot be the case with Turing machines (nor with the recursive functions), as they are interpretation-complete (Theorem 24). Hence, the Church-Turing Thesis is well-defined for arbitrary computational models.

Due to this completeness of Turing machines, we can also sensibly define what it means for a string representation of a constructible domain to be "effective" (Section 5.2). Such a representation ρ is effective when the domain's constructing functions are Turing computable via ρ (Definition 48). Hence, one may ask about the effectiveness of a single function over a constructible domain, provided that the means of construction of the domain are defined and are computable.

Equipped with a plausible interpretation of the Church-Turing Thesis over arbitrary domains, one may investigate the general class of "effective computational models". This is done in [5].

Previous work. Usually, the handling of multiple domains in the literature is done by choosing specific representations, like Gödel numbering, Church numerals, unary representation of numbers, etc. This is also true of the usual handling of representations in the context of the Church-Turing Thesis.

A more general approach for comparing the power of different computational models is to allow any representation based on an injective mapping between their domains. This is done, for example, by Rogers [18, p. 27], Sommerhalder [21, p. 30], and Cutland [7, p. 24]. A similar approach is used for defining the effectiveness of an algebraic structure by Froehlich and Shepherdson [8], Rabin [16], and Mal'cev [11]. Our notion of comparing computational power is very similar to this.

Richard Montague [13] raises the problem of representation when applying Turing's notion of computability to other domains, as well as the circularity in choosing a "computable representation".

Stewart Shapiro [20] raises the very same problem of representation when applying computability to number-theoretic functions. He suggests a definition of an "acceptable notation" (string representation of natural numbers), based on some intuitive concepts. We discuss his notion in Section 5.1 and 5.2.

Klaus Weihrauch [25,26] deals heavily with the representation of arbitrary domains by numbers and strings. He defines computability with respect to a representation, and provides justifications for the effectiveness of the standard representations. We elaborate on his justifications in Section 5.2.

Our definition of an "effective representation" resembles Shapiro's notion of "acceptable notation" and goes along the lines of Weihrauch's justifications for the effectiveness of the standard representations.

To the best of our knowledge, our work in [2–4] was the first to point out and handle the possible influence of the representation on the extensionality of computational models. Sections 2, 3, and 4 organize the main results of these papers, while adding some new ones, particularly Proposition 14, Proposition 17, Theorem 21, Theorem 25, Theorem 27, and Theorem 46. Section 5 summarizes the first part of our paper [5].

Terminology. We refer to the natural numbers, denoted $\mathbb{N}$, as including zero, and denote by $\mathbb{N}^+$ the natural numbers excluding zero. When we speak of the recursive functions, denoted $\mathbb{REC}$, we mean the partial recursive functions. Similarly, the set of Turing machines, denoted TM, includes both halting and non-halting machines. We use the term "domain" of a computational model and of a (partial) function to denote the set of elements over which it operates, not only those for which it is defined. By "image", we mean the values that a function actually takes: $\operatorname{Im} f := \{f(x) \mid x \in \operatorname{dom} f\}$.

Proofs are omitted for conciseness reasons.

2. The Sensitivity of Models to the Domain Interpretation

Computational models. Our research concerns computational models. Obviously, a computational model should perform some computation; however demarcating a clear border between what is a computational model and what is not is problematic. Accordingly, for achieving maximum generality, we do not want to limit computational models to any specific mechanism; hence, we allow a model to be any object, as long as it is associated with a set of functions that it computes.

As models may have non-terminating computations, we deal with sets of partial functions. For convenience, we assume that the domain and range (co-domain) of functions are identical. For simplicity, we mainly deal with deterministic computational models. Most of our definitions and theorems can be directly extended to non-determinism, while the more involved ones are handled in Section 4.1.1.

Definition 1 (Computational Model)

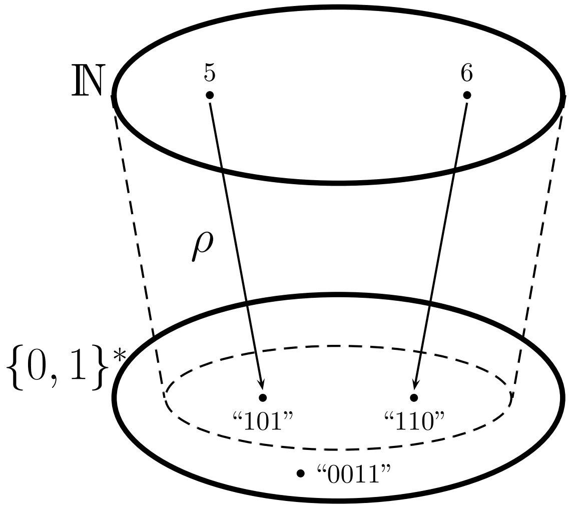
- *A* domain *D is any set of atomic elements.*
- *A* computational model *A over domain D is any object associated with a set of partial functions $f : D^i \to D$, $i \in \mathbb{N}^+$. This set of functions is called the* extensionality *of the computational model, denoted $[\![A]\!]$.*
- *We write* dom *A for the domain over which model A operates.*

In what follows, we often speak of a "submodel" or a "supermodel" of a model, referring to the containment relation between their extensionalities. That is, model A is a *submodel* of B, and B is a *supermodel* of A, if $[\![A]\!] \subseteq [\![B]\!]$. By a "strict submodel" we mean that the containment is proper.

In the following subsections, we explore the sensitivity of models to domain interpretations, ending up with a sufficient and necessary condition for a "harmless representation" (see Figure 2).

2.1. *Injective representations*

Injective representations are the most frequently used ones. The standard decimal and binary notations of the natural numbers are injective (they are not bijective since leading zeros are ignored). Comparisons between computational models are usually done by injective encodings; for example: Church numerals and Gödel numbering are used for comparing the recursive functions and λ-calculus.



$\rho(5) =$"101"

5 is *represented* by "101" via ρ

$[\![\text{"101"}]\!]_\rho = 5$

"101" is *interpreted* as 5 via ρ

"0011" $\notin \operatorname{Im} \rho$

"0011" has no *interpretation* via ρ

Fig. 3. Domain interpretation. Strings are interpreted as natural numbers via the representation ρ.

We begin by defining "representation" to be an injective mapping.

Definition 2 (Representation)

Domain. *Let D_A and D_B be two domains (arbitrary sets of atomic elements). A* representation *of D_A over D_B is an injection $\rho : D_A \to D_B$ (i.e. ρ is total and one-one).*

Function and Relation. *Representations ρ naturally extend to functions and relations f, which are sets of tuples of domain elements: $\rho(f) := \{\langle \rho(x_1), \ldots, \rho(x_n)\rangle \mid \langle x_1, \ldots, x_n\rangle \in f\}$.*

Model. *Representations also naturally extend to (the extensionalities of) computational models, which are sets of functions: $\rho([\![B]\!]) := \{\rho(f) \mid f \in [\![B]\!]\}$.*

An almost dual concept to representation is "interpretation" (see Figures 3 and 4):

Definition 3 (Interpretation) *Assume a representation $\rho : D_A \to D_B$. Then:*

(i) The interpretation of a domain element *$b \in D_B$ via the representation ρ, denoted $[\![b]\!]_\rho$, is the element $\rho^{-1}(b)$ of D_A. If $b \notin \operatorname{Im} \rho$ then its interpretation via ρ is undefined.*

(ii) The interpretation of a function *g over D_B via the representation ρ, denoted $[\![g]\!]_\rho$, is the function $\rho^{-1}(g)$ over D_A, which is $\rho^{-1}\circ g\circ\rho$. If $g\circ\rho(a) \notin \operatorname{Im} \rho$ for some element $a \in D_A$ then $[\![g]\!]_\rho$ is a partial function, where $[\![g]\!]_\rho(a)$ is undefined.*

(iii) The interpretation of a computational model *B via the representation ρ, denoted $[\![B]\!]_\rho$, is the set of functions $\rho^{-1}([\![B]\!])$, which is $\{[\![g]\!]_\rho \mid g \in [\![B]\!]\}$.*

(iv) When considering only total functions, the interpretation of a total computational model *B via the representation ρ, denoted $[\![B]\!]_\rho$, is the set of functions $\{[\![g]\!]_\rho \mid g \in [\![B]\!]$ and $[\![g]\!]_\rho$ is total$\}$.*

"Interpretation via ρ" is the reverse of "representation via ρ", up to the image of ρ. When the representation ρ is bijective we have that "interpretation via ρ" is exactly as "representation via ρ^{-1}."

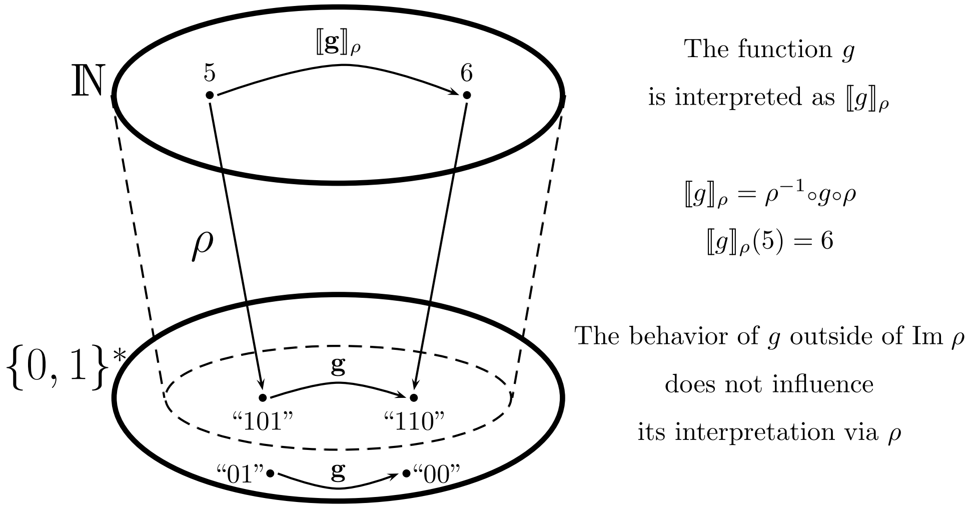
Interpretation and extensionality share the same notation. Indeed, the interpretation of some model B via a representation $\rho : D_A \to D_B$ is its extensionality over the domain D_A, which results from the representation ρ. Note that the extensionality $[\![A]\!]$ of a model A is its interpretation $[\![A]\!]_\iota$ via the identity representation ι.

Sensitivity. Injective representations, however, are prone to hide some computational power. Below is a simple example of such a case.

Example 4 *The set of "even" recursive functions (R_2) can be interpreted as the set of all the recursive functions ($\mathbb{REC}$), by mapping the original natural numbers into the even numbers*

$$R_2 := \left\{ \lambda n. \begin{Bmatrix} 2f(n/2) & n \text{ is even} \\ n & \text{otherwise} \end{Bmatrix} \mid f \in \mathbb{REC} \right\} \qquad\qquad \rho := \lambda n.2n$$

We have that $[\![R_2]\!]_\rho = \mathbb{REC} \supsetneq R_2$.



$[\![\mathbf{g}]\!]_\rho$
$\mathbb{N}$
5
6
ρ
$\{0,1\}^*$
g
"101"
"110"
"01"
g
"00"
The function g
is interpreted as $[\![g]\!]_\rho$
$[\![g]\!]_\rho = \rho^{-1}\circ g \circ \rho$
$[\![g]\!]_\rho(5) = 6$
The behavior of g outside of Im ρ
does not influence
its interpretation via ρ

Fig. 4. Function interpretation. The string functions are interpreted as numeral functions via the representation ρ.

The above anomaly does not appear only with "synthetic" models, but also with some standard ones. An example of such a model is the standard two-counter machines model (see Section 3.3.3).

2.2. *Bijective representations*

The previous subsection demonstrated the sensitivity of models to injective representations. One may ask whether the restriction of representations to bijective mappings might solve the problem. We show that the answer is "no", obtaining that a model might be isomorphic to some of its strict supermodels.

Definition 5 (Isomorphism) *Models A and B (or their extensionalities) are* isomorphic*, denoted $A \cong B$ (or $[\![A]\!] \cong [\![B]\!]$), if there is a bijection π such that $[\![A]\!]_\pi = [\![B]\!]$.*

Theorem 6 ([3]) *There are models isomorphic to a strict supermodel of themselves. That is, there are models A and B, such that $A \cong B$ and $[\![A]\!] \subsetneq [\![B]\!]$.*

A concrete example of such models is given in Example 11 and in [3].

It will be shown, in Section 3, that this process is infinite and symmetric (Theorem 12) – once the model is sensitive to bijective representations, we can always choose a different representation via which we get more, or fewer, functions.

2.3. *Harmless representations*

Are there "harmless representations", via which all models are "protected" from having better and worse interpretations? The answer is "yes", however this family of representations is too limited for being really useful. It is exactly the family of what we call "narrow" permutations:

Definition 7 (Narrow Permutation) *A permutation $\pi : D \to D$ is* narrow *if all its orbits (cycles) are bounded in length by some constant. More precisely, if $\exists k \in \mathbb{N}.\ \forall x \in D.\ |\{\pi^n(x) : n \in \mathbb{N}\}| \leq k$.*

Proposition 8 *A permutation $\pi : D \to D$ is narrow iff there is a positive constant $k \in \mathbb{N}^+$, such that for all $x \in D$ we have $\pi^k(x) = x$. In other words, if $\pi^k = \iota$.*

Theorem 9 ([3]) *For every representation $\rho : D \to D$, there are models A and B such that $[\![A]\!]_\rho = [\![B]\!] \supsetneq [\![A]\!]$, if and only if ρ is not a narrow permutation.*

The family of narrow permutations is very limited and cannot be used as the only mean of interpreting models over different domains. Moreover, this family is not closed under composition. That is, there are narrow permutations π and η such that $\pi \circ \eta$ is not narrow! This situation is very problematic, since interpretations are often used in the context of order relations, for example when saying that two models have the

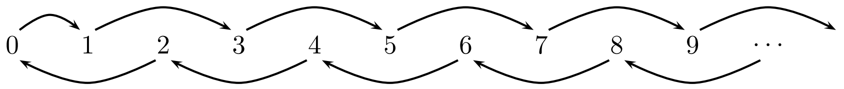


Fig. 5. The permutation π of Example 11.

same extensionality up to the domain interpretation. Any equivalence or order relation should be transitive, which is not the case if representations are limited to narrow permutations.

2.3.1. *Purely harmless representations*

Proceeding in the above direction of seeking a family of representations that would be harmless and closed under functional composition leads us to look for some strict subset of the narrow permutations. However, considering that there are many such (maximal) subsets, which is a reasonable choice?

It turns out that there is a clear distinction between two types of narrow permutations – the "problematic" and the "non-problematic" ones. For every problematic narrow permutation ρ there is a narrow permutation η, such that $\rho\circ\eta$ is not narrow. On the other hand, a non-problematic narrow permutation π guarantees that for every narrow permutation ξ we have that $\xi\circ\pi$ and $\pi\circ\xi$ are narrow.

The family of non-problematic narrow permutations is the "almost identity" permutations (defined bellow), while the rest are problematic.

Definition 10 (Almost Identity) *A permutation $\pi : D \to D$ is* almost identity *if $|\{x \in D \mid \pi(x) \neq x\}| < \infty$.*

The above results suggest that sticking to harmless representations is not a viable direction, as the concept of interpreting models of different domains almost completely evaporates.

2.4. *Models with standard computational properties*

It was shown above that restricting the family of applicable representations cannot solve the sensitivity problem. A different approach is to restrict the definition of a computational model. Nonetheless, in order to allow a variety of internal mechanisms, we seek a restriction on the model's extensionality. We consider four restrictions: (i) closure under functional composition; (ii) inclusion of the identity function; (iii) inclusion of all constant functions; and (iv) the successor function for models operating over $\mathbb{N}$.

In this section we show that the sensitivity problem remains, even when considering only models with all the above properties and allowing only bijective representations.

Closure under functional composition. Denote by $cl(\mathcal{F})$ the set $\mathcal{F}$ of functions closed under functional composition. Considering only models closed under functional composition does not change the sufficient and necessary condition for a harmless representation (Theorem 9), as all models involved in the proof are closed under functional composition.

The identity and constant functions. Adding, or removing, the identity function ι from a model has no influence on its sensitivity to any representation, as $\rho^{-1}\circ(f\circ\iota)\circ\rho = \rho^{-1}\circ(\iota\circ f)\circ\rho = \rho^{-1}\circ f\circ\rho$, for every injection ρ and function f.

Let K be the set of all constant functions over a domain D. Adding, or removing, K from a model A over D, such that $A \cap K \in \{K, \emptyset\}$, has no influence on the sensitivity of A, as $[\![A\circ K]\!]_\rho = [\![K\circ A]\!]_\rho = K$, with respect to total functions, for every injection ρ and model A.

The successor function. It turns out that a model including the successor function and closed under functional composition can still be isomorphic to a strict supermodel of itself:

Example 11 *Define the permutation π over $\mathbb{N}$ (illustrated in Figure 5):*

$$\pi(n) := 1 \textit{ if } n = 0; \quad n + 2 \textit{ if } n \textit{ is odd;} \quad \textit{and} \quad n - 2 \textit{ if } n \textit{ is even}$$

Let s be the successor function over $\mathbb{N}$, and let A be a computational model with the extensionality $[\![A]\!] := \{\pi^i(s) \mid i \in \mathbb{N}\}$. Let B be the computational model obtained from A by closure under functional composition. That is, $[\![B]\!] := cl([\![A]\!])$. It can be shown that B is isomorphic to a strict supermodel of itself.

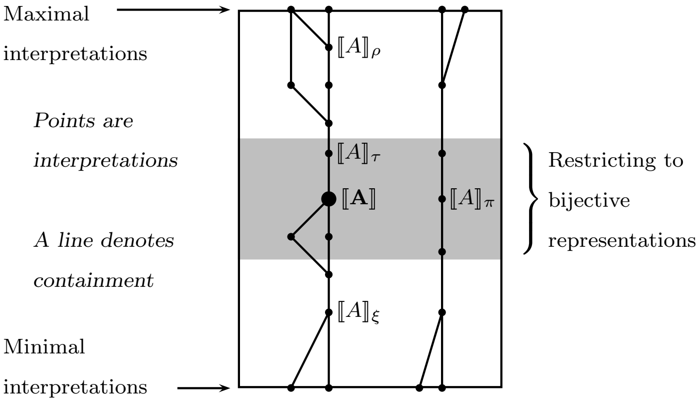


Fig. 6. An illustration of the partially ordered set of interpretations of a model A

3. Organizing Model Interpretations

For examining the influence of the domain interpretation on a model we should compare its different interpretations. Accordingly, we are interested only in the model's interpretations over its original domain. Its interpretations over other domains are isomorphic to those over its original domain, as long as it is not a domain of a lower cardinality. For example, a model has two interpretations with strict containment between them if and only if it has such two interpretations over its original domain.

We are interested in the containment relation between interpretations. That is, examining when some interpretations are better than others in the sense of strictly containing them. Accordingly, the interpretations of a model A form a partially ordered set with respect to containment (illustrated in Figure 6).

Viewing interpretations as a partially ordered set, raises a few natural questions:

- How varied can these partially ordered sets be?
- Are there always maximal interpretations?
- Are there models already in their maximal interpretation (termed "interpretation-complete")?
- How does one choose a proper interpretation?

In what follows we shall shed some light on the subject, considering the above questions and others. In Section 3.1 we answer the first two questions, showing how varied the set of interpretations can be. In Section 3.2 we answer the second pair of questions, dealing with the interpretation-completeness of models.

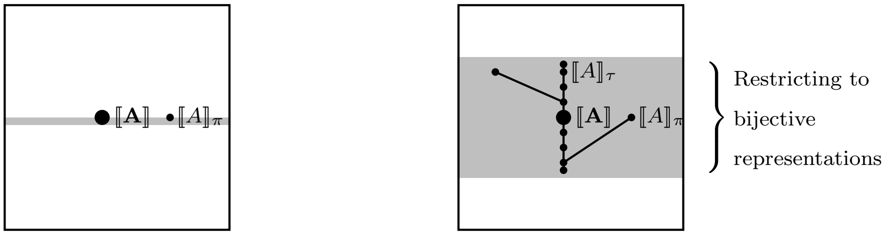
3.1. *The variety of interpretations*

In general, the set of interpretations may be very varied:

- Some interpretations may be better than the original extensionality while others are worse.
- There might be infinitely many interpretations each contained in the next.
- There might be models with no maximal interpretation!
- Non-bijective interpretations might sometimes add to bijective ones, while in other cases only spoil.
- There are models already having their maximal interpretation ("interpretation-complete") or at least so with respect to bijective representations ("interpretation-stable").

A simple example of how different interpretations may enlarge or decrease the original extensionality is given in Example 4. Interpreting the model via the representation $\lambda n.2n$ enlarges the original extensionality, providing all the recursive functions. On the other hand, interpreting the model via the representation $\lambda n.2n+1$ decreases the original extensionality, leaving only the identity function.

We saw, in Section 2.2, that a model can be isomorphic to a strict supermodel of itself. In such a case, there are infinitely many interpretations enlarging the original extensionality, as well as infinitely many decreasing it, while each is contained in the next. Note that this is true for bijective representations, but not necessarily for injective representations.


$\llbracket \mathbf{A} \rrbracket$
$\llbracket A \rrbracket_\pi$
$\llbracket A \rrbracket_\tau$
$\llbracket \mathbf{A} \rrbracket$
$\llbracket A \rrbracket_\pi$
Restricting to bijective representations

Stable model Unstable model

Fig. 7. An illustration of interpretation-stable and unstable models

Theorem 12 *If A is a model and π a bijection such that $\llbracket A \rrbracket \subsetneq \llbracket A \rrbracket_\pi$, then for every $i \in \mathbb{N}$ we have that $\llbracket A \rrbracket_{\pi^i} \subsetneq \llbracket A \rrbracket_{\pi^{i+1}}$ and $\llbracket A \rrbracket_{\pi^{-i}} \supsetneq \llbracket A \rrbracket_{\pi^{-(i+1)}}$.*

Corollary 13 *If A is a model for which there is no bijection π such that $\llbracket A \rrbracket \subsetneq \llbracket A \rrbracket_\pi$, then there is also no bijection η such that $\llbracket A \rrbracket \supsetneq \llbracket A \rrbracket_\eta$.*

We see that once a model has a better interpretation via a bijective representation it cannot have a maximal interpretation via a bijective representation. Nevertheless, it might have a maximal interpretation via an injective representation. There are, however, models with no maximal interpretation at all.

Proposition 14 *There are computational models with no maximal interpretation that extends their original extensionality. That is, there is a computational model A, such that for every representation ρ for which $\llbracket A \rrbracket_\rho \supseteq \llbracket A \rrbracket$ there is a representation η such that $\llbracket A \rrbracket_\eta \supsetneq \llbracket A \rrbracket_\rho$.*

We can show that model A of Example 11 is such a model. We also get that interpretations via non-bijective representations might sometimes only decrease the model's extensionality, while we saw that in other cases they can further enlarge it on top of bijective ones.

3.2. *Interpretation-completeness and interpretation-stability*

We saw that the extensionality of computational models is sensitive to the domain interpretation. There are, however, models that are already in their maximal interpretation (called "interpretation-complete" or in short "complete"), or at least so with respect to bijective representations (called "interpretation-stable" or in short "stable").

Definition 15 *A model A is* interpretation-complete *if there is no representation $\rho : \mathrm{dom}\ A \to \mathrm{dom}\ A$ such that $\llbracket A \rrbracket_\rho \supsetneq \llbracket A \rrbracket$.*

Note that when considering only total functions, the interpretation of a model is also defined to contain only total functions. In such a case a model is considered complete even if some interpretations can extend it with partial functions.

Though we generally consider all injective representations, there are also good justifications to stick to bijective representations, as briefly seen in Section 2, and elaborated on in Section 5 and in [2,4]. Accordingly, we also define completeness with respect to bijective representations, called "interpretation-stability":

Definition 16 *A model A is* interpretation-stable *if there is no bijective representation $\pi : \mathrm{dom}\ A \to \mathrm{dom}\ A$ such that $\llbracket A \rrbracket_\pi \supsetneq \llbracket A \rrbracket$.*

Sticking to bijective representations, there are exactly two options for the representation influence:

- Stable model – totally immune to the influence of bijective representations. No better or worse interpretations are possible via bijective representations.
- Unstable model – there is no maximum, nor minimum, interpretation via bijective representations.

The above is illustrated in Figure 7, and proved in Theorem 12 and Corollary 13.

The general case, allowing non-bijective representations, is much more varied (see Figure 8):

- There are stable models that are incomplete.
- There might be a complete model with worse interpretations.
- Bijective representations preserve completeness.

Completeness obviously implies stability, but the opposite is not true. A simple example is the set of all constant functions except for a single one. A model having this extensionality is stable but incomplete.

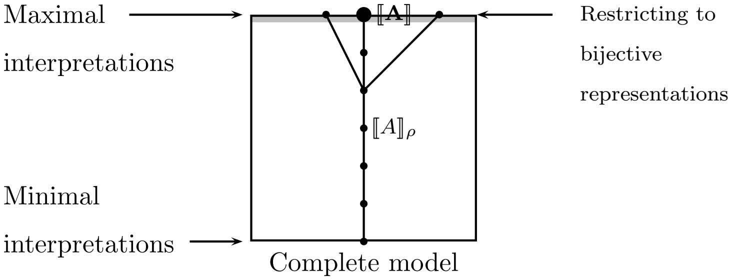


Fig. 8. An illustration of an interpretation-complete model

Completeness assures us that the model cannot have an interpretation better than its original extensionality. However, it might have an interpretation that decreases its original extensionality.

Proposition 17 *An interpretation-complete model might have an interpretation decreasing its extensionality. That is, there is an interpretation-complete model A and a representation ρ such that $[\![A]\!]_\rho \subsetneq [\![A]\!]$.*

Using only bijective representations, we cannot harm the extensionality of a complete model. This follows directly from Corollary 13, as a complete model is also stable.

Corollary 18 *Let A be a model and ρ a representation such that $[\![A]\!]_\rho$ is complete and $[\![A]\!]_\rho \supsetneq [\![A]\!]$. Then there is no bijective representation π such that $[\![A]\!]_\pi = [\![A]\!]_\rho$.*

Corollary 19 *Isomorphism preserves stability and completeness. That is, let A and B be isomorphic models, then A is interpretation-stable iff B is, and A is interpretation-complete iff B is.*

Proposition 20 *A model with a finite extensionality (implementing finitely many functions) is complete.*

Complete models have interesting properties and are generally more convenient to work with. We elaborate on some of the properties concerning power comparison and isomorphism in Section 4.

3.2.1. *Getting maximal interpretations*

A natural question is how to choose a proper representation for getting a maximal interpretation. We should consider two cases, depending on whether the relevant model is complete or not.

For an interpretation-complete model A, we can get a maximal interpretation by one of the following means:

- Its original extensionality.
- Via any bijective representation.
- If A is closed under functional composition: via a representation ρ for which there exists a total injective function $f \in [\![A]\!]$, such that $\mathrm{Im}\, f = \mathrm{Im}\, \rho$ and $f^{-1} \in [\![A]\!]$.

For an incomplete model:

- There is no general known criterion.
- A bijective representation cannot help.
- If one finds a representation via which there is a maximal interpretation, then he can get additional maximal interpretations with the above techniques for complete models.

The special case of proper representations with respect to effectiveness is considered in Section 5.2.

The claims above follow directly from results of previous sections and the following theorem:

Theorem 21 *Let A be a model closed under functional composition, and let $\rho : D \to \mathrm{dom}\, A$ be some representation. Then $[\![A]\!]_\rho$ is isomorphic to $[\![A]\!]$ if there is a total injective function $h \in [\![A]\!]$ such that $\mathrm{Im}\, h = \mathrm{Im}\, \rho$ and $h^{-1}{\restriction_{\mathrm{Im}\, h}} \in [\![A]\!]{\restriction_{\mathrm{Im}\, h}}$.*

3.3. *Specific models*

We turn now to investigate the influence of the domain interpretation on some well known computational models, as well as on hypercomputational models.

3.3.1. *The recursive functions*

The recursive functions (both total and partial) are interpretation-complete! Their completeness is of special importance due to their rôle in the notion of effectiveness. This is elaborated on in Section 5. The completeness is of both the unary recursive functions and of the functions of any arity.

In Section 3.3.2, it will be shown that the recursive functions are isomorphic to the functions computed by Turing machines. They are also isomorphic to 3-counter machines, while being a maximal interpretation of the incomplete 2-counter machines model.

Theorem 22 ([3]) *The unary recursive functions are interpretation-complete.*

Theorem 23 *The partial recursive functions and the total recursive functions are interpretation-complete.*

3.3.2. *Turing machines*

Turing machines are interpretation-complete. As with the recursive functions, this completeness is of special importance due to the rôle of Turing machines in the notion of effectiveness (see Section 5).

Theorem 24 ([3]) *Turing machines are interpretation-complete.*

When seeking a maximal interpretation for Turing machines or for the recursive functions, the criteria of Section 3.2.1 may be extended:

Theorem 25 *An interpretation* $[\![\mathrm{TM}]\!]_\rho$ *of Turing machines via some injection* ρ *is maximal if* $|\mathrm{Im}\ \rho|$ *is infinite and there is a function* $h \in [\![\mathrm{TM}]\!]$ *such that* $\mathrm{Im}\ h = \mathrm{Im}\ \rho$.

Note that the function h above needs not be total nor injective.

3.3.3. *Counter machines*

The model of two counter machines is very interesting. It was shown independently by Janis Barzdins [1], Rich Schroeppel [19], and Frances Yao that two-counter machines cannot compute the function $\lambda x.2^x$. On the other hand, since two-counter machines can compute all the recursive functions via an injective representation (viz. $n \mapsto 2^n$; see, for example, [12]), it follows that two-counter machines are interpretation-incomplete.

It turns out that the models of one-counter machines as well as of three-or-more counter machines are interpretation-complete.

3.3.4. *Hypercomputational models*

A computational model is generally said to be "hypercomputational" if it computes more than Turing machines or more than the recursive functions (see Definition 47 in Section 5). Due to the completeness of Turing machines and the recursive functions, such a model may indeed be regarded as more powerful. Power comparison is treated in detail in Section 4, and the issue of effective computation over arbitrary domains is treated in detail in Section 5.

Can we conclude from the interpretation-completeness of the recursive functions that every hypercomputational model is interpretation-complete? The answer is, in general, "no". However, if the hypercomputational model preserves the basic closure properties of the recursive functions, then the answer is "yes".

The following example is an incomplete hypercomputational model:

Example 26 *Let h be the (incomputable) halting function. Define:*

$$h_i := \lambda n. \begin{cases} 2^i h(n/2^i) & 2^i \textit{ divides } n \\ 0 & \textit{otherwise} \end{cases} \qquad\qquad \rho := \lambda n.2n$$

Let A be a computational model with the extensionality $[\![A]\!] := \mathbb{REC} \cup \{h_i \mid i \in \mathbb{N}^+\}$. That is, $[\![A]\!]$ includes all the recursive functions and all functions h_i for $i \geq 1$. We have that $[\![A]\!]_\rho = \mathbb{REC} \cup \{h_i \mid i \in \mathbb{N}\} \supsetneq [\![A]\!]$.

Yet, the completeness proof of the recursive functions (Theorem 23) may be extended to hypercomputational models, as long as they have the relevant closure properties. It also means that their domain is denumerable, as with higher cardinalities there is no meaning to primitive recursion or minimalization:

Theorem 27 *Let A be a computational model over $\mathbb{N}$ computing all the recursive function and closed under functional composition, primitive recursion and minimalization. Then A is interpretation-complete.*

A special case of such an interpretation-complete hypercomputational model is an oracle Turing machine.

Corollary 28 *An oracle Turing machine is interpretation-complete.*

4. Comparing Computational Power

It is standard in the literature to compare the power of computational models. However, neglecting the possibility of interpretation-incomplete models, it is common to say that model B is strictly stronger than model A when it computes strictly more functions. This might allow one to show that some computational models are *strictly* stronger than themselves (see Section 2).

We define a comparison notion over arbitrary domains based on model interpretations. With this notion, model B is strictly stronger than A if B has an interpretation that contains A, whereas A cannot contain B under *any* interpretation.

We start, in Section 4.1, by providing the mathematical definition of the comparison notion. We give a basic definition, allowing all interpretations, and two additional definitions with some restrictions on the allowed interpretations. In Section 4.1.1 we extend the notion no non-deterministic models.

In Section 4.2 we provide several results with respect to power comparison, isomorphism and completeness. In Section 4.3, we compare between some standard models, as Turing machines, stack machines, etc..

4.1. *The comparison notions*

Since we are only interested here in the extensional quality of a computational model (the set of functions or relations that it computes), not complexity-based comparison or step-by-step simulation, we use model interpretations as the basis for comparison.

We generally say that model B is at least as powerful as model A if it can compute whatever A can. When both models operate over the same domain it simply means containment: B is at least as powerful as A if $\llbracket B \rrbracket \supseteq \llbracket A \rrbracket$. However, when the models operate over different domains we ought to interpret one model over the domain of the other. Hence, the general comparison notion would say that B is at least as powerful as A if it has an interpretation that contains A.

As one textbook states [21, p. 30]:

> Computability relative to a coding is the basic concept in comparing the power of computation models.... The computational power of the model is represented by the extension of the set of all functions computable according to the model. Thus, we can compare the power of computation models using the concept 'incorporation relative to some suitable coding'.

We provide three notions of comparison, depending on the allowed representations. The basic, most permissive, comparison notion allows any injection, while the firmest notion allows only bijections. In between, we define a notion that allows injections for which the "as-powerful" model can fix their image.

Definition 29 (Power Comparison)

- *Model B is* at least as powerful *as model A, denoted $B \gtrsim A$, if it has an interpretation that contains the extensionality of A. That is, $B \gtrsim A$ iff exists an injection $\rho : \mathrm{dom}\, A \rightarrow \mathrm{dom}\, B$, such that $\llbracket B \rrbracket_\rho \supseteq \llbracket A \rrbracket$.*
- *Model B is* decently at least as powerful *as model A, denoted $B \succsim A$, if it has an interpretation that contains the extensionality of A via a representation for which it can fix its image. That is, $B \succsim A$ iff exists an injection $\rho : \mathrm{dom}\, A \rightarrow \mathrm{dom}\, B$, such that $\llbracket B \rrbracket_\rho \supseteq \llbracket A \rrbracket$ and there is a total function $g \in \llbracket B \rrbracket$ for which $\mathrm{Im}\, g = \mathrm{Im}\, \rho$ and such that for every $y \in \mathrm{Im}\, \rho$ we have that $g(y) = y$.*
- *Model B is* bijectively at least as powerful *as model A, denoted $B \gtrapprox A$, if it has a bijective interpretation that contains the extensionality of A. That is, $B \gtrapprox A$ iff exists a bijection $\pi : \mathrm{dom}\, A \rightarrow \mathrm{dom}\, B$, such that $\llbracket B \rrbracket_\pi \supseteq \llbracket A \rrbracket$.*

Proposition 30 *The computational power relations $\gtrsim, \succsim, \gtrapprox$ are quasi-orders.*

Obviously, the two latter comparison notions imply the former one. For the third notion to imply the second one, it is sufficient that the as-powerful model has some surjective function, even the identity function. Additionally, when assuming a little more about the representation and the model, the second notion also implies the third one (see Theorem 21).

Note 31 *When comparing the computational power of models, it should be noted that one function cannot be "better" than another; only a model can be better than another. There is only an equivalence relation between functions, where two partial functions, f and g, over the same domain D are deemed* equal, *denoted $f = g$, if they are defined for exactly the same elements of the domain and have the same value whenever they are defined. For example, a total function f is not better than nor equal to a partial function g that has the same values as f when it converges. This is clearly demonstrated by taking g to be a 'halting function', which returns* 1 *when the input encodes a halting machine and diverges otherwise.*

Our comparison notions go along with some standard approaches, for instance in [7, p. 24], [18, p. 27] and [21, p. 30]. Our decent-comparison notion follows Rabin's definition of a computable group [16, p. 343]:" DEFINITION 3. An *indexing* of a set S is a one to one mapping $i : S \to I$ such that $i(S)$ is a recursive subset of I."

Example 32 *Turing machines are at least as powerful as the recursive functions via a unary representation of the natural numbers. See, for example, [9, p. 147]. Indeed, it is so by all three notions (see Theorem 45).*

One may wonder why we do not require the representation to be Turing-computable. A detailed answer to that is given in Section 5. The main points are:

- When comparing models in the scope of defining effectiveness, the requirement of a Turing-computable representation leads to a circular definition.
- One may wish to compare sub-recursive models or hypercomputational models, for which Turing-computability is not necessarily the proper constraint.

Power equivalence. The power equivalence relation between models follows directly from the power comparison notion. Models A and B are of equivalent power if $A \gtrsim B$ and $B \gtrsim A$.

Definition 33 (Power Equivalence)

- *Models A and B are* power equivalent, *denoted $A \sim B$, if $A \gtrsim B$ and $B \gtrsim A$.*
- *Models A and B are* decently power equivalent, *denoted $A \simeq B$, if $A \succsim B$ and $B \succsim A$.*
- *Models A and B are* bijectively power equivalent, *denoted $A \approx B$, if $A \gtrapprox B$ and $B \gtrapprox A$.*

Example 34 *The (untyped) λ-calculus is power equivalent to the recursive functions, via Church numerals, on the one hand, and via Gödelization, on the other. However, it is not interpretation-complete, as it cannot even compute the identity function over an arbitrary lambda-term. Hence, it is not bijectively-power equivalent to the recursive functions (otherwise contradicting the completeness of the recursive functions).*

Strictly stronger. We generally think of model B as stronger than model A if it can compute more. However, because of the sensitivity to the domain interpretation (Section 2), proper containment does not imply more computational power. That is, $[\![B]\!] \supsetneq [\![A]\!] \not\Rightarrow B \gnsim A$. This is so for all three comparison notions.

The proper definition of model B being strictly stronger than model A is that $B \gtrsim A$ while $A \not\gtrsim B$.

Definition 35 (Stronger)

- *Model B is* stronger *than model A, denoted $B \gnsim A$, if $B \gtrsim A$ while $A \not\gtrsim B$.*
- *Model B is* decently stronger *than model A, denoted $B \succnsim A$, if $B \succsim A$ while $A \not\succsim B$.*
- *Model B is* bijectively stronger *than model A, denoted $B \gnapprox A$, if $B \gtrapprox A$ while $A \not\gtrapprox B$.*

Note that for model B to be stronger than model A there should be no injection ρ via which $A \gtrsim_\rho B$. In contradistinction to the "as powerful" case, some model B may be bijectively stronger than a model A, but not stronger than A. However, if A is interpretation-complete, we do have that $B \gnapprox A$ implies $B \gnsim A$ (Theorem 42).

Example 36 *Real recursive functions [14], operating over $\mathbb{R}^2$, are decently stronger than Turing machines. The comparison is done via an injective representation $\psi : \{0,1\}^* \to \mathbb{R}^2$, defined by $[\![x]\!]_\psi := (0, [\![x]\!]_\rho)$, where $[\![x]\!]_\rho$ is the standard binary interpretation over the natural numbers [14, p. 849]. Since the model computes the floor function ($\lambda x. \lfloor x \rfloor$) [14, p. 843], it follows that it also has a function which fixes the representation image. On the other hand, the real recursive functions are obviously not bijectively stronger than the recursive functions, as their domain is of a higher cardinality.*

Universal machines. Computation is often performed via universal machines, also referred to as "interpreters". That is, a single machine (or computer program) gets as input both the machine to interpret and the latter's input. One may wonder how to compare the computational power of universal machines. From our point of view, computational power is extensionality, meaning the set of computed functions. Hence, when comparing universal machines we compare the sets of functions that they interpret. Accordingly, it is exactly like comparing computational models.

4.1.1. *Non-deterministic models*

The extension of most of the definitions and theorems given so far (in this section and the previous ones) to non-deterministic models is quite straightforward. There are, however, some specific issues concerning the power comparison of non-deterministic models, which will be discussed below.

We begin by defining what we mean by a non-deterministic computational model and its extensionality:

Definition 37 (Non-deterministic Computational Model) *A non-deterministic computational model A over domain D is any object associated with a set of (non-unary) relations over $D \cup \bot$. This set of relations is called the* extensionality *of the non-deterministic computational model, denoted $\llbracket A \rrbracket$.*

Note that we use the special symbol $\bot$ in the above definition for denoting a non-halting computation, while we did not need it in the definition of a deterministic model (Definition 1). The reason is that a non-deterministic computation might sometimes diverge on a domain element e and sometimes converge to some value v. In such a case, the value of the computation will be both $\bot$ and v, denoted by $\langle e, \bot \rangle, \langle e, v \rangle$ in the (multivalued) function's description. On the other hand, the divergence of a deterministic function on a domain element e, may be simply denoted by not having a tuple with e in the function's description.

As a result of using $\bot$ to denote divergence, we may assume that all functions have at least one value for every domain element.

When investigating the influence of the domain interpretation, we are concerned with the containment relation between the different interpretations of a computational model. Hence, the definitions given in this section and the previous ones apply to both deterministic and non-deterministic models, as in both types of models the interpretations are sets of relations.

When we compare the power of computational models, two functions are considered equal if and only if they are described by the same relation, while a model is as powerful as another if it contains all the functions of the former (see Note 31). This is also the case when we compare between non-deterministic models. There might be cases in which a different approach is required, assuming a different equality notion between functions. For example, two non-deterministic functions might be considered equal if they *may* always produce the same value. That is, a function that sometimes diverges and sometimes converges with the value v is considered equal to a function that always converges with the value v. In such cases, the comparison notions should be adjusted to take into account the special equality notion between functions.

By this comparison approach, a non-deterministic model B may be as powerful as a deterministic model A (if it deterministically computes all the functions of A, in addition to its non-deterministic computations), while the opposite is impossible (when B actually has some non-deterministic computations).

4.2. *Power comparison, isomorphism, and interpretation-completeness*

The general approach for showing that model B is stronger than model A is tedious – we should negate the possibility that $A \gtrsim_\rho B$ for all injections ρ. The situation is much simpler with interpretation-complete models, for which proper containment does imply more power:

Theorem 38 ([3]) *For an interpretation-complete model A and some model B, we have that $B \gnsim A$ iff there exists an injection ρ such that $\llbracket B \rrbracket_\rho \supsetneq \llbracket A \rrbracket$.*

Isomorphism and bijective power equivalence are very similar notions, though not the same. However, when sticking to interpretation-stable models they do coincide:

Theorem 39

(i) Isomorphism implies bijective power equivalence.

(ii) Bijective power equivalence does not imply isomorphism.

(iii) For stable models, bijective power equivalence implies isomorphism. That is, if model A is stable, then for every model B, B is isomorphic to A if and only if it is bijectively power equivalent to A.

Isomorphism preserves stability and completeness (Corollary 19). This is also the case, by the above theorem, with bijective power equivalence.

Corollary 40 *Bijective power equivalence preserves stability and completeness. That is, let A and B be bijectively power equivalent models, then A is stable iff B is, and A is complete iff B is.*

The formulation of the previous theorem can be strengthened for complete models:

Lemma 41 ([3]) *If model A is complete and $A \gtrsim B \gtrapprox A$ for some model B, then A and B are isomorphic.*

Interpretation-completeness also helps with showing that some model is stronger than another.

Theorem 42 *If model A is complete, then $B \gnapprox A$ implies $B \gnsim A$*

Theorem 43 ([3]) *If model A is complete, $A \approx B$ and $[\![B]\!] \subsetneq [\![C]\!]$, for models B and C, then $C \gnsim A$.*

4.3. *Comparison of some Standard Models*

As discussed in the beginning of this section, the proper containment between the recursive functions and the primitive recursive functions does not imply that the former are strictly more powerful than the latter. Nonetheless, the recursive functions are indeed strictly more powerful, even by the general comparison notion, allowing all possible interpretations of the primitive recursive functions.

Theorem 44 ([3]) *The primitive recursive functions are strictly weaker than the recursive functions.*

We do not know if the primitive recursive functions are interpretation-complete, Theorem 44 notwithstanding.

It is common to show that Turing machines and the recursive functions are of equivalent computational power. Actually, they are isomorphic. We base the proof on known results, given in [10].

Theorem 45 ([3]) *Turing machines, over a binary alphabet, and the recursive functions are isomorphic.*

From [10,3] we also have that random access machines and counter machines with unlimited number of counters have exactly the same extensionality as the recursive functions.

This is not the case with two-counter machines. They are of equivalent power to the recursive functions, however not isomorphic to them. Indeed, they are bijectively-weaker than them (otherwise contradicting the stability of the recursive functions).

By known results, two-stack machines have the same extensionality as Turing machines.

Due to the closure properties of Turing machines and the recursive functions, they are bijectively at least as powerful as some model A if and only if they are decently at least as powerful as A.

Theorem 46 *Let A be a computational model operating over a denumerable domain. Then Turing machines and the recursive functions are decently at least as powerful as A if and only if they are bijectively at least as powerful as A. That is, $\mathrm{TM} \succsim A$ iff $\mathrm{TM} \gtrapprox A$ iff $\mathbb{REC} \succsim A$ iff $\mathbb{REC} \gtrapprox A$.*

5. Effective Computation

In 1936, Alonzo Church and Alan Turing each formulated a claim that a particular model of computation completely captures the conceptual notion of “effective” computability. Church [6, p. 356] proposed that effective computability of numeric functions be identified with Gödel and Herbrand’s general recursive functions, or – equivalently, as it turned out [6] – with Church and Kleene’s lambda-definable functions of positive integers.

Turing, on the other hand, explicitly extends the notion of “effective” beyond the natural numbers [23, fn. p. 166] (emphasis added):

> We shall use the expression “computable function” to mean a function calculable by a machine, and we let “effectively calculable” refer to the intuitive idea without particular identification with one of these definitions. *We do not restrict the values taken by a computable function to be natural numbers*; we may for instance have computable propositional functions.

Our purpose, in Section 5.1, is to formalize and analyze the Church-Turing Thesis, referring to functions over arbitrary domains.

Equipped with this definition, and due to the interpretation-completeness of Turing machines, we define, in Section 5.2, effective representations of constructible domains.

5.1. *The Church-Turing Thesis over arbitrary domains*

Simply put, the Church-Turing Thesis is not well defined for arbitrary domains: the choice of domain interpretation might have a significant influence on the outcome. We explore below the importance of the domain interpretation and suggest how to overcome this problem.

Computational model versus single function. A single function over an arbitrary domain cannot be classified as computable or not. Its computability depends on the representation of the domain.[2] For example, the (uncomputable) halting function over the natural numbers (sans the standard order) is isomorphic to the simple parity function, under a permutation of the natural numbers that maps the usual codes of halting Turing machines to strings ending in "0", and the rest of the numbers to strings ending with "1". The result is a computable standalone "halting" function.

An analysis of the classes of number-theoretic functions that are computable relative to different notations (representations) is provided by Shapiro [20, p. 15]:

> It is shown, in particular, that the class of number-theoretic functions which are computable relative to every notation is too narrow, containing only rather trivial functions, and that the class of number-theoretic functions which are computable relative to some notation is too broad containing, for example, every characteristic function.

An intuitive approach is to restrict the representation only to "natural" mappings between the domains. However, when doing so in the scope of defining "effectiveness", one must use a vague and undefined notion.

This problem was already pointed out by Richard Montague on 1960 [13, pp. 430–431]:

> Now Turing's notion of computability applies directly only to functions on and to the set of natural numbers. Even its extension to functions defined on (and with values in) another denumerable set S cannot be accomplished in a completely unobjectionable way. One would be inclined to choose a one-to-one correspondence between S and the set of natural numbers, and to call a function f on S computable if the function of natural numbers induced by f under this correspondence is computable in Turing's sense. But the notion so obtained depends on what correspondence between S and the set of natural numbers is chosen; the sets of computable functions on S correlated with two such correspondences will in general differ. The natural procedure is to restrict consideration to those correspondences which are in some sense 'effective', and hence to characterize a computable function on S as a function f such that, for some effective correspondence between S and the set of natural numbers, the function induced by f under this correspondence is computable in Turing's sense. But the notion of effectiveness remains to be analyzed, and would indeed seem to coincide with computability.

Stewart Shapiro suggests a definition of "acceptable notation", based on intuitive concepts [20, p. 18]:

> This suggests two informal criteria on notations employed by algorithms:
>
> (1) The computist should be able to *write* numbers in the notation. If he has a particular number in mind, he should (in principle) be able to write and identify tokens for the corresponding numeral.
>
> (2) The computist should be able to *read* the notation. If he is given a token for a numeral, he should (in principle) be able to determine what number it denotes.
>
> It is admitted that these conditions are, at best, vague and perhaps obscure.

Michael Rescorla argues that the circularity is inherent in the Church-Turing Thesis [17].

A possible solution is to allow any representation (injection between domains), while checking for the effectiveness of an entire computational model. That is, to check for the computability of a function together with the other functions that are computable by that computational model. The purpose lying behind this idea is to view the domain elements as arbitrary objects, deriving all their meaning from the model's

[2] There are functions that are inherently uncomputable, via all domain representations. For example, a permutation of some countable domain, in which the lengths of the orbits are exactly the standard encodings of the non-halting Turing machines.

functions. For example, it is obvious that the halting function has a meaning only if one knows the order of the elements of its domain. In that case, the successor function provides the meaning for the elements.

Two variants of this solution, corresponding to the variants of our comparison notion (Definition 29), are to either allow only bijective representations, or else allow injections provided that their images are computable.

Adopting the above approach of checking for computability of an entire computational model, we interpret the Church-Turing Thesis as follows:

Thesis A. *All "effective" computational models are of equivalent power to, or weaker than, Turing machines.*

By "effective", in quotes, we mean effective in its intuitive sense.

By a "computational model" we refer to any object that is associated with a set of partial functions (Definition 1). By "equivalent to, or weaker than" we refer to the comparison notions of Definition 29.

A strict supermodel of the recursive functions (or Turing machines) is a "hypercomputational" model.

Definition 47 (Hypercomputational Model) *A computational model H is* hypercomputational *if it is stronger than Turing machines. That is, if $H \mathrel{\underset{\not\approx}{>}} \mathrm{TM}$. (The corresponding variations, using bijective power comparison or decent power comparison, are $H \mathrel{\underset{\not\approxeq}{\geq}} \mathrm{TM}$ or $H \mathrel{\underset{\not\approx}{\succ}} \mathrm{TM}$.)*

Our interpretation of the Church-Turing Thesis (Thesis A) agrees with Rabin's definition of a computable group [16, p. 343], as well as with its generalization, by Lambert [24, p. 594], to any algebraic structure. Similar notions were also presented by Froehlich and Shepherdson [8] and Mal'cev [11].

Influence of representations. The Church-Turing Thesis, as stated above (Thesis A), matches the intuitive understanding only due to the interpretation-completeness of Turing machines (Theorem 24). Were the thesis defined in terms of two-counter machines (2CM), for example, it would make no sense: a computational model is not necessarily stronger than 2CM even if it computes strictly more functions.

5.2. *Effective representations*

What is an effective representation? We argued above that a "natural representation" must be a vague notion when used in the context of defining effectiveness. We avoided the need of restricting the representation by checking the effectiveness of entire computational models. But what if we adopt the Church-Turing Thesis; can we then define what is an effective string representation?

Simply put, there is a problem here. Turing machines operate only over strings. Thus a string representation, which is an injection from some domain D to Σ^*, is not itself computable by a Turing machine. All the same, when we consider, for example, string representations of natural numbers, we can obviously say regarding some of them that they are effective. How is that possible? The point is that we look at the natural numbers as having some structure, usually assuming their standard order. A function over the natural numbers without their order is not really well-defined. As we saw, the halting function and the simple parity function are exactly the same (isomorphic) function when numbers are unordered.

Hence, even when adopting the Church-Turing Thesis, a domain without any structure cannot have an effective representation. It is just a set of arbitrary elements. However, if the domain comes with a generating mechanism (as the natural numbers come with the successor) we can consider effective representations.

Due to the interpretation-completeness of the recursive functions and Turing machines, we can define what is an effective string representation of the natural numbers (with their standard structure). A similar definition can be given for other domains, provided that they come with some finite means of generating them all, akin to successor for the naturals.

Definition 48 *An* effective representation *of the natural numbers by strings is an injection $\rho : \mathbb{N} \to \Sigma^*$, such that $\rho(s)$ is Turing-computable ($\rho(s) \in [\![\mathrm{TM}]\!]$), where s is the successor function over $\mathbb{N}$.*

That is, a representation of the natural numbers is effective if the successor function is Turing-computable via this representation.

Note 49 *One may also require that the image of the representation ρ is Turing computable, along the decent power comparison notion. In such a case, there would also be a corresponding bijective representation (see Theorem 21 and 25).*

We justify the above definition of an effective representation by showing that: (a) every recursive function is Turing-computable via any effective representation; (b) every non-recursive function is not Turing-computable via any effective representation; and (c) for every non-effective representation there is a recursive function that is not Turing-computable via it.

Theorem 50 ([5])

(a) Let f be a recursive function and $\rho : \mathbb{N} \to \Sigma^*$ an effective representation. Then $\rho(f) \in [\![\mathrm{TM}]\!]$.

(b) Let g be a non-recursive function and $\rho : \mathbb{N} \to \Sigma^*$ an effective representation. Then $\rho(g) \notin [\![\mathrm{TM}]\!]$.

(c) Let $\eta : \mathbb{N} \to \Sigma^*$ be a non-effective representation. Then there is a recursive function f, such that $\eta(f) \notin [\![\mathrm{TM}]\!]$.

To see the importance of the interpretation-completeness for the definition of an effective representation, one can check that an analogous definition cannot be provided with two-counter machines as the yardstick.

Our definition of an effective representation resembles Shapiro's notion of an "acceptable notation" [20, p. 19] and goes along with Weihrauch's justifications for the effectiveness of the "standard numberings" (representations by natural numbers) [25, p. 80–81]. A description of the correlation between our notion and Shapiro's and Weihrauch's notions can be found in [5].

6. Discussion

Key points. In this paper, we have explored various aspects of the influence of domain interpretations on the extensionality of computational models, and suggested how we believe they should be handled. Some of the key points are:

- The extensionality of computational models is a very interesting set (of functions) – it varies from "fluid" sets for which containment does not mean more power, as with ordinary sets, to explicit, complete, sets for which nothing can be added without additional power.
- Get to know your model: Is it interpretation-stable or interpretation-complete?
- Compare computational power properly.
- Turing machines and the recursive functions are shown to be robust, this time from the representation/interpretation point of view.
- Effectiveness does not apply to a single function; it is a set of functions, or a function together with its domain constructors, that may be deemed effective.

Domain Representation. One might think that our study is a part of the "domain representation" research area. This is not the case. Indeed, the basic concept of "representation" is the same: one set of elements is represented by a subset of another set. Nevertheless, the issues studied are very different. *Domain representation* concerns mapping of some (possibly topological/metric) sets into *domains* (partially ordered sets with some properties), and studies the (topological/metric) properties preserved via the mappings.

Further research. A central issue, yet to be understood, is the relation between the internal mechanism of a computational model and its extensional properties of interpretation-completeness and interpretation-stability. With the recursive functions, this relation is apparent, shown in the proof of their interpretation-completeness. However, we do not know how to relate, in general, the completeness or stability of a computational model to its internal mechanism. A more specific open question is whether there is some standard, well known, computational model that is unstable.

References

[1] J. Barzdins. About one class of Turing machines (Minsky machines). *Algebra and Logic (seminar)*, 1(6):42–51, 1962.

[2] U. Boker and N. Dershowitz. How to compare the power of computational models. In *Computability in Europe 2005: New Computational Paradigms (Amsterdam), S. Barry Cooper, Benedikt Löwe, Leen Torenvliet, eds.*, volume 3526 of *Lecture Notes in Computer Science*, pages 54–64, Berlin, Germany, 2005. Springer-Verlag.

[3] U. Boker and N. Dershowitz. Comparing computational power. *Logic Journal of the IGPL*, 14(5):633–648, 2006.

[4] U. Boker and N. Dershowitz. A hypercomputational alien. *Applied Mathematics and Computation*, 178(1):44–57, 2006.

[5] U. Boker and N. Dershowitz. The Church-Turing thesis over arbitrary domains. In A. Avron, N. Dershowitz, and A. Rabinovich, editors, *Pillars of Computer Science*, volume 4800 of *Lecture Notes in Computer Science*, pages 199–229. Springer, 2008.

[6] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58:345–363, 1936.

[7] N. Cutland. *Computability: An Introduction to Recursive Function Theory.* Cambridge University Press, Cambridge, 1980.

[8] A. Froehlich and J. Shepherdson. Effective procedures in field theory. *Philosophical Transactions of the Royal Society of London*, 248:14–20, 1956.

[9] F. Hennie. *Introduction to Computability.* Addison-Wesley, Reading, MA, 1977.

[10] N. D. Jones. *Computability and Complexity from a Programming Perspective.* The MIT Press, Cambridge, Massachusetts, 1997.

[11] A. Mal'cev. Constructive algebras I. *Russian Mathematical Surveys*, 16:77–129, 1961.

[12] M. L. Minsky. *Computation: Finite and Infinite Machines.* Prentice-Hall, Englewood Cliffs, N.J., 1967.

[13] R. Montague. Towards a general theory of computability. *Synthese*, 12(4):429–438, 1960.

[14] J. Mycka and J. F. Costa. Real recursive functions and their hierarchy. *Journal of Complexity*, 20(6):835–857, 2004.

[15] J. Myhill. Some philosophical implications of mathematical logic. Three classes of ideas. *The Review of Metaphysics*, 6(2):165–198, 1952.

[16] M. O. Rabin. Computable algebra, general theory and theory of computable fields. *Transactions of the American Mathematical Society*, 95(2):341–360, 1960.

[17] M. Rescorla. Church's thesis and the conceptual analysis of computability. *Notre Dame Journal of Formal Logic*, 48(2):253–280, 2007.

[18] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, New York, 1966.

[19] R. Schroeppel. A two counter machine cannot calculate 2^N. Technical report, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1972. available at: ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-257.pdf.

[20] S. Shapiro. Acceptable notation. *Notre Dame Journal of Formal Logic*, 23(1):14–20, 1982.

[21] R. Sommerhalder and S. C. van Westrhenen. *The Theory of Computability: Programs, Machines, Effectiveness and Feasibility.* Addison-Wesley, Workingham, England, 1988.

[22] G. J. Tourlakis. *Computability.* Reston Publishing Company, Reston, VA, 1984.

[23] A. M. Turing. Systems of logic based on ordinals. *Proceedings of the London Mathematical Society*, 45:161–228, 1939.

[24] J. W. M. Lambert. A notion of effectiveness in arbitrary structures. *The Journal of Symbolic Logic*, 33(4):577–602, 1968.

[25] K. Weihrauch. *Computability*, volume 9 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin, 1987.

[26] K. Weihrauch. *Computable Analysis — An introduction.* Springer-Verlag, Berlin, 2000.

On the Convergence of a Population Protocol When Population Goes to Infinity

Olivier Bournez [a] Philippe Chassaing [b] Johanne Cohen [a]
Lucas Gerin [b] Xavier Koegler [c]

[a]LORIA/INRIA-CNRS, *615 Rue du Jardin Botanique, 54602*
Villers-Lès-Nancy, FRANCE

[b]IECN/UHP, *BP 239, 54506 Vandoeuvre-Lès-Nancy Cedex, FRANCE*

[c] Ecole Normale Supérieure, *45, rue d'Ulm, 75230 Paris cedex 05, FRANCE*

Abstract

Population protocols have been introduced as a model of sensor networks consisting of very limited mobile agents with no control over their own movement. A population protocol corresponds to a collection of anonymous agents, modeled by finite automata, that interact with one another to carry out computations, by updating their states, using some rules.

Their computational power has been investigated under several hypotheses but always when restricted to finite size populations. In particular, predicates stably computable in the original model have been characterized as those definable in Presburger arithmetic.

In this paper, we study mathematically a particular population protocol that we show to compute in some natural sense some algebraic irrational number, whenever the population goes to infinity. Hence we show that these protocols seem to have a rather different computational power when considered as computing functions, and when a huge population hypothesis is considered.

1 Motivation

The computational power of networks of finitely many anonymous resource-limited mobile agents has been investigated in several recent papers. In partic-

Email addresses: bournez@loria.fr (Olivier Bournez), chassain@iecn.u-nancy.fr (Philippe Chassaing), jcohen@loria.fr (Johanne Cohen), Lucas.Gerin@iecn.u-nancy.fr (Lucas Gerin), koegler@clipper.ens.fr (Xavier Koegler).

ular, the population protocol model, introduced in [1], consists of a population of finite-state agents that interact in pairs, where each interaction updates the state of both participants according to a transition based on the previous states of the participants. When all agents converge after some finite time to a common value, this value represents the result of the computation.

Several variants of the original model have been considered but with common features. Following survey [3]: anonymous finite-state agents (the system consists of a large population of indistinguishable finite-state agents), computation by direct interaction (an interaction between two agents updates their states according to a joint transition table), unpredictable interaction patterns (the choice of interactions is made by an adversary, possibly limited to pairing only agents in an interaction graph), distributed input and outputs (the input to a population protocol is distributed across the initial state of the entire population, similarly the output is distributed to all agents), convergence rather than termination (the agents' output are required to converge after some time to a common correct value).

Typically, in the spirit of [1] and following papers (see again [3] for a survey), population protocols are assumed to (stably) compute predicates: a population protocols stably computes a predicate ϕ, if for any possible input x of ϕ, whenever $\phi(x)$ is true all agents of the population eventually stabilize to a state corresponding to 1, and whenever $\phi(x)$ is false, all agents of the population eventually stabilize to a state corresponding to 0.

Predicates stably computable by population protocols in this sense have been characterized as being precisely the semi-linear predicates, that is to say those predicates on counts of input agents definable in first-order Presburger arithmetic [9]. Semilinearity was shown to be sufficient in [1] and necessary in [2].

In this paper, we want to study a new variant: we assume a population close to infinity (we call this *a huge population hypothesis*), and we don't want to focus on protocols as predicate recognizers, but as computing functions. We assume outputs to correspond to proportions, which are clearly the analog of counts whenever the population is infinite or close to infinity.

Indeed, we consider a particular population protocol, that we prove to converge, whatever its initial state is, to a fraction of $\frac{\sqrt{2}}{2}$ agents in a given state. We hence show that some algebraic irrational values can be computed in this sense. We also give an asymptotic development of the convergence.

Our motivation is to show that protocols, considered with these two hypotheses (huge population, computing functions and not only predicates), have a rather different power.

We consider this paper as a first step towards understanding which numbers can be computed by such protocols. Whereas we prove in this paper that $\frac{\sqrt{2}}{2}$ can be computed, and whereas this is easy to see that computable numbers in this sense must be algebraic numbers of $[0, 1]$, we didn't succeed yet to characterize precisely computable numbers.

In this more long term objective, the aim of this current paper is first to discuss in which sense one can say that these protocols compute an irrational algebraic value such as $\frac{\sqrt{2}}{2}$, and second to study mathematically formally the convergence.

Our discussion is organized as follows. In Section 2, we present classical finite-size population protocols and related work. In Section 3, we present the considered system. In Section 4, as a preliminary discussion, we discuss what can be said when population is assumed to be finite. The rest of the paper is devoted to consider the case of an infinite population. To do so, we first do some mathematical computations in Section 5, in order to use a general theorem presented in Section 6 from [10] about approximation of diffusions. This theorem yields the proof of convergence in Section 7. We prove in Section 8 that this is even possible to use the same theorem to go further and get an asymptotic development of the convergence. Section 9 is devoted to a conclusion and a discussion.

2 Related Work

Population protocols have been introduced in [1]. In the paper, the authors proved that all semi-linear predicates can be computed but left open the question of their exact power. This was solved in [2], where it has been proved that no-more predicates can be computed.

The population protocol model was inspired in part by the work by Diamadi and Fischer on trust propagation in social networks [5]. The model proposed in [1] was motivated by the study of sensor networks in which passive agents were carried along by other entities. The canonical example given in this latter paper was sensors attached to a flock of birds.

Much of the work so far on population protocols has concentrated on characterizing what predicates on the input configurations can be stably computed in different variants of the models and under various assumptions, such as bounded-degree interaction graphs and random scheduling [3].

Variants considered includes restriction to one-way communications, restriction to particular interaction graphs, random interactions, self-stabilizing solutions through population protocols to classical problems in distributed algorithmic, the taking into account of various kind of failures of agents, etc. See survey [3]. As far as we know, a huge population hypothesis in the sense of this paper, has not been considered yet.

Notice that we assume that interactions happen in probabilistic way, according to some uniform law. In the original population protocol model, only specific fairness hypotheses were assumed on possible adversaries [1]. Somehow our notion of adversary is stronger: for finite state systems, this satisfies the fairness hypotheses of the original model, but for infinite state systems, we think that this becomes more natural to expect such a notion, since fairness hypotheses in the sense [1] become problematic to generalize. Notice that this notion of adversary has already be considered for finite state systems [3], in order to study speed of convergence of specific algorithms.

The result proved in this paper can be considered as a macroscopic abstraction of a system given by microscopic rules of evolutions. See survey [7] for general discussions about extraction of macroscopic dynamics.

Whereas the ordinary differential equation (9) can be immediately abstracted in a physicist approach from the dynamic (1), the formal mathematical equivalence of the two is not so immediate, and is somehow a strong motivation of this paper.

Actually, these problems seem to arise in many macroscopic justification of models from their microscopic description in experimental science: See for example the very instructive discussion in [8] about assumptions required for the justification of the Lotka-Volterra (predator-prey) model of population dynamics. In particular, observe that the fact that microscopic correlations must be neglected (i.e. $E[XY] = E[X]E[Y]$ is needed, where E is expectation). With a rather similar hypothesis (here assuming $E[p^2] = E[p]^2$), dynamic (9) is clear from rules (1). Somehow, we prove here that this hypothesis is not necessary for our system.

The techniques used in this paper are based on weak convergence techniques, introduced in [10], relating a stochastic differential equation (whose solutions are called diffusions) to approximations by a family of Markov processes. Refer also to [6] for an introduction to these techniques. The theorem used here is actually based on the presentation of [4] of a theorem from [10].

3 The Considered System

We now present our system, in a self-contained manner, to avoid to redefine formally population protocols. However, the reader can check that this is indeed a (non-stably-converging in the sense of [1]) population protocol.

We consider a set of n anonymous agents. Each agent can be in state $+$ or in state $-$. A configuration hence corresponds to an element of $S = \{+, -\}^n$. There are 2^n such configurations.

Suppose that time is discrete.

At each discrete round, two agents are paired. These two agents are chosen according to a uniform law (without choosing twice the same). The effect of a pairing is given by the following rules:

$$\begin{cases} ++ \to +- \\ +- \to ++ \\ -+ \to ++ \\ -- \to +- \end{cases} \tag{1}$$

These rules must be interpreted as follows: if an agent in state $+$ is paired with an agent in state $+$, then the second becomes $-$. If an agent in state $+$ is paired with an agent in state $-$, then the second becomes $+$, and symmetrically. If an agent in state $-$ is paired with an agent in state $-$, then the first becomes in state $+$.

We want to discuss the limit of the proportion $p(k)$ of agents in state $+$ in the population at discrete time k. If $n_+(k)$ denotes the number of agents in state $+$, and $n_-(k) = n - n_+(k)$ the number of agents in state $-$,

$$p(k) = \frac{n_+(k)}{n}.$$

The object of the rest of this paper is to show the convergence of p to $\frac{\sqrt{2}}{2}$ whenever k goes to infinity, and n goes to infinity.

4 A Preliminary Discussion: The Case of Finite Size Populations

Let us first restate what we are considering by discussing the case of a fixed n. Clearly, the previous rules of interactions can be considered as a description of a discrete time homogeneous Markov chain. This Markov chain has 2^n states corresponding to all configurations. Special configuration $s^- = (-, -, \cdots, -)$ where all agents are in state $-$ is immediately left with probability 1 to a configuration of $S^* = S - \{s^-\}$. Now, any configuration $s' \in S^*$ is clearly reachable from any configuration $s \in S^*$ with positive probability.

Hence, the sequence $p(k)_{k\geq 1}$ is an irreducible Markov chain on S^*.

Let us discuss the basic transition probabilities of this irreducible Markov chain.

At any time step, when selecting an agent in the soup uniformly, it will be in state + with probability $p(k)$, and in state $-$ with probability $1 - p(k)$.

The other agent with whom it will be paired is selected in the rest of the population:

- if the first agent was in state +, then the other will be in state + with probability $\frac{n_+(k)-1}{n-1} = p(k)\frac{n}{n-1} - \frac{1}{n-1}$, and in state $-$ with probability $\frac{n_-(k)}{n-1} = \frac{n}{n-1} - p(k)\frac{n}{n-1}$.

  Hence the probability that two agents in state + are paired, and that an agent in state + is paired to an agent in state $-$ are given respectively by

  $$\pi_{++} = p(k)^2 \frac{n}{n-1} - p(k)\frac{1}{n-1}$$

  and

  $$\pi_{+-} = p(k)\frac{n}{n-1} - p(k)^2\frac{n}{n-1}.$$

- Otherwise, the first agent is in state $-$. In this case, the other will be in state + with probability $\frac{n_+(k)}{n-1} = p(k)\frac{n}{n-1}$, and in state $-$ with probability $\frac{n_-(k)-1}{n-1} = 1 - p(k)\frac{n}{n-1}$.

  Hence the probability that an agent in state $-$ is paired with an agent in state +, and that an agent in state $-$ is paired with an agent in state $-$ are given respectively by

  $$\pi_{-+} = p(k)\frac{n}{n-1} - p(k)^2\frac{n}{n-1}$$

  and

  $$\pi_{--} = 1 + p(k)\frac{1-2n}{n-1} + p(k)^2\frac{n}{n-1}.$$

To any state $s \in S^*$ of the Markov chain is associated some proportion $p(k) = \frac{n_+}{n}$ of agents in state +, that takes value in $V = \{\frac{1}{n}, \frac{2}{n}, \frac{n-1}{n}, \cdots, 1\}$. Clearly, from above discussions, the Markov chain on S^* can be abstracted on an irreducible Markov chain on this latter set V. As it evolves on finite set V, it is positive recurrent.

The number of agents in state + is increased by one by the second, third and fourth rule, hence with probability

$$\pi_{+1} = \pi_{+-} + \pi_{-+} + \pi_{--} = 1 - \pi_{++},$$

and is decreased by one by the first rule, hence with probability

$$\pi_{-1} = \pi_{++}.$$

By ergodic theorem whatever the initial distribution of probability on states is, the sequence $p(k)$ will ultimately converge in law to the unique stationary distribution π of the Markov chain on V. Distribution π is given by the unique solution with $\sum_{i=1}^{n} \pi(\frac{i}{n}) = 1$ to the global balance equations

$$\pi(\frac{i}{n}) = \pi(\frac{i-1}{n})\pi_{+1} + \pi(\frac{i+1}{n})\pi_{-1},$$

for $i = 1, 2, \cdots, n$ (interpreting $\pi(0)$ and $\pi(\frac{n+1}{n})$ as 0).

As the unique solution to a rational system of equations is rational, the probabilities $\pi(\frac{i}{n})$ are rational, and hence expectation $E[p]$ on the stationary distribution, that can be computed from this stationary distribution by

$$E[p] = \sum_{i=1}^{n} \frac{i}{n}\pi(\frac{i}{n})$$

is rational.

Hence, when the population is finite, this is clear that the proportion of agents in state + converges in law to some rational value, that can be computed as above.

The purpose of the rest of the discussion, is to see that when n goes to infinity, the mean value of $p(k)$ converges to some irrational algebraic value, i.e. to $\frac{\sqrt{2}}{2}$.

Notice also that it follows from the fact that the chain restricted to S^* is irreducible that all configurations of S^* are visited with some positive probability. Hence, in the classical model of [1] this protocol cannot stably compute any non-trivial predicate. Our notion of convergence is different, and based on convergence towards limit distributions on proportions, which is a natural notion when considering huge populations.

5 Computing Expectation and Variance of Increments

As all rules increase or decrease by 1 the number of agents in state +, given $n_+(k)$, one knows that the increment $\Delta_n = n_+(k+1) - n_+(k)$ takes its value in $\{-1, 1\}$.

From previous discussions, we have:

$$\pi_{+1} = 1 - p(k)^2 \frac{n}{n-1} + p(k) \frac{1}{n-1}$$

and

$$\pi_{-1} = p(k)^2 \frac{n}{n-1} - p(k) \frac{1}{n-1}.$$

We get

$$E[\Delta_n | n_+(k)] = 1 \times \pi_{+1} - 1 \times \pi_{-1},$$

from which we get the fundamental equation at the source of the following discussion:

$$E[n_+(k+1) - n_+(k) | n_+(k)] = 1 - 2p(k)^2 \frac{n}{n-1} + p(k) \frac{2}{n-1} \tag{2}$$

Remark 1 *When n goes to infinity, this converges to $1 - 2p(k)^2$.*

Assuming that limit commutes, and that the limit p^* of $p(k)$ when k goes to infinity exists, it must cancel this quantity. Indeed, the system must converge to configuration(s) when one doesn't create nor destroy + in mean.

We get clearly that the limit can only be $p^* = \frac{\sqrt{2}}{2}$.

The remaining problem is hence to justify and discuss mathematically the convergence.

We will first compute

$$\begin{aligned} E[\Delta_n^2 | n_+(k)] &= 1 \times \pi_{+1} + 1 \times \pi_{-1} \\ &= 1. \end{aligned} \tag{3}$$

It follows, from Equations (2) and (3), that we have

$$E[p(k+1) - p(k) | p(k)] = \frac{1}{n}(1 - 2p(k)^2 \frac{n}{n-1} + p(k) \frac{2}{n-1}), \tag{4}$$

which yields the equivalent

$$nE[p(k+1) - p(k) | p(k)] \approx 1 - 2p(k)^2 \tag{5}$$

when n goes to infinity, and

$$E[(p(k+1) - p(k))^2 | p(k)] = \frac{1}{n^2}, \tag{6}$$

which yields the equivalent

$$nE[(p(k+1)-p(k))^2|p(k)] \approx \frac{1}{n}, \tag{7}$$

when n goes to infinity.

6 A General Theorem about Approximation of Diffusions

We will use the following theorem from [10]. We use here the formulation of it in [4] (Theorem 5.8 page 96).

Suppose that we have for all integer $n \geq 1$, an homogeneous Markov chain $(Y_k^{(n)})$ in $\mathbb{R}^d$ of transition $\pi^{(n)}(x,dy)$, that is to say so that the law of $Y_{k+1}^{(n)}$ conditioned by $Y_0^{(n)}, \cdots, Y_k^{(n)}$ depends only on $Y_k^{(n)}$ and is given, for all Borelian B, by

$$P(Y_{k+1}^{(n)} \in B|Y_k^{(n)}) = \pi^{(n)}(Y_k^{(n)}, B).$$

almost surely.

Define for $x \in \mathbb{R}^d$,

$$b^{(n)}(x) = n\int (y-x)\pi^{(n)}(x,dy),$$

$$a^{(n)}(x) = n\int (y-x)(y-x)^*\pi^{(n)}(x,dy),$$

$$K^{(n)}(x) = n\int (y-x)^3\pi^{(n)}(x,dy),$$

$$\Delta_\epsilon^{(n)}(x) = n\pi^{(n)}(x, B(x,\epsilon)^c),$$

where $B(x,\epsilon)^c$ is the complement of the ball centered in x of radius ϵ.

Define

$$X^{(n)}(t) = Y_{\lfloor nt \rfloor}^{(n)} + (nt - \lfloor nt \rfloor)(Y_{\lfloor nt+1 \rfloor}^{(n)} - Y_{\lfloor nt \rfloor}^{(n)}).$$

The coefficients $b^{(n)}$ and $a^{(n)}$ can be interpreted as the instantaneous drift and variance (or matrix of covariance) of $X^{(n)}$.

Theorem 1 (Theorem 5.8, page 96 of [4]) *Suppose that there exist some continuous functions a, b, such that for all $R < +\infty$,*

$$\lim_{n\to\infty} sup_{|x|\leq R}|a^{(n)}(x) - a(x)| = 0$$

$$\lim_{n\to\infty} sup_{|x|\leq R}|b^{(n)}(x) - b(x)| = 0$$

$$\lim_{n\to\infty} sup_{|x|\leq R}\Delta_\epsilon^{(n)} = 0, \forall \epsilon > 0$$

$$\sup_{|x|\leq R} K^{(n)}(x) < \infty.$$

With σ a matrix such that $\sigma(x)\sigma^*(x) = a(x)$, $x \in \mathbb{R}^d$, we suppose that the stochastic differential equation

$$dX(t) = b(X(t))dt + \sigma(X(t))dB(t), \qquad X(0) = x, \tag{8}$$

has a unique weak solution for all x. This is in particular the case, if it admits a unique strong solution.

Then for all sequence of initial conditions $Y_0^{(n)} \to x$, the sequence of random processes $X^{(n)}$ converges in law to the diffusion given by (8).

In other words, for all function $F : \mathcal{C}(\mathbb{R}^+, \mathbb{R}) \to \mathbb{R}$ bounded and continuous, one has

$$\lim_{n\to\infty} E[F(X^{(n)})] = E[F(X)].$$

7 Proving Convergence

Consider $Y_i^{(n)}$ as the homogeneous Markov chain corresponding to $p(k)$, when n is fixed. From previous discussions, $\pi^{(n)}(x,.)$ is a weighted sum of two Dirac that weight $x-\frac{1}{n}$ and $x+\frac{1}{n}$, with respective probabilities π_{-1} and π_{+1}, whenever x is of type $\frac{i}{n}$ for some i.

Set $a(x) = 1 - 2x^2$, and $b(x) = 0$. From the equivalent (5) and (7), we have clearly

$$\lim_{n\to\infty} sup_{|x|\leq R}|a^{(n)}(x) - a(x)| = 0$$
$$\lim_{n\to\infty} sup_{|x|\leq R}|b^{(n)}(x) - b(x)| = 0$$

for all $R < +\infty$.

Since the jumps of $Y^{(n)}$ are bounded in absolute value by $\frac{1}{n}$, $\Delta_\epsilon^{(n)}$ is null, as soon as $\frac{1}{n}$ is smaller than ϵ, and so

$$\lim_{n\to\infty} sup_{|x|\leq R}\Delta_\epsilon^{(n)} = 0, \forall \epsilon > 0$$

$$\sup_{|x|\leq R} K^{(n)}(x) < \infty$$

is easy to establish.

Now, (ordinary and deterministic) differential equation

$$dX(t) = (1 - 2X^2)dt \tag{9}$$

has an unique solution for any initial condition.

It follows from above theorem that the sequence of random processes $X^{(n)}$ defined by

$$X^{(n)}(t) = Y^{(n)}_{\lfloor nt \rfloor} + (nt - \lfloor nt \rfloor)(Y^{(n)}_{\lfloor nt+1 \rfloor} - Y^{(n)}_{\lfloor nt \rfloor})$$

converges in law to the unique solution of differential equation (9).

Clearly, all solutions of ordinary differential equation (9) converge to $\frac{\sqrt{2}}{2}$. Doing the change of variable $Z(t) = X(t) - \frac{\sqrt{2}}{2}$, we get

$$dZ(t) = (-2Z^2 + 2\sqrt{2}Z)dt, \tag{10}$$

that converges to 0.

Coming back to $p(k)$ using definition of $X^{(n)}(t)$, we hence get

Theorem 2 *We have for all t,*

$$p(\lfloor nt \rfloor) = \frac{\sqrt{2}}{2} + Z_n(t),$$

where $Z_n(t)$ converges in law when n goes to infinity to the (deterministic) solution of ordinary differential (10)*. Solutions of this ordinary differential equation go to* 0 *at infinity.*

This implies that $p(k)$ must converge to $\frac{\sqrt{2}}{2}$ when k and n go to infinity.

8 An Asymptotic Development of the Dynamic

This is actually possible to go further and prove the equivalent of a central limit theorem, or if one prefers, to do an asymptotic development of the convergence, in terms of stochastic processes.

As $p(k)$ is expected to converge to $\frac{\sqrt{2}}{2}$, consider the following change of variable:

$$Y^{(n)}(k) = \sqrt{n}(p(k) - \frac{\sqrt{2}}{2}).$$

The subtraction of $\frac{\sqrt{2}}{2}$ is here to get something centered, and the $\sqrt{n}$ factor is here in analogy with classical central limit theorem.

Clearly, $Y^{(n)}(.)$, that we will also note $Y(.)$ in what follows when n is fixed, is still an homogeneous Markov Chain.

We have

$$E[Y(k+1) - Y(k)|Y(k)] = \sqrt{n}(E[p(k+1) - p(k)|p(k)]),$$

hence, from (4),

$$E[Y(k+1) - Y(k)|Y(k)] = \frac{1}{\sqrt{n}}(1 - 2p(k)^2\frac{n}{n-1} + p(k)\frac{2}{n-1}).$$

Using $p(k) = \frac{\sqrt{2}}{2} + \frac{Y(k)}{\sqrt{n}}$, we get

$$\begin{aligned} E[Y(k+1) - Y(k)|Y(k)] = {} & \tfrac{\sqrt{2}-1}{\sqrt{n}(n-1)} + Y(k)(-\tfrac{2\sqrt{2}}{n-1} \\ & + \tfrac{2}{n(n-1)}) + Y(k)^2(-\tfrac{2}{\sqrt{n}(n-1)}) \end{aligned}$$

which yields the equivalent

$$nE[Y(k+1) - Y(k)|Y(k)] \approx -2\sqrt{2}Y(k)$$

when n goes to infinity.

We have

$$E[(Y(k+1) - Y(k))^2|Y(k)] = n(E[(p(k+1) - p(k))^2|p(k)]),$$

hence, from equation (6),

$$nE[(Y(k+1) - Y(k))^2|Y(k)] = 1.$$

Set $a(x) = -2\sqrt{2}x$, $b(x) = 1$.

From the above calculations we have clearly

$$\lim_{n\to\infty} sup_{|x|\leq R}|a^{(n)}(x) - a(x)| = 0$$

$$\lim_{n\to\infty} sup_{|x|\leq R}|b^{(n)}(x) - b(x)| = 0$$

for all $R < +\infty$.

Since the jumps of $Y^{(n)}$ are bounded in absolute value by $\frac{1}{\sqrt{n}}$, $\Delta_\epsilon^{(n)}$ is null, as soon as $\frac{1}{\sqrt{n}}$ is smaller than ϵ, and so

$$\lim_{n\to\infty} sup_{|x|\leq R}\Delta_\epsilon^{(n)} = 0, \forall\epsilon > 0$$

$$\sup_{|x|\leq R} K^{(n)}(x) < \infty$$

is still easy to establish.

Now stochastic differential equation

$$dX(t) = -2\sqrt{2}X(t)dt + dB(t) \tag{11}$$

is of a well-known type. This is an Orstein-Uhlenbeck process, i.e. a stochastic differential equation of type

$$dX(t) = -bX(t)dt + \sigma dB(t).$$

Such an equation is known to have a unique solution for all initial condition $X(0) = x$. This solution is given by (see e.g. [4])

$$X(t) = e^{-bt}X(0) + \int_0^t e^{-b(t-s)}\sigma dB(s).$$

It is known for these processes, that for all initial condition $X(0)$, $X(t)$ converges in law when t goes to infinity to the Gaussian $\mathcal{N}(0, \frac{\sigma^2}{2b})$. This latter Gaussian is invariant. See for e.g. [4].

We have all the ingredients to apply Theorem 1 again, and get:

Theorem 3 *We have for all t,*

$$p(\lfloor nt \rfloor) = \frac{\sqrt{2}}{2} + \frac{1}{\sqrt{n}}A_n(t),$$

where $A_n(t)$ converges in law to the unique solution of stochastic differential equation (11)*, and hence to the Gaussian $\mathcal{N}(0, \frac{\sqrt{2}}{8})$ when t goes to infinity.*

9 Conclusion

In this paper we considered a particular system of rules. This system describes a particular population protocol. These protocols have been introduced in [1] as a sensor network model. Whereas for original definitions of the latter paper it is not considered as (stably) convergent, we proved that it actually computes in some natural sense some irrational algebraic value: indeed, the proportion of agents in state + converges to $\frac{\sqrt{2}}{2}$, whatever the initial state of the system is.

One aim of this paper was to formalize the proof of convergence. We did it using a diffusion approximation technique, using a theorem due to [10]. We detailed fully the proof in order to convince our reader that our reasoning can be easily generalized to other kinds of rules of the same type. In particular, this is easy to derive from the protocol considered here another protocol that would compute $\sqrt{\sqrt{\frac{1}{2}}}$, by working with an alphabet made of pairs of states. Clearly, the arguments here would prove its convergence.

We consider this paper as a first step towards understanding which numbers can be computed by such protocols. Whereas we prove in this paper that $\frac{\sqrt{2}}{2}$ can be computed, and whereas this is easy to see that computable numbers in this sense must be algebraic numbers of $[0, 1]$, we didn't succeed yet to characterize precisely computable numbers.

References

[1] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. In *Twenty-Third ACM Symposium on Principles of Distributed Computing*, pages 290–299. ACM Press, July 2004.

[2] Dana Angluin, James Aspnes, and David Eisenstat. Stably computable predicates are semilinear. In *PODC '06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 292–299, New York, NY, USA, 2006. ACM Press.

[3] James Aspnes and Eric Ruppert. An introduction to population protocols. In *Bulletin of the EATCS*, volume 93, pages 106–125, 2007.

[4] F. Comets and T. Meyre. *Calcul stochastique et modeles de diffusions.* Dunod Paris, 2006.

[5] Z. Diamadi and M.J. Fischer. A simple game for the study of trust in distributed systems. *Wuhan University Journal of Natural Sciences*, 6(1-2):72–82, 2001.

[6] R. Durrett. *Stochastic Calculus: A Practical Introduction.* CRC Press, 1996.

[7] D. Givon, R. Kupferman, and A. Stuart. Extracting macroscopic dynamics: model problems and algorithms. *Nonlinearity*, 17(6):R55–R127, 2004.

[8] Annick Lesne. Discrete vs continuous controversy in physics. *Mathematical Structures in Computer Science*, 2006. In print.

[9] M. Presburger. Uber die Vollstandig-keit eines gewissen systems der Arithmetik ganzer Zahlen, in welchemdie Addition als einzige Operation hervortritt. *Comptes-rendus du I Congres des Mathematicians des Pays Slaves*, pages 92–101, 1929.

[10] D.W. Stroock and SRS Varadhan. *Multidimensional Diffusion Processes.* Springer, 1979.

Emergence as a Computability-Theoretic Phenomenon

S. Barry Cooper

Department of Pure Mathematics
University of Leeds, Leeds LS2 9JT, U.K.

Abstract

In dealing with emergent phenomena, a common task is to identify useful descriptions of them in terms of the underlying atomic processes, and to extract enough computational content from these descriptions to enable predictions to be made. Generally, the underlying atomic processes are quite well understood, and (with important exceptions) captured by mathematics from which it is relatively easy to extract algorithmic content.

A widespread view is that the difficulty in describing transitions from algorithmic activity to the emergence associated with chaotic situations is a simple case of complexity outstripping computational resources and human ingenuity. Or, on the other hand, that phenomena transcending the standard Turing model of computation, if they exist, must necessarily lie outside the domain of classical computability theory.

In this article we suggest that much of the current confusion arises from conceptual gaps and the lack of a suitably fundamental model within which to situate emergence. We examine the potential for placing emergent relations in a familiar context based on Turing's 1939 model for interactive computation over structures described in terms of reals. The explanatory power of this model is explored, formalising informal descriptions in terms of mathematical definability and invariance, and relating a range of basic scientific puzzles to results and intractable problems in computability theory.

Key words: Computability, emergence, definability, Turing invariance.

1 Computability and Emergence

Since the time of Turing, computability as a concept has hardened, become hedged around by its impressive technical development and its history, until its role from almost any viewpoint has become tangential to the very real mysteries of how one models the real universe. This turn of events has had an air of inevitability, in that even Turing, with his remarkable ability for clarifying concepts and basic questions, was unable to fully import his concerns about the nature of computability into the burgeoning formal framework of recursion theory. And many of those who took up the technical development of the subject not only lacked Turing's vision, but became diverted by the pure excitement and mathematical beauty of the new academic field. Thomas Kuhn's 'normal science' contains its own excitements and its minor paradigm shifts, as well as delivering safe research careers.

From the point of view of the logician, recursion theory concerns on the one hand a *computable* universe whose theory derives its significance from computer-scientific concerns, with a technical content owing only a very basic and vestigial debt to its logical origins. And on the other hand, exhibits an arcane preoccupation with the development of a theory of *incomputability*, for which its practitioners have no explanation or evidence for its existence in the material world. One may be uneasy about the public criticisms by Martin Davis, Stephen Simpson, and others (see [12]), but their views are widely respected.

This leaves many, with eyes wide enough open to see the accumulated evidence of real-world misbehaviour, looking elsewhere for models. Presented with phenomena with seemingly no hope of ever being reduced to a simple classical computational model, the natural alternative has been to develop models with direct links to quite particular instances of apparent incomputability in a physical setting. Much of this work, giving rise to a wide range of so-called 'new computational paradigms', has taken on a distinctly ad hoc aspect. Even though the theoretical underpinnings of this newness are absent — even the standard model of quantum computation is not free from continued scrutiny — the delivery of computational outcomes sufficiently separated from the model's real-world template is taken as a pointer to useful applications.

One can highlight three key challenges to a reductionist view of the computational content of the universe, and to the explanational potentialities of the computability framework. All three are familiar to the informed non-specialist, are strikingly hard for the specialist to deal with, and are associated with controversies, speculations, and a missing clarity which suggests a corresponding missing conceptual ingredient. Quantum phenomena, and the human brain, present the two most unavoidable challenges to the reductionist agenda. There are other relatively specific examples, such as the puzzle of the origins of life. But these are less dramatic, and less in the public domain. The third challenge — *emergence* — is at first sight less obviously disturbing, but is more prevalent, more *protean* in its manifestations, more theoretically deconstructable, and — ultimately — more likely to give rise to a basic theoretical model of wide application. And potentially of wide enough relevance to throw light on the two first and more immediate challenges to our understanding of the world.

Emergence lies at the core of a number of controversies in science, often used in a descriptive and speculative way to challenge more mechanistic and reductive attempts to interpret the universe. Out of this dichotomy arises a less-than-illuminating polarisation into a relative faithfulness to the simpler Laplacian constructs of the scientific age, and a contemporary counter-culture insistent on the essential mystery and predominance of emergent phenomena. The purpose of this article is to point to some sort of reconciliation, mediated by classical computability concepts going back to Turing — the unifying personality both in his overall concerns with computability, and in his breadth of interests, taking in his seminal work on emergence, in the form of his work on morphogenesis, and specifically phyllotaxis.

2 What is Emergence?

The term *emergence* is increasingly used in all sorts of contexts, often to describe any situation in which there appears to be a breakdown in reductionist explanation, or where there appears to be a global rather than purely local causal dynamic at work. This is how Stuart Kauffman [26] argues in his recent book on *Reinventing the Sacred: A New View of Science, Reason and Religion* (p.281):

> We are beyond reductionism: life, agency, meaning, value, and even consciousness and morality almost certainly arose naturally, and the evolution of the biosphere, economy, and human culture are stunningly creative often in ways that cannot be foretold, indeed in ways that appear to be partially lawless. The latter challenge to current science is radical. It runs starkly counter to almost four hundred years of belief that natural laws will be sufficient to explain what is real anywhere in the universe, a view I have called the Galilean spell. *The new view of emergence and ceaseless creativity partially beyond natural law is a truly new scientific worldview in which science itself has limits.* [My emphasis.]

If one is going to give emergence such a key role in restructuring the Laplacian model of science, and to come up with a suitably basic explanatory model, one needs to be more clear about what are the defining characteristics of emergent phenomena. Ronald, Sipper and Capcarrère [37] draw a parallel with the development of the Turing Test for intelligent machines, and use Turing's observer-based approach to formulate an *emergence test*. They comment that "overly facile use of the term emergence has made it controversial", and quote Arkin [2, p.105]:

> Emergence is often invoked in an almost mystical sense regarding the capabilities of behavior-based systems. Emergent behavior implies a holistic

> capability where the sum is considerably greater than its parts. It is true that what occurs in a behavior-based system is often a surprise to the system's designer, but does the surprise come because of a shortcoming of the analysis of the constituent behavioral building blocks and their coordination, or because of something else?

Ronald, Sipper and Capcarrère's emergence test "centers on an observer's avowed incapacity (amazement) to reconcile his perception of an experiment in terms of a global world view with his awareness of the atomic nature of the elementary interactions". As well as an observer, there is a 'designer' in the picture, whose existence is used to assist the description of certain qualifying features of the atomic interactions of the system to be tested. The test is comprised of three criteria:

(1) *Design:* The system has been constructed by the designer, by describing *local* elementary interactions between components (e.g., artificial creatures and elements of the environment) in a language $\mathcal{L}_1$.

(2) *Observation:* The observer is *fully aware* of the design, but describes *global* behaviors and properties of the running system, over a period of time, using a language $\mathcal{L}_2$.

(3) *Surprise:* The language of design $\mathcal{L}_1$ and the language of observation $\mathcal{L}_2$ are distinct, and the causal link between the elementary interactions programmed in $\mathcal{L}_1$ and the behaviors observed in $\mathcal{L}_2$ is *non-obvious* to the observer — who therefore experiences surprise. In other words, there is a cognitive dissonance between the observer's mental image of the system's design stated in $\mathcal{L}_1$ and his contemporaneous observation of the system's behavior stated in $\mathcal{L}_2$.

Ronald, Sipper and Capcarrère elaborate on this third condition to eliminate evanescent instances of surprise. Notice that one can apply versions of these criteria to a wide range of situations in which one is effectively capable of 'looking over the shoulder' of a putative designer — say one in which the local science is handed down to us by Nature, and is though to be well-understood, e.g., self-contained systems implementing Newtonian laws. The early history of chaos theory is replete with examples exhibiting the right quality of surprisingness, nicely communicated by the term 'strange attractor' coined [38] by David Ruelle and Floris Takens in 1971.

Of course, there is now quite a long history (see for example [1]) aimed at describing and improving our understanding of emergence, and as time goes on the observer 'surprise' criterion may not be as robust as the corresponding element of the Turing test. Turing himself played an innovative role in developing demystifying mathematics related to morphogenesis, and more particularly phyllotaxis, both in his seminal published paper [44] on the mathematical theory of biological pattern formation, and in his more opaque and incomplete writings contained in the posthumous collected works [45].

What is important though is not just the demystifying role of descriptions of emergent phenomena, but the *representational* functionality they point to. It is this latter aspect that takes us beyond emergence to a view of complexity in Nature in which emergence plays a key *inductive* role. And it is the first two of Ronald, Sipper and Capcarrère's conditions which make us look for something else within particular highly complex situations in which emergence clearly plays a role, though not a definitive one. These first two conditions also point to the a route to isolating the computational content of aspects of the physical universe which appear on the one hand to transcend standard computability-theoretic frameworks, and on the other entice reductionist explanations of increasing implausibility.

3 Representations, Recursions, Memetic Transmission

In [9] we considered the computational content of features of the real world, and more particularly, of developing computational practice. We looked at instances in which there appeared to be a fairly basic transgression of the 'Turing barrier' (defined by the limit of what is computable by an ideal computer as captured theoretically by a universal Turing machine), and more complex examples such as human intelligence and quantum uncertainty. In the former case one finds the emergence test broadly applicable, and in so doing can get a more informative theoretical grasp of what emergence is as a computational process.

For instance, going back to the influential 1988 paper of Paul Smolensky in *Behavioral and Brain Sciences*, we find [39, p.3] him examining a model qualifying under criteria one and two of the emergence test, along with an indication of an outcome which is surprising, judged according to computability-theoretic expectations:

> There is a reasonable chance that connectionist models will lead to the development of new somewhat-general-purpose self-programming, massively parallel analog computers, and a new theory of analog parallel computation: they may possibly even challenge the strong construal of Church's Thesis as the claim that the class of well-defined computations is exhausted by those of Turing machines.

Computational parallelism is an obviously important aspect of connectionist models and many others, but one needs to be careful about claiming that this is not simulated by a Turing machine. As is well-known (see, for example, David Deutsch [19, p.210]), the parallelism delivered by the standard model of quantum computation can be explained within the classical sequential model. A key ingredient, the addition of which does seem to stretch the classical Turing model, is that of internal connectivity. Goldin and Wegner [23] quote from Robin Milner's 1991 Turing Award lecture [30, p.80]:

> Through the seventies, I became convinced that a theory of concurrency and interaction requires a new conceptual framework, not just a refinement of what we find natural for sequential computing.

At the same time, parallelism and interactivity do seem to be basic features of situations exhibiting emergence.

Another idea which runs through a number of hypercomputational proposals, including Copeland's [15] rediscovery of oracle Turing machines, is that of adding *contextual* interactions. But as Davis has argued effectively, there is plenty of scope to widen the definition of what is 'internal' to a given system to bring a proposed new computational paradigm based on inadequately sourced oracles back into the classical fold.

But in [24], for instance, Goldin and Wegner are not just talking about parallelism and internal interactivity. And the inherent vagueness of examples they quote both stretch the mathematical analysis, and the reductionist agenda which feeds on that, to its limits:

> One example of a problem that is not algorithmic is the following instruction from a recipe [31]: 'toss lightly until the mixture is crumbly.' This problem is not algorithmic because it is impossible for a computer to know how long to mix: this may depend on conditions such as humidity that cannot be predicted with certainty ahead of time. In the function-based mathematical worldview, all inputs must be specified at the start of the computation, preventing the kind of feedback that would be necessary to determine when it's time to stop mixing.

But such interactions, such as those involving physical oracles as in [3], appear to take us beyond an analysis directly relevant to the computational ingredients of emergence as a basic computational phenomenon, and towards the more hybrid computational environments presaged at the end of the previous section.

A computational context which is both widely suspected of transcending the standard Turing model, and of whose inner workings we have a high level of detailed knowledge, is the human brain. And although we do know a great deal about the way the human brain works, it clearly fails to satisfy the first two conditions of the emergence test.

Part of the brain's potential for enrichment of our modelling of the computationally complex lies in the way it seems to successfully deal with the sort of imaging of the real world we would dearly like our computing machines to perform. More important, the brain shows the capacity to perform representations of mental imaging to enable recursive development of complex conceptual structures. At the same time, new techniques for relating structural and functional features of the brain, for example, using positron emission scan (PET) or a functional magnetic resonance imaging scan (fMRI), bring us much closer to obtaining useful models.

As we noted in [9], connectionist models of computation based on the workings of the human brain have developed in sophistication since Turing's [43] discussion of 'unorganised machines' (cf. Jack Copeland and Diane Proudfoot's article [16] *On Alan Turing's Anticipation of Connectionism*), and McCulloch and Pitts' initial paper [32] on neural nets. But despite the growth of computational neuroscience as an active research area, putting together ingredients from both artificial neural networks and neurophysiology, something does seem to be missing. As Rodney Brooks [5] says "neither AI nor Alife has produced artifacts that could be confused with a living organism for more than an instant." Or as Steven Pinker puts it: "... neural networks alone cannot do the job", going on to describe [34, p.124] "a kind of mental fecundity called recursion":

> We humans can take an entire proposition and give it a role in some larger proposition. Then we can take the larger proposition and embed it in a still-larger one. Not only did the baby eat the slug, but the father saw the baby eat the slug, and I wonder whether the father saw the baby eat the slug, the father knows that I wonder whether he saw the baby eat the slug, and I can guess that the father knows that I wonder whether he saw the baby eat the slug, and so on.

We are good at devising computational models capable of imaging, and of going some way to emulate how the brain comes up with neural patterns representing quite complex formations. But the mechanisms the brain uses to represent such patterns and relate them in complex ways is more elusive. What makes the sort of recursion Stephen Pinker has in mind so difficult to get to grips with at the structural level, is that it seems wound up with the puzzle of consciousness and its relationship to emotions and feelings. Antonio Damasio [17, p.169] describes the hierarchical development of a particular instance of consciousness within the brain (or, rather, 'organism'), interacting with some external object:

> ... both organism and object are mapped as neural patterns, in first-order maps; all of these neural patterns can become images. ... The sensorimotor maps pertaining to the object cause changes in the maps pertaining to the organism. ... [These] changes ... can be re-represented in yet other maps

> (second-order maps) which thus represent the relationship of object and organism. . . . The neural patterns transiently formed in second-order maps can become mental images, no less so than the neural patterns in first-order maps.

What is important here is the re-representation of neural patterns formed across some region of the brain, in such a way that they can have a computational relevance in forming new patterns. This is where the clear demarcation between computation and computational effect becomes blurred. The key conception is of computational loops incorporating these 'second-order' aspects of the computation itself. Building on this one can derive a plausible schematic picture of of the global workings of the brain.

Considering how complex a structure the human brain is, it is surprising one does not find more features needing reflecting in any basic computational model based on it. However, a thorough trawl through the literature, and one's own experiences, fails to bring to light anything that might be held up as computational principle transcending in a fundamental way what we have already identified. The key ingredients we expect in a model are imaging, parallelism, interconnectivity, and a counterpart to the second-order recursions pointed to above.

Mathematically, the imaging appears to be dependent on the parallelism and interconnectivity. This is what connectionist models are strong on. The recursions are not so easy to model, though. Looked at logically, one has representations of complex patternings of neural events underlying which there is no clear local mechanism, but for which one would expect a description in terms of the structures pertaining. Looked at physically, such descriptions appear to emerge, and be associated with (but not exclusively) the sort of non-linear mathematics governing the emergence of new relations from chaotic environments. This leads us to turn the picture of re-representations of mental imaging as a describable mapping on its head, and think (see [8]) in terms of descriptions in terms of a structure *defining*, and hence determining, the mental re-representations.

Looking at this more closely, what seems to be happening is that the brain stores away not just the image, but a route to accessing that image as a whole. This is what people who specialise in memorising very long numbers seem to display — rather than attempting to go directly into the detailed memory of a given number, they use simple representational tricks to call the entire number up. Here is how Damasio summarises the process (and the quotation from [17, p.170] is worth giving in full):

> As the brain forms images of an object — such as a face, a melody, a toothache, the memory of an event — and as the images of the object *af-*

> *fect* the state of the organism, yet another level of brain structure creates a swift nonverbal account of the events that are taking place in the varied brain regions activated as a consequence of the object-organism interaction. The mapping of the object-related consequences occurs in first-order neural maps representing the proto-self and object; the account of the *causal relationshi*p between object and organism can only be captured in second-order neural maps. . . . one might say that the swift, second-order nonverbal account narrates a story: *that of the organism caught in the act of representing its own changing state as it goes about representing something else.*

So what is going on here, and how can one make sense of this in a fundamental enough way to apply to it computability-theoretic analysis? Let us describe what seems to be the key idea in abstract terms, and then reinforce this powerful conceptual lever via something more familiar, but with new eyes.

What we first looked at, in a fairly schematic way, is a particular physical system whose constituents are governed by perfectly well-understood basic rules. These rules are usually *algorithmic*, in that they can be described in terms of functions simulatable on a computer, and their simplest consequences are mathematically predictable. But although the global behaviour of the system is *determined* by this algorithmic content, it may not itself be recognisably algorithmic. We certainly encounter this in the mathematics, which may be *nonlinear* and not yield the exact solutions needed to retain predictive control of the system. We may be able to come up with a perfectly precise *description* of the system's development which does not have the predictive — or algorithmic — ramifications the atomic rules would lead us to expect.

If one is just looking for a broad understanding of the system, or for a prediction of selected characteristics, the description may be sufficient. Otherwise, one is faced with the practical problem of extracting some hidden algorithmic content, perhaps via useful approximations, special cases, or computer simulations. Geroch and Hartle [22] discuss this problem in their 1986 paper, in which they suggest that "quantum gravity does seem to be a serious candidate for a physical theory for whose application there is no algorithm."

For the logician, this is a familiar scenario, for whom something describable in a structure is said to be *definable*. The difference between computability and definability is well-known. For example, if you go to any basic computability text (e.g., Cooper [7]) you will find in the *arithmetical hierarchy* a usable metaphor for what is happening here. What the arithmetical hierarchy encapsulates is the smallness of the computable world in relation to what we can describe. And Post's Theorem [35] shows us how language can be used to progressively describe increasingly incomputable objects and phenomena within computable structures. An analysis of lower levels of the hierarchy even gives us a clue to the formal role of computable approximations in constraining objects computably beyond our reach.

Now, the important thing to notice is that a description in some language can be viewed as being essentially *a code for an algorithm for reconstruction meaning from the real world within the human brain.* More precisely, a description conveys an epistemological algorithm which enables us to emulate emergent aspects, non-algorithmic, aspects of the world within the architecture of the brain. Key to this is the logical structure of the relevant word, sentence, or more extensive module of language. This, of course, is why certain ideas or human creations have *memetic content.* They come with a representation of, a recipe for, their mental recreation and simulation. The simulated phenomenon may be far from being algorithmic in its full manifestation, but the brain may be able to by-pass the computational barriers via an algorithmic device for activating and directing the brain's capacity for reproducing its own emergent features.

Of course, this process depends on humanly constructed language. But the universe has the capacity to handle descriptions, memetic content, and codings for algorithms which perform hugely sophisticated tasks, in a wide spectrum of situations, even though this may be via ad hoc emergent language of its own. Probably the most familiar example of this is the reproduction of various life forms via chromosomes and other genetic materials. A chromosome is a structured package of DNA and DNA-bound protein, involving genes, regulatory elements and other nucleotide sequences. Its coding functionality has algorithmic content, enabling the reproduction of complex aspects of the world — but this only within a context which is not obviously algorithmic, and which seems to ride upon undeniably emergent processes. Another example, involving the human brain, but not a particular language, is the process whereby experts in such tasks remember long seemingly random numbers. This is commonly achieved by algorithmically coding the details of the numbers into images simulable in the brain, the simulation itself being dependent upon higher order mental processes.

In order to associate a sufficiently basic model with such situations, which replaces the simple Laplacian determinism captured via Turing computability, one needs to look more closely at how science describes the world, and at the scientist's historic agenda. In particular, we will need to look at Turing's 1939 extension of his basic machine model of computation. The aim will be to go beyond an analysis of the computability-theoretic content of emergence, to that leading to a better understanding of the computational role of emergence in the wider context.

4 The Turing Model

Turing's extended [42] 1939 model, able to capture the algorithmic content of those structures which are presented in terms of real numbers can be seen in implicit form in Newton's *Principia*, published some 272 years earlier. Newton's work established a more intimate relationship between mathematics and science, and one which held the attention of Turing, in various guises, throughout his short life (see Hodges [25]). Just as the history of arithmetically-based algorithms, underlying many human activities, eventually gave rise to models of computation such as the Turing machine, so the oracle Turing machine schematically addresses the scientific focus on the extraction of predictions governing the form of computable relations over the reals. Whereas the inputting of data presents only time problems for the first model, the second model is designed to deal with possibly incomputable inputs, or at least inputs for which we do not have available an algorithmic presentation. One might reasonably assume that data originating from observation of the real world carries with it some level of computability, but we are yet to agree a mathematical model of physical computation which dispenses with the relativism of the oracle Turing machine. In fact, even as the derivation of recognisable incomputability in mathematics arises from quantification over algorithmic objects, so definability may play an essential role in fragmenting and structuring the computational content of the real world. The Turing model of computability over the natural numbers appears to many people to be a poor indicator of what to expect in science.

Typically, specialist computability theorists are loath to speculate about real-world significance for their work. Since the time of Turing, the theory of computability has taken on a Laputa-like[1] aspect in the eyes of many people, an arcane world disconnected from naturally arising information. Below, we look at Post's legacy of relating computability-theoretic concepts to intuitively immediate information content, and examine how that can be further extended to an informative relationship with the mathematics of contemporary science.

The oracle Turing machine, which made its first appearance in Turing [42], should be familiar enough. The details are not important, but can be found in most reasonable introductions to computability (see for instance [7]). One just needs to add to the usual picture of a Turing machine the capacity for questioning an oracle set about the membership status of individual natural numbers.

The basic form of the questioning permitted is modelled on that of everyday scientific practice. This is seen most clearly in today's digital data gathering,

[1] Swift even has a Laputan professor introduce Gulliver to *The Engine*, an (appropriately useless) early anticipation of today's computing machines, and more.

whereby one is limited to receiving data which can be expressed, and transmitted to others, as information essentially finite in form. But with the model comes the capacity to collate data in such a way as enable us to deal with arbitrarily close approximations to infinitary inputs and hence outputs, giving us an exact counterpart to the computing scientist working with real-world observations. If the different number inputs to the oracle machine result in 0-1 outputs from the corresponding Turing computations, one can collate the outputs to get a binary real computed from the oracle real, the latter now viewed as an input. This gives a partial computable functional Φ, say, from reals to reals, which may sometimes be described as a *Turing reduction*.

As usual, one cannot computably know when the machine for Φ computes on a given natural number input, so Φ may not always give a fully defined real output. So Φ may be partial. One can computably list all oracle machines, and so index the infinite list of all such Φ, but one cannot computably sift out the partial Φ's from the list.

Anyway, put $\mathbb{R}$ together with this list, and we get the Turing Universe. That is, we obtain a structure involving information in the form of real numbers, algorithmically related by all possible Turing reductions. Depending on one's viewpoint, this is either a rather reduced scientific universe (if you are a poet, a philosopher, or a string-theorist), or (if one is vainly looking for the richness of algorithmic content contained on our list in the physical context, being familiar with the richness of emergent structure in the Turing universe) a much expanded one. But we will defer difficult comparisons between the information content of the Turing universe and that of the physical universe until later. For the moment we will follow Emil Post in his search for the informational underpinnings of computational structure in a safer mathematical context.

Post's first step was to gather together binary reals which are computationally indistinguishable from each other, in the sense that they are mutually Turing computable from each other. Mathematically, this delivered a more standard mathematical structure to investigate — the familiar upper semi-lattice of the *degrees of unsolvability*, or *Turing degrees*. There is no simple scientific counterpart of the mathematical model, or any straightforward justification for what Post did with the Turing universe for perfectly good mathematical reasons — if one wants to get a material avatar of the Turing landscape one needs both a closer and a more comprehensive view of the physical context.

5 Definability in Science

Schematically, any causal context framed in terms of everyday computable mathematics can be modelled in terms of Turing reductions. Then emergence can be formalised as definability over the appropriate substructure of the Turing universe; or more generally, as invariance under automorphisms of the Turing universe. Simple and fundamental as the notion of definability is, and basic as it is to everyday thought and discourse, as a concept it is not well understood outside of logic. This is seen most strikingly in the physicists' apparent lack of awareness of the concept in interpreting the collapse of the wave function. Quantum decoherence and the many-worlds hypothesis comprise a far more outlandish interpretive option than does speculating that measurements, in enriching an environment, merely lead to an assertion of definability. It appears a sign of desperation to protect consistent histories by inventing new universes, when the mathematics of our observable universes already contains a straightforward explanation. We have argued (see for instance [13]) that many scientific puzzles can be explained in terms of failures of definability in different contexts, and that the key task is to identify useful theoretical models within which to investigate the nature of definability more fully. One of the most relevant of these models has to be that of Turing, based as it is on a careful analysis of the characteristics of algorithmic computation.

This brings us to a well-known and challenging research programme, initiated by Hartley Rogers in his 1967 paper [36], in which he drew attention to the fundamental problem of characterising the Turing invariant relations. Again, the intuition is that these are key to pinning down how basic laws and entities emerge as mathematical constraints on causal structure. It is important to notice how the richness of Turing structure discovered so far becomes the raw material for a multitude of non-trivially definable relations, matching in its complexity what we attempt to model.

Unfortunately, the current state of Rogers' programme is not good. For a number of years research in this area was dominated by a proposal originating with the Berkeley mathematician Leo Harrington, which can be (very) roughly stated:

Bi-interpretability Conjecture: *The Turing definable relations are exactly those with information content describable in second-order arithmetic.*

Most importantly, bi-interpretability is not consistent with the existence of non-trivial Turing automorphisms. Despite decades of work by a number of leaders in the field, the exact status of the conjecture is still a matter of controversy.

For those of us who have grown up with Thomas Kuhn's 1962 book [29] on the structure of scientific revolutions, such difficulties and disagreements are not seen as primarily professional failures, or triggers to collective shame (although they may be that too), but rather signs that something scientifically important is at stake. A far more public controversy currently shapes developments around important issues affecting theoretical physics — see, for example the recent books of Lee Smolin [40] and Peter Woit [47].

This turns out to be very relevant to our theme of the importance of fundamental notions, such as that of mathematical definability, to the formation of basic scientific theories. In this context, the specific focus on string theory of the above-mentioned books of Smolin and Woit is important, given that string theory was initially intended to remedy a number of inadequacies in current scientific thinking, without really getting to grips with fundamental issues. Our argument is that string theory does very validly point towards a substitution of abstract mathematics for inaccessible observational data. And that it has produced some very beautiful and useful mathematics, and widened our conceptual horizons in relation to models of the universe. But — that it has failed to enlist notions of global definability to pin down important elements of the real world.

As Peter Woit [47, p.1] describes, according to purely pragmatic criteria particle physics has produced a standard model which is remarkably successful, and has great predictive power:

> By 1973, physicists had in place what was to become a fantastically successful theory of fundamental particles and their interactions, a theory that was soon to acquire the name of the standard model. Since that time, the overwhelming triumph of the standard model has been matched by a similarly overwhelming failure to find any way to make further progress on fundamental questions.

The reasons why people are dissatisfied echo misgivings going back to Einstein himself [20, p.63]:

> ... I would like to state a theorem which at present can not be based upon anything more than upon a faith in the simplicity, i.e. intelligibility, of nature ... nature is so constituted that it is possible logically to lay down such strongly determined laws that within these laws only rationally completely determined constants occur (not constants, therefore, whose numerical value could be changed without destroying the theory) ...

If one really does have a satisfying description of how the universe is, it should not contain arbitrary elements with no plausible explanation. In particular, a theory containing arbitrary constants, which one adjusts to fit the intended interpretation of the theory, is not complete. And as Woit observes:

> One way of thinking about what is unsatisfactory about the standard model is that it leaves seventeen non-trivial numbers still to be explained, ...

At one time, it had been hoped that string theory would supply a sufficiently fundamental framework to provide a much more coherent and comprehensive description, in which such arbitrary ingredients were properly pinned down. But despite its mathematical attractions, there are growing misgivings about its claimed status as “the only game in town” as a unifying explanatory theory. Here is how one time string theorist Daniel Friedan [21] combatively puts it:

> The longstanding crisis of string theory is its complete failure to explain or predict any large distance physics. . . . String theory is incapable of determining the dimension, geometry, particle spectrum and coupling constants of macroscopic spacetime. . . . The reliability of string theory cannot be evaluated, much less established. String theory has no credibility as a candidate theory of physics.

Smolin starts his book [40]:

> From the beginning of physics, there have been those who imagined they would be the last generation to face the unknown. Physics has always seemed to its practitioners to be almost complete. This complacency is shattered only during revolutions, when honest people are forced to admit that they don’t know the basics.

He goes on to list what he calls the “five great [unsolved] problems in theoretical physics”. Gathering these together, and slightly editing, they are [40, pp.5-16]:

1. Combine general relativity and quantum theory into a single theory that can claim to be the complete theory of nature.
2. Resolve the problems in the foundations of quantum mechanics.
3. The unification of particles and forces problem: Determine whether or not the various particles and forces can be unified in a theory that explains them all as manifestations of a single, fundamental entity.
4. Explain how the values of the free constants in the standard model of physics are chosen in nature.
5. Explain dark matter and dark energy. Or, if they do not exist, determine how and why gravity is modified on large scales.

That each of these questions can be framed in terms of definability is not so surprising, since that is exactly how, essentially, they are approached by researchers. The question is the extent to which progress is impeded by a lack of consciousness of this fact, and an imperfect grip of what is fundamental. Quoting Einstein again (from a letter to Robert Thornton, dated 7 December 1944, Einstein Archive 61-754), this time on the relevance of a philosophical approach to physics:

> So many people today – and even professional scientists – seem to me like someone has seen thousands of trees but has never seen a forest. A knowl-

> edge of the historical and philosophical background gives that kind of independence from prejudices of his generation from which most scientists are suffering. This independence created by philosophical insight is – in my opinion – the mark of distinction between a mere artisan or specialist and a real seeker after truth.

Smolin's comment [40, p.263] is in the same direction, though more specifically directed at the string theorists:

> The style of the string theory community ... is a continuation of the culture of elementary-particle theory. This has always been a more brash, aggressive, and competitive atmosphere, in which theorists vie to respond quickly to new developments ... and are distrustful of philosophical issues. This style supplanted the more reflective, philosophical style that characterized Einstein and the inventors of quantum theory, and it triumphed as the center of science moved to America and the intellectual focus moved from the exploration of fundamental new theories to their application.

So what is it that is fundamental that is being missed? For Smolin [40, p.241], it is *causality*:

> It is not only the case that the spacetime geometry determines what the causal relations are. This can be turned around: Causal relations can determine determine the spacetime geometry ... Its easy to talk about space or spacetime emerging from something more fundamental, but those who have tried to develop the idea have found it difficult to realize in practice. ... We now believe they failed because they ignored the role that causality plays in spacetime. These days, many of us working on quantum gravity believe that *causality itself is fundamental* – and is thus meaningful even at a level where the notion of space has disappeared.

Citing Penrose as an early champion of the role of causality, he also mentions Rafael Sorkin, Fay Dowker, and Fotini Markopoulou, known in this context for their interesting work on causal sets (see [4]), which abstract from causality relevant aspects of its underlying ordering relation. Essentially, causal sets are partial orderings which are locally finite, providing a model of spacetime with built-in discreteness. Despite the apparent simplicity of the mathematical model, it has had striking success in approximating the known characteristics of spacetime. An early prediction, in tune with observation, concerned the value of Einstein's cosmological constant.

Of course, this preoccupation with causality might suggest to a logician a need to also look at its computational content. Smolin's comment that "Causal relations can determine the spacetime geometry" touches on one of the biggest disappointments with string theory, which turns out to be a 'background dependant' theory with a vengeance — one has literally thousands of candidate

Calabi-Yau spaces for shaping the extra dimensions of superstring theory. In current superstring models, Calabi-Yau manifolds are those qualifying as possible space formations for the six hidden spatial dimensions, their undetected status explained by the assumption of their being smaller than currently observable lengths.

Ideally, a truly fundamental mathematical model should be background independent, bringing with it a spacetime geometry arising from within.

6 The Emergence-Definability Symbiosis

There are obvious parallels between the Turing universe and the material world. Each of which in isolation, to those working with specific complexities, may seem superficial and unduly schematic. But the lessons of the history of mathematics and its applications is that the simplest of abstractions can yield unexpectedly far-reaching and deep insights into the nature of the real world. The main achievement of the Turing model, and its definable content, is to illuminate and structure the role of computability theoretic expressions of emergence.

At the most basic level, science describes the world in terms of real numbers. This is not always immediately apparent, any more that the computer on ones desk is obviously an avatar of a universal Turing machine. Nevertheless, scientific theories consist, in their essentials, of postulated relations upon reals. These reals are abstractions, and do not come necessarily with any recognisable metric. They are used because they are the most advanced presentational device we can practically work with. There is no faith that reality itself consists of information presented in terms of reals. In fact, those of us who believe that mathematics is indivisible, no less in its relevance to the material world, have a due humility about the capacity for our science to capture more than a surface description of reality.

Some scientists would take us in the other direction, and claim that the universe is actually finite, or at least countably discrete. We have argued elsewhere (see for example [14]) that to most of us a universe without algorithmic content is inconceivable. And that once one has swallowed that bitter pill, infinitary objects are not just a mathematical convenience (or inconvenience, depending on one's viewpoint), but become part of the mathematical mold on which the world depends for its shape. As it is, we well know how essential algorithmic content is to our understanding of the world. The universe comes with recipes for doing things. It is these recipes which generate the rich information content we observe, and it is reals which are the most capacious receptacles we can humanly carry our information in, and practically unpack.

Globally, there are still many questions concerning the extent to which one can extend the scientific perspective to a comprehensive presentation of the universe in terms of reals — the latter being just what we need to do in order to model the immanent emergence of constants and natural laws from an entire universe. Of course, there are many examples of presentations entailed by scientific models of particular aspects of the real world. But given the fragmentation of science, is fairly clear that less natural presentations may well have an explanatory role, despite their lack of a role in practical computation.

The natural laws we observe are largely based on algorithmic relations between reals. For instance, Newtonian laws of motion will computably predict, under reasonable assumptions, the state of two particles moving under gravity over different moments in time. And the character of the computation involved can be represented as a Turing functional over the reals representing different time-related two-particle states. One can point to physical transitions which are not obviously algorithmic, but these will usually be composite processes, in which the underlying physical principles are understood, but the mathematics of their workings outstrip available analytical techniques. Over forty years ago, Georg Kreisel [27] distinguished between classical systems and *co-operative phenomena* not known to have Turing computable behaviour, and proposed [28, p.143, Note 2] a collision problem related to the 3-body problem, which might result in "an analog computation of a non-recursive function (by repeating collision experiments sufficiently often)". However, there is a qualitatively different apparent breakdown in computability of natural laws at the quantum level — the *measurement problem* challenges us to explain how certain quantum mechanical probabilities are converted into a well-defined outcome following a measurement. In the absence of a plausible explanation, one is denied a computable prediction. The physical significance of the Turing model depends upon its capacity for explaining what is happening here. If the phenomenon is not composite, it does need to be related in a clear way to a Turing universe designed to model computable causal structure. We will need to talk more about definability and invariance.

For the moment, let us think in terms of what an analysis of the automorphisms of *any* sufficiently comprehensive, sufficiently fundamental, mathematical model of the material universe might deliver.

Let us first look at the relationship between automorphisms and many-worlds. When one says "I tossed a coin and it came down heads, maybe that means there is a parallel universe where I tossed the coin and it came down tails", one is actually predicating a large degree of correspondence between the two parallel universes. The assumption that *you* exist in the two universes puts a huge degree of constraint on the possible differences — but nevertheless, some relatively minor aspect of our universe has been rearranged in the parallel one. There are then different ways of relating this to the mathematical concept of an automorphism. One could say that the two parallel worlds are actually isomorphic, but that the structure was not able to *define* the outcome of the coin toss. So it and its consequences appear differently in the two worlds. Or one could say that what has happened is that the worlds are *not* isomorphic, that actually we were able to change quite a lot, without the parallel universe looking very different, and that it was these fundamental but hidden differences which forces the worlds to be separate and not superimposed, quantum fashion. The second view is more consistent with the view of quantum ambiguity displaying a failure of definability. The suggestion here being that the observed existence of a particle (or cat!) in two different states at the same time merely exhibits an automorphism of our universe under which the classical level is rigid (just as the Turing universe displays rigidity above $\mathbf{0}''$) but under which the sparseness of defining structure at the more basic quantum level enables the automorphism to re-represent our universe, with everything at our level intact, but with the particle in simultaneously different states down at the quantum level. And since our classical world has no need to decohere these different possibilities into parallel universes, we live in a world with the automorphic versions superimposed. But when we make an observation, we establish a link between the undefined state of the particle and the classical level of reality, which destroys the relevance of the automorphism. To believe that we now get parallel universes in which the alternative states are preserved, one now needs to decide how much else one is going to change about our universe to enable the state of the particle destroyed as a possiblity to survive in the parallel universe — and what weird and wonderful things one must accommodate in order to make that feasible. It is hard at this point to discard the benefits brought by a little mathematical sophistication. Quantum ambiguity as a failure of definability is a far more palatable alternative than the invention of new worlds of which we have no evidence or scientific understanding.

Another key conceptual element in the drawing together of a global picture of our universe with a basic mathematical model is the correspondence between emergent phenomena and definable relations. This gives us a framework within which to explain the particular forms of the physical constants and natural laws familiar to us from the standard model science currently provides. It goes some way towards substantiating Penrose's [33, pp$\dot{1}$06-107] 'strong determinism', according to which "all the complication, variety and apparent randomness that we see all about us, as well as the precise physical laws, are all exact and unambiguous consequences of one single coherent mathematical structure" — and repairs the serious failure of the standard model pointed to by researchers such as Smolin and Woit. It also provides a hierarchical model of the fragmentation of the scientific enterprise. This means that despite the causal connections between say particle physics and the study of living organisms, the corresponding disciplines are based on quite different basic entities and natural laws, and there is no feasible and informative reduction of one to another. The entities in one field may emerge through phase transitions characterised in terms of definable relations in the other, along with their distinct causal structures. In this context, it may be that the answer to Smolin's first 'great unsolved problem in theoretical physics' consists of an explanation of why there is no single theory (of the kind that makes useful predictions) combining general relativity and quantum theory.

For further discussion of such issues, see [6], [9], [10], [11], [13] and [14].

References

[1] I. Adler, D. Barab and R. V. Jean, A history of the study of phyllotaxis, *Annals of Botany*, **80** (1997), 231–244.

[2] R. C. Arkin, *Behaviour-Based Robotics,* M.I.T. Press, Cambridge, MA, 1998.

[3] E. J. Beggs and J. V. Tucker, Experimental computation of real numbers by Newtonian machines, to appear in *Proceedings of the Royal Society Series A*.

[4] L. Bombelli, J. Lee, D. Meyer and R. D. Sorkin, Spacetime as a causal set, Phys. Rev. Lett. **59** (1987), 521–524.

[5] R. Brooks, The relationship between matter and life. *Nature* **409** (2001), 409–411.

[6] S. B. Cooper, Clockwork or Turing U/universe? - Remarks on causal determinism and computability, in *Models and Computability* (S. B. Cooper and J. K. Truss, eds.), London Mathematical Society Lecture Notes Series 259, Cambridge University Press, Cambridge, New York, Melbourne, 1999, pp.63–116.

[7] S. B. Cooper, *Computability Theory,* Chapman & Hall/CRC, Boca Raton, London, New York, Washington, D.C., 2004.

[8] S. B. Cooper, Computability and emergence. In *Mathematical Problems from Applied Logic. New Logics for the XXI-st Century II* (eds. Gabbay D., Goncharov S., Zakharyaschev M.) Kluwer/Springer International Mathematical Series, Vol. 5, 2005.

[9] S. B. Cooper, Definability as hypercomputational effect, Applied Mathematics and Computation, **178** (2006), 72–82.

[10] S. B. Cooper, How Can Nature Help Us Compute?, in *SOFSEM 2006: Theory and Practice of Computer Science - 32nd Conference on Current Trends in Theory and Practice of Computer Science, Merin, Czech Republic, January 2006* (J. Wiedermann, J. Stuller, G. Tel, J. Pokorny, M. Bielikova, eds.), Springer Lecture Notes in Computer Science No. 3831, 2006, pp.1–13.

[11] S. B. Cooper, Computability and emergence, in *Mathematical Problems from Applied Logic I. Logics for the XXIst Century* (D.M. Gabbay, S.S. Goncharov, M. Zakharyaschev, eds.), Springer International Mathematical Series, Vol. 4, 2006, pp. 193–231.

[12] S. B. Cooper, The Incomputable Alan Turing, in the *Proceedings of Alan Mathison Turing 2004: A celebration of his life and achievements, Manchester University, 5 June, 2004* (Janet Delve and Jeff Paris, eds.), electronically published by the British Computer Society (2008): http://www.bcs.org/server.php?show=nav.9917.

[13] S. B. Cooper, Extending and interpreting Post's Programme, to appear.

[14] S. B. Cooper and P. Odifreddi, Incomputability in Nature, in *Computability and Models* (S.B. Cooper and S.S. Goncharov, eds.), Kluwer Academic/Plenum, New York, Boston, Dordrecht, London, Moscow, 2003, pages 137–160.

[15] J. Copeland, Turing's O-machines, Penrose, Searle, and the brain. *Analysis*, **58** (1998), 128–38.

[16] J. Copeland and D. Proudfoot, On Alan Turing's anticipation of connectionism. *Synthese*, **108** (1996), 361–377. Reprinted in *Artificial Intelligence: Critical Concepts in Cognitive Science* (R. Chrisley, ed), Volume 2: Symbolic AI. Routledge, London, 2000.

[17] A. Damasio, *The Feeling Of What Happens.* Harcourt, Orlando, FL., 1999.

[18] M. Davis (Ed.), *Solvability, Provability, Definability: The Collected Works of Emil L. Post*, Birkhäuser, Boston, Basel, Berlin, 1994.

[19] Deutsch D. (1997) *The Fabric of Reality.* Penguin, London, New York.

[20] A. Einstein, Autobiographical Notes, in *Albert Einstein: Philosopher-Scientist* (P. Schilpp, ed.), Open Court Publishing, 1969.

[21] D. Friedan, A Tentative Theory of Large Distance Physics, J. High Energy Phys. JHEP10 (2003) 063.

[22] R. Geroch and J. B. Hartle, Computability and physical theories, *Foundations of Physics* **16** (1986), 533–550.

[23] D. Goldin and P. Wegner, Computation Beyond Turing Machines: seeking appropriate methods to model computing and human thought. *Communications of the ACM* **46** (2003), 100–102.

[24] D. Goldin and P. Wegner, The Church-Turing Thesis: Breaking the Myth, in *CiE 2005: New Computational Paradigms: Papers presented at the conference in Amsterdam, June 8–12, 2005* (Cooper S. B., Löwe B., Torenvliet L., eds.), Lecture Notes in Computer Science No. 3526, Springer-Verlag, 2005.

[25] A. Hodges, *Alan Turing: The Enigma,* Vintage, London, Melbourne, Johannesburg, 1992.

[26] S. A. Kauffman, *Reinventing the Sacred: A New View of Science, Reason and Religion,* Basic Books, New York, 2008.

[27] G. Kreisel, Mathematical logic: What has it done for the philosophy of mathematics?, in *Bertrand Russell, Philosopher of the Century* (R. Schoenman, ed.), Allen and Unwin, London, 1967, pp. 201–272.

[28] G. Kreisel, Church's Thesis: a kind of reducibility axiom for constructive mathematics, in *Intuitionism and proof theory: Proceedings of the Summer Conference at Buffalo N.Y. 1968* (A. Kino, J. Myhill and R.E. Vesley, eds.), North-Holland, Amsterdam, London, 1970, pp. 121–150.

[29] T. S. Kuhn, *The Structure of Scientific Revolutions*, Third edition 1996, University of Chicago Press, Chicago, London.

[30] R. Milner, Elements of interaction:Turing award lecture. *Communications of the ACM* **36** (1993), 78–89.

[31] D. Knuth, *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*, Addison-Wesley, 1968.

[32] W. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.* **5** (2003), 115–133.

[33] R. Penrose, Quantum physics and conscious thought, in *Quantum Implications: Essays in honour of David Bohm* (BJ̇. Hiley and F.D. Peat, eds.), Routledge & Kegan Paul, London, New York, 1987, pp. 105–120.

[34] S. Pinker, *How the Mind Works.* W. W. Norton, New York, 1997.

[35] E. L. Post, Recursively enumerable sets of positive integers and their decision problems, *Bulletin of the American Mathematical Society*, **50** (1944), 284–316; reprinted in E. L. Post, *Solvability, Provability, Definability: The Collected Works of Emil L. Post*, pages 461–494.

[36] H. Rogers, Jr., Some problems of definability in recursive function theory, in *Sets, Models and Recursion Theory* (JṄĊrossley, ed)̇, Proceedings of the Summer School in Mathematical Logic and Tenth Logic Colloquium, Leicester, August–September, 1965, North-Holland, Amsterdam, pp. 183–201.

[37] E. M. A. Ronald, M. Sipper and M. S. Capcarrère, Design, observation, surprise! A test of emergence, Artificial Life, **5** (1999), 225–239.

[38] D. Ruelle and F. Takens, On the nature of turbulence, *Communications of Mathematical Physics*, **20** (1971), 167–192.

[39] P. Smolensky, On the proper treatment of connectionism, *Behavioral and Brain Sciences* **11** (1988), 1–74.

[40] L. Smolin, *The Trouble With Physics: The Rise of String Theory, the Fall of Science and What Comes Next,* Allen Lane/Houghton Mifflin, London, New York, 2006.

[41] A. Turing, On computable numbers, with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society*, Vol. 42, 1936–37, pages 230–265; reprinted in A. M. Turing, *Collected Works: Mathematical Logic*, pages 18–53.

[42] A. Turing, Systems of logic based on ordinals, *Proceedings of the London Mathematical Society*, Vol. 45, 1939, pages 161–228; reprinted in A.M. Turing, *Collected Works: Mathematical Logic*, pages 81–148.

[43] A. M. Turing, *Intelligent machinery.* National Physical Laboratory Report, 1948. In *Machine Intelligence 5* (Meltzer B. and Michie D., eds.), Edinburgh University Press, Edinburgh, 1969, pp. 3–23. Reprinted in A. M. Turing, *Collected Works: Mechanical Intelligence* (Ince D. C., ed.), North-Holland, Amsterdam, New York, Oxford, Tokyo, 1992.

[44] A. M. Turing, The chemical basis of morphogenesis, *Philosophical Transactions of the Royal Society of London*, **B 237** (1952), 37–72; reprinted in A.M. Turing, *Collected Works: Morphogenesis*.

[45] A. M. Turing, *Collected Works, Vol. 3: Morphogenesis* (P. T. Saunders, Editor), Elsevier, Amsterdam, New York, Oxford, Tokyo, 1992.

[46] A. M. Turing, *Collected Works: Mathematical Logic* (R. O. Gandy and C. E. M. Yates, Editors), Elsevier, Amsterdam, New York, Oxford, Tokyo, 2001.

[47] P. Woit, *Not Even Wrong: The Failure of String Theory and the Continuing Challenge to Unify the Laws of Physics,* Jonathan Cape, London, 2006.

How to build a hypercomputer

N. C. A. da Costa and F. A. Doria*

Advanced Studies Research Group and Fuzzy Sets Laboratory
PIT, Production Engineering Program
COPPE, UFRJ
P.O. Box 68507
21945–972 Rio RJ Brazil.

NCACOSTA@USP.BR
FADORIA@GMAIL.COM
FAMADORIA@PEP.UFRJ.BR

Version 2.0
July 9, 2008

Abstract

We claim that the *theoretical hypercomputation problem* has already been solved, and that what remains is an engineering problem. We review our construction of the Halting Function (the function that settles the Halting Problem) and then sketch possible blueprints for an actual hypercomputer.

1 Prologue

The authors have degrees in engineering. Engineers build things. So, the goal of this paper is to sketch a series of steps at whose conclusion we would have an actual, working hypercomputer.

Will it work? We leave that question unanswered. But, as we insist, we are cautiously optimistic that its answer may turn out to be a "yes."

We take our cue from the following remark by Scarpellini [15] :

*The authors are partially funded by CNPq–Brazil, Philosophy Section. They also acknowledge support from the Brazilian Academy of Philosophy.

> In this connection one may ask whether it is possible to construct an analog–computer which is in a position to generate functions $f(x)$ for which the predicate $\int f(x)\cos(nx)\,dx > 0$ is not decidable while the machine itself decides by direct measurement whether $\int f(x)\cos(nx)\,dx$ is greater than zero or not. Such a machine is naturally only of theoretical interest, since faultless measuring is assumed, which requires the (absolute) validity of classical electrodynamics and probably such technical possibilities as the existence of innitely thin perfectly conducting wires. All the same, the (theoretical) construction of such a machine would illustrate the possibility of non-recursive natural processes.

(Scarpellini's paper discusses among other results the decidability of such predicates.)

The hypercomputation problem

The hypercomputation problem splits into two questions:

1. *The theoretical hypercomputation problem.* Can we conceive a hypercomputer, given ideal operating conditions?

2. *The practical hypercomputation problem.* Given a positive answer to the preceding question, can we build a concrete, working, hypercomputer?

We argue here that the answer to the first question is a definite "yes," if we accept that ideal analog machines fit into the requirements ("ideal operating conditions" — just take a look at Scarpellini's papers [15, 16]). The second question boils down according to our viewpoint to an engineering problem; we may answer it with a "maybe," or "we have to see." Or, if we follow the classical injunction: build a prototype!

Then we will see how it performs; which engineering problems must be overcome in order to have a decently working hypercomputer.

Structure of the paper

The paper is divided into two major sections. The first major section explicitly constructs the Halting Function, that is, the function that settles Turing's Halting Problem. Of course that cannot be done within Peano Arithmetic, or with the help of partial recursive functions. But we can do it if we extend the language of arithmetic in adequate ways, for there are infinitely many explicit expressions for the Halting Function. (No extraordinary tools are required.)

The second major section sketches the construction of a hypercomputer in a series of steps. We discuss pros and cons; our conclusion is cautiously optimistic — we believe that the gizmo can be made, and that something like it will eventually be built in the near future.

Motivation

The ideas presented here were originally published in [3]. We were interested in the decision problem for chaotic systems, which appeared to us as a possible example of an undecidable question with practical implications.

A suggestion by P. Suppes [18] led us to Richardson's transforms [13], and out of those we obtained an explicit expression for the Halting Function (see the references; see also [4, 6] for a survey). It came as a surprise to us; we had long had the incorrect idea that none of the usual mathematical languages could explicitly contain an expression for the Halting Function, as it seemed to be something so deeply fundamental.

Then it was clear from start that we were dealing with a construction that had something to do with super–Turing machines [17]. However a more explicit presentation of our ideas on the matter of hypercomputation was only published in 1996 [5].

Style of the paper

Since most results in the present paper have already been discussed elsewhere, we have decided to argue in an informal, almost naïve way. We have also added some redundancy to our presentation.

Informed readers will notice that we avoid a detailed characterization of analog computers. (We simply supposed that the functions we use in the construction of an expression for the Halting Function are computable.) We are aware of the main theoretical contributions to the field, from Shannon to Pour–El. However our starting point is Scarpellini's remark quoted above: analog computers are usually seen as poor relatives of digital machines, as they can only compute — in the sense of analog computation — a relatively small family of functions. Yet Scarpellini surprised everybody by showing that ideal analog computers can decide undecidable predicates (with respect to some theory with a recursively enumerable set of theorems). So, they are definitely very powerful in an unique way, even if in an idealized domain.

2 Hypercomputation theory

(We stress that this is just an informal sketch.) If we define a hypercomputer to be a machine that can generate, or decide, all arithmetical truths, then very little is required to build a theory that fits the requirement, namely, Peano Arithmetic (PA) plus Shoenfield's version of the ω–rule is enough (for a nice presentation of Shoenfield's rule see [10]).

PA plus Shoenfield's rule proves all true arithmetic sentences, that is, those that hold of the standard model. The hypercomputation theory we sketched in [5] does so. The idea is: we start from Turing machine theory. We then add to it an expression for the function that settles the Halting Problem, and postulate it to be (hyper)computable. The jump [14] allows us to decide arithmetic degrees beyond $\mathbf{0}$ [4, 5]. Given a solution for the Halting Problem, and given the jump, we can decide any arithmetical sentence in a finite number of steps.

We have also mentioned that a "geometrical principle" implies hypercomputation [5]:

> *We can always decide whether two smooth curves in the interior of a rectangle in the plane intersect.*

That "geometrical principle" has an advantage and a disadvantage. The advantage: it is easy to understand — actually a stricter version of it is enough, but the general version stated above is more synthetic. The disadvantage: we must show how to translate that principle, which deals with smooth, continuous geometrical objects, into the discrete objects that are in the domain of arithmetic. The trick is done through Richardson's transforms, soon to be introduced.

So we take:

Definition 2.1 *A hypercomputer is a theoretical device that settles at least the Halting Problem.* □

Follows by our constructions (see the references) that such a device can be extended to another device that settles all arithmetical truths along the standard model.

3 From Richardson's transforms to the Halting Function

The present section is based on [1]; for the proofs see [13]. Our presentation splits into several topics:

- Formalized arithmetic and Turing machines.
- Richardson's maps.
- The Halting Function in formal languages that extend arithmetic.

We refer to [12] for notation and requirements from logic. We use: $\neg$, "not," $\vee$, "or," $\wedge$, "and," $\rightarrow$, "if... then...," $\leftrightarrow$, "if and only if," $\exists x$, "there is a x," $\forall x$, "for every x." $P(x)$ is a formula with x free; it roughly means "x has property P." Finally $T \vdash \xi$ means T proves ξ, or ξ is a theorem of T. ω is the set of natural numbers, $\omega = \{0, 1, 2, \ldots\}$.

Algorithmic functions are given by their programs coded in Gödel numbers e [14]. We will sometimes use Turing machines (noted by sans–serif letters with the Gödel number as index M_e) or partial recursive functions, noted $\{e\}$.

We start from a very simple theory of arithmetic, noted $A1$. Its language includes variables $x, y, \ldots$, two constants, $\mathbf{0}$ and $\mathbf{1}$, the equality sign $=$, and two operation signs, $+, \times$. Basically $A1$ has axioms for the operations $+$ and $\times$, the behavior of constants $\mathbf{0}$ and $\mathbf{1}$, and the trichotomy axiom, that is, given two natural numbers x and y, either $x < y$ or $x = y$ or $x > y$. $A1$ contains no induction axiom.

The *standard interpretation* for $A1$ is: the variables $x, y, \ldots$ range over the natural numbers, and $\mathbf{0}$ and $\mathbf{1}$ are seen as, respectively, zero and one. The only requirement we impose on $A1$ is: that theory should be strong enough to formally include all of Turing machine theory. Recall that a Turing machine is given by its Gödel number, which recursively codes the machine's program. Rigorously, for $A1$, we must have:

Definition 3.1 *A Turing machine of Gödel number e operating on x with output y, $\{e\}(x) = y$ is* **representable** *in theory $A1$ if there is a formula $F_e(x, y)$ in the language of $A1$ so that:*

1. *$A1 \vdash \{F_e(x, y) \wedge F_e(x, z) \rightarrow [y = z]\}$, and*
2. *For natural numbers a, b, if $\{e\}(a) = b$, then $A1 \vdash F_e(a, b)$.* $\Box$

Then we have the representation theorem for partial recursive functions in $A1$:

Proposition 3.2 *Every Turing machine is representable in $A1$. Moreover there is an effective procedure that allows us to obtain F_e from the Gödel number e.* $\Box$

We restrict here our interest to theories that are *arithmetically sound*, that is, which have a model with standard arithmetic for its arithmetical segment.

Richardson's map

We now describe the Richardson transforms [6, 13]. We start from a strengthening of Proposition 3.2:

Proposition 3.3 *If $\{e\}(a) = b$, for natural numbers a, b, then we can algorithmically construct a polynomial p_e over the natural numbers so that $[\{e\}(a) = b] \leftrightarrow [\exists x_1, x_2, \ldots, x_k \in \omega\, p_e(a, b, x_1, x_2, \ldots, x_k) = 0]$.* $\square$

Follows:

Proposition 3.4 *$a \in R_e$, where R_e is a recursively enumerable set, if and only if there are e and p so that $\exists x_1, x_2, \ldots, x_k \in \omega\, [p_e(a, x_1, x_2, \ldots, x_k) = 0]$.* $\square$

Richardson's map [6, 13] allows us to obtain in an algorithmic way, given any such $p_e(a, \ldots)$, a real–defined and real–valued function $f_e(a, \ldots)$ that has roots if and only if $p_e(a, \ldots)$ has roots as a Diophantine equation.

Richardson's map: multidimensional version

We can be more specific: let $\mathcal{A}$ be the algebra of subelementary functions (polynomials over the reals, sines, cosines; everything closed under $+, -$, products by real numbers and by the functions that generate the algebra, to which we add function composition). Let R denote the real line.

(We do not require the exponential function in our constructions.)

We now state the first of Richardson's main results: given that $A1 \subset$ ZFC, and if $\mathcal{P}$ is the set of all finite–lenght polynomials over ω:

Proposition 3.5 (Richardson's Map, I) *There is an injection $\kappa_{\mathcal{P}} : \mathcal{P} \to \mathcal{A}$, where $\mathcal{P}$ denotes the algebra of ω–valued polynomials in a finite number of variables, and $\mathcal{A}$ is the algebra of subelementary functions described above, such that:*

1. *$\kappa_{\mathcal{P}}$ is constructive, that is, given the expression for p in $A1$, there is an effective procedure so that we can obtain the corresponding expression for $F = \kappa_{\mathcal{P}}(p)$ in* ZFC.

2. *$\kappa_{\mathcal{P}}$ is 1–1.*

3. *For $\mathbf{x} = (x_1, \ldots, x_n)$, $\exists \mathbf{x} \in \omega^n \; p(m, \mathbf{x}) = 0$ if and only if $\exists \mathbf{x} \in \mathsf{R}^n \; F(m, \mathbf{x}) = 0$ if and only if $\exists \mathbf{x} \in \mathsf{R}^n \; F(m, \mathbf{x}) \leq 1$, for $p \in \mathcal{P}$ and $F \in \mathcal{A}$.*

4. *The injection $\kappa_{\mathcal{P}}$ is proper.* □

The crucial property is given in step 3.: it allows us to translate the existence of roots for Diophantine equations into roots of the corresponding transformed real–defined and real–valued function, with some extras.

Next step gives us a 1–dimensional version of Richardson's map.

Richardson's map: one–dimensional version

Corollary 3.6 (Richardson's Map, II) *Let $\mathcal{A}_1$ be the algebra of subelementary functions over a single real variable x. Then there is a map $\kappa' : \mathcal{P} \to \mathcal{A}_1$ such that:*

1. *κ' is constructive.*

2. *κ' is 1–1.*

3. *The inclusion $\kappa'(\mathcal{P}) \subset \mathcal{A}_1$ is proper.*

4. *$\exists \mathbf{x} \in \omega^n \; p(m, \mathbf{x}) = 0$ if and only if $\exists x \in \mathsf{R} \; L(m, x) = 0$ if and only if $\exists x \in \mathsf{R} \; G(m, x) \leq 1$, for adequate L, G, whose expressions can be explicitly exhibited.* □

The Halting Function

The main result in Alan Turing's remarkable 1937 paper, "On computable numbers, with an application to the Entscheidungsproblem" [19], is a proof of the algorithmic unsolvability of a version of the halting problem: given an arbitrary Turing machine of Gödel number e, for input x, there is no algorithm that decides whether $\{e\}(x)$ stops and outputs something, or enters an infinite loop.

Remark 3.7 Let $\mathsf{M}_m(a) \downarrow$ mean: "Turing machine of Gödel number m stops over input a and gives some output." Similarly $\mathsf{M}_m(a) \uparrow$ means, "Turing machine of Gödel number m enters an infinite loop over input a." Then we can define the halting function θ:

- $\theta(m, a) = 1$ if and only if $\mathsf{M}_m(a) \downarrow$.
- $\theta(m, a) = 0$ if and only if $\mathsf{M}_m(a) \uparrow$.

$\theta(m, a)$ is the halting function for M_m over input a. $\square$

θ isn't algorithmic, of course [14, 19], that is, there is no Turing machine that computes it.

Remark 3.8 As we now show, we can explicitly write an expression for a function in the language of classical analysis that settles the halting problem. We proceed as follows:

- Given Turing machine $\mathsf{M}_m(a) = b$, for natural numbers a, b, we can algorithmically obtain a polynomial $p_m(\langle a, b\rangle, x_1, \ldots, x_k)$ so that:

$$\mathsf{M}_m(a) = b \leftrightarrow \exists x_1, \ldots, x_2 \in \omega \, [p_m(\langle a, b\rangle, x_1, \ldots, x_k) = 0].$$

- Given F_m, real–defined and real–valued, we have that:

$$\exists x_1, \ldots, x_2 \in \omega \, [p_m(\langle a, b\rangle, x_1, \ldots, x_k) = 0] \leftrightarrow$$

$$\leftrightarrow \exists x_1, \ldots, x_k \in \mathsf{R} \, F_m(\langle a, b\rangle, x_1, \ldots, x_k) \leq 1.$$

and

$$\forall x_1, \ldots, x_2 \in \omega \, [p_m(\langle a, b\rangle, x_1, \ldots, x_k) \neq 0] \leftrightarrow$$

$$\leftrightarrow \forall x_1, \ldots, x_k \in \mathsf{R} \, F_m(\langle a, b\rangle, x_1, \ldots, x_k) > 1.$$

- That is to say: $\mathsf{M}_m(a) \downarrow$ if and only if $F_m(a, \ldots)$ goes below 1, and $\mathsf{M}_m(a) \uparrow$ if and only if $F_m(a, \ldots)$ stays above 1.

  This is the property we use in order to construct the halting function θ_m. $\square$

We now need the concept of an *universal Diophantine polynomial.* Martin Davis [7] describes an algorithmic procedure out of which, given a Turing machine with input a $\mathsf{M}_m(a)$, we obtain a polynomial $p_m(a, x_1, \ldots)$ so that it has roots if and only if $\mathsf{M}_m(a)$ converges (outputs some result). Now, if $\mathsf{U}(m, a)$ is an universal Turing machine [14, 19], we can similarly obtain a polynomial $p(\langle m, a\rangle, \ldots)$ which stands for $p_m(a, \ldots)$.

More precisely, if $[\exists x_1, \ldots, x_k \, p_m(\langle a, b\rangle, x_1, \ldots, x_k) = 0] \leftrightarrow [\mathsf{M}_m(a) = b]$, then, for the universal polynomial $p(\langle m, a, b\rangle, \ldots)$:

$$[\exists x_1, \ldots, x_r \, p(\langle m, a, b\rangle, x_1, \ldots, x_r) = 0] \leftrightarrow$$
$$\leftrightarrow [\exists x_1, \ldots, x_k \, p_m(\langle a, b\rangle, x_1, \ldots, x_k) = 0].$$

From the preceding considerations, if σ is the sign function, $\sigma(\pm x) = \pm 1$ and $\sigma(0) = 0$:

Proposition 3.9 (The Halting Function.) *The Halting Function $\theta(n, q)$ is explicitly given by:*

$$\theta(n, q) = \sigma(G_{n,q}),$$
$$G_{n,q} = \int_{-\infty}^{+\infty} C_{n,q}(x) e^{-x^2} dx,$$
$$C_{m,q}(x) = |F_{m,q}(x) - 1| - (F_{m,q}(x) - 1).$$
$$F_{n,q}(x) = \kappa_P p_{n,q}. \ \square$$

Here $p_{n,q}$ is the two–parameter universal Diophantine polynomial

$$p(\langle n, q\rangle, x_1, x_2, \ldots, x_r)$$

and κ_P is as in Proposition 3.5.

There are infinitely many alternative explicit expressions for the halting function θ [6].

Remark 3.10 We do not require Richardson's transform to obtain an expression for the Halting Function. There is also an expression for the Halting Function even within a simple extension of $A1$. Let $p(n, \mathbf{x})$ be a 1–parameter universal polynomial; $\mathbf{x}$ abbreviates $x_1, \ldots, x_p$. Then either $p^2(n, \mathbf{x}) \geq 1$, for all $\mathbf{x} \in \omega^p$, or there are $\mathbf{x}$ in ω^p such that $p^2(n, \mathbf{x}) = 0$ sometimes. As $\sigma(x)$ when restricted to ω is primitive recursive, we may define a function $\psi(n, \mathbf{x}) = 1 - \sigma p^2(n, \mathbf{x})$ such that:

- Either for all $\mathbf{x} \in \omega^p$, $\psi(n, \mathbf{x}) = 0$;
- Or there are $\mathbf{x} \in \omega^p$ so that $\psi(n, \mathbf{x}) = 1$ sometimes.

Thus the halting function can be represented as:

$$\theta(n) = \sigma[\sum_{\tau^q(\mathbf{x})} \frac{\psi(n, \mathbf{x})}{\tau^q(\mathbf{x})!}],$$

where $\tau^q(\mathbf{x})$ denotes the positive integer given out of $\mathbf{x}$ by the pairing function τ: if τ^q maps q–tuples of positive integers onto single positive integers, $\tau^{q+1} = \tau(x, \tau^q(\mathbf{x}))$. Recall that the infinite sum can be given by a simple iterative definition. $\square$

4 How to build a hypercomputer

There are two ways we can attack the question. The first one is based on the previous construction of the Halting Function. The second one is based on an alternative, well established technique.

First notice the following:

> *The Halting Problem is: given an arbitrary Turing machine $\{e\}$ and an arbitrary input n, can we check whether $\{e\}(n)$ stops?*

Now pick up some e, n and keep it fixed. As it will turn out, there is an algorithmic procedure (sketched below) that allows us to check whether $\{e\}(n)$ stops or not. (There is a price to pay; it may be of high complexity, but we can leave that difficulty aside for the moment). Difficulties appear when we go from this particular instance to a construction that encompasses all instances, that is, when we add the universal quantifier.

We will now consider two possibilities for a hypercomputer.

Our proposal for a hypercomputer

The present construction is based on the following result by Richardson [2, 13, 15, 16]:

Remark 4.1

- Given Turing machine $\mathsf{M}_m(a) = b$, for natural numbers a, b, we can algorithmically obtain a polynomial $p_m(\langle a, b\rangle, x_1, \ldots, x_k)$ so that:

$$\mathsf{M}_m(a) = b \leftrightarrow \exists x_1, \ldots, x_2 \in \omega\, [p_m(\langle a, b\rangle, x_1, \ldots, x_k) = 0].$$

- Recall that we can construct an expression for a F_m, real–defined and real–valued, so that:

$$\exists x_1, \ldots, x_2 \in \omega\, [p_m(\langle a, b\rangle, x_1, \ldots, x_k) = 0] \leftrightarrow$$

$$\leftrightarrow \exists x_1, \ldots, x_k \in \mathsf{R}\, F_m(\langle a, b\rangle, x_1, \ldots, x_k) \leq 1.$$

  and

$$\forall x_1, \ldots, x_2 \in \omega\, [p_m(\langle a, b\rangle, x_1, \ldots, x_k) \neq 0] \leftrightarrow$$

$$\leftrightarrow \forall x_1, \ldots, x_k \in \mathsf{R}\, F_m(\langle a, b\rangle, x_1, \ldots, x_k) > 1.$$

- In a more figurative way: $\mathsf{M}_m(a) \downarrow$ (converges) if and only if $F_m(a, \ldots)$ dives below 1, and $\mathsf{M}_m(a) \uparrow$ (diverges, enters an infinite loop) if and only if $F_m(a, \ldots)$ stays above 1.

(See Remark 3.8.) So, machine e stops over input n if Richardson's transform $F_{(e,n)}$ of the corresponding Diophantine polynomial $p(\langle e, n\rangle, \ldots)$ dives below 0; otherwise it stays above 1. Of course we can modify the corresponding Richardson transform in such a way that the "signaling gap" from 0 to 1 is widened to an arbitrary k.

Now consider the following analog machine:

- First compactify F by some simple elementary–function map, over the semi–open interval $[0, K)$, that is to say, map $[0, \infty)$ over $[0, K)$ and carry F over. We will note F' the compactified version of function F.
- Given each pair (e, n), $F'_{(e,n)}$ is built out of elementary functions plus π.
- So there is an *ideal* analog computer that computes it.
- Notice that the values (e, n) can be easily coded as parameters in our analog machine.
- So, build it.

By the preceding considerations, it is our belief that the *theoretical* hypercomputation problem has already been settled. The *realistic* hypercomputation problem is another matter, but is a problem that belongs to the realm of engineering.

We see three main difficulties (there may be others) in the actual construction of a hypercomputer if we follow the preceding steps:

- *The construction of* π*.* How are we going to insert π as a parameter in our machine? We have considered the possibility that π should be derived out of a geometrical, graphical construction.
- *Overshoot problems.* This is another difficulty to be considered. The compactification procedure may lead to irregular, out–of–control behavior when one goes to the (compactified) infinite point.
- *The size of* e *and* n*.* For actual computers e and n may be very large, and coding them into an analog computer may be quite difficult.

We do not consider the usual objection that says, the world is ruled by quantum mechanics, and is therefore discrete, so any analog, continuous construction will end up by being impossible. We suggest reading on that a brilliant, half–forgotten text, Havemann's essay of the discrete and the continuous [11].

Turing–Fefermann hypercomputer

The main reason why the authors have always believed that someday someone will build a hypercomputer stems from the following fact: given one instance, or a finite but arbitrary number of instances of the Halting Problem, we can always algorithmically decide them. Here it goes why it is so.

- Each instance of the Halting Problem (if it fails to stop over an input) is arithmetically equivalent to a Π_1 sentence,

  $$\forall x_1, x_2, \ldots, p(\langle m, n\rangle x_1, x_2, \ldots) \neq 0.$$

  (Here $\{m\}$ is the machine, and n is the input we wish to test.)

- Then we know [9, 10, 20] that there will be some theory T_k in the Turing–Fefermann hierarchy, $T_0 = \mathrm{PA}$, $T_1 = \mathrm{PA} + \mathrm{Consis}\,\mathrm{PA}, \ldots$, that will prove such a Π_1 sentence, if it is true. On the other hand, we can recursively test for its negation, $\exists x_1, x_2, \ldots, p(\ldots) = 0$. Therefore we can build a recursive procedure in order to decide whether $\{m\}(n)$ stops or not.

- We thus generate two listings of results.

- List A: we test $p(\langle m, n\rangle, x_1, x_2, \ldots)$ for each n–tuple $x_1, x_2, \ldots$ in order to check whether some n–tuple is such that $p(\langle m, n\rangle, \ldots) = 0$.

  If it does, the procedure stops, and we learn that $\{m\}(n)$ stops.

- List B: we go by dovetailing, and generate all theorems for T_0, T_1, T_2, in the Turing–Fefermann hierarchy.

  If $\forall x_1, \ldots\ p(\langle m, n\rangle, x_1, \ldots) \neq 0$ appears among the theorems in that listing, we stop the procedure.

There is a catch: procedure B is extremely costly in computational terms (procedure A may also be very costly). However we may use — we expect — our analog machine to speed it up. Anyway, the point is: finite *arbitrary* instances of the Halting Problem are always algorithmically decidable. (Franzen's objections [10] do not apply precisely because of that: we only have to check a finite number of instances of the Halting Problem, in actual situations.)

Remark 4.2 An important point: we cannot formally "put together" all such procedures that settle particular instances of the Halting Problem and obtain a theory *with a recursively enumerable set of theorems* that settles *all* instances of the Halting Problem, as that theory would then violate the Gödel incompleteness theorems. In fact, it can be shown that such a theory implies Shoenfield's ω–rule, and PA plus Shoenfield's rule is a theory with a nonrecursively enumerable set of theorems.

More precisely: *no extension of arithmetic with a recursively enumerable set of theorems can prove that the algorithm sketched above delivers what it promises.* □

5 Conclusion

We rest our case. To stress the point:

- We believe that the theoretical hypercomputation problem is solved. We can — ideally — build an analog machine that settles arbitrary instances of the Halting Problem.

- The actual construction of such a machine isn't a mathematical problem anymore; it is an engineering problem. One has to build a prototype and see which problems will eventually come up.

If one settles arbitrary instances of the Halting Problem, we may now conceive a machine that decides sentences along the arithmetical hierarchy [4, 6]. The implications are obvious, and we will not comment on them.

The proof of the pudding is on the eating: next step is to build a prototype.

We thus rest our case.

6 Acknowledgments

The authors wish to thank Professors C. Calude and J. F. Costa who kindly invited them to present this paper at the Vienna workshop on Unusual Models of Computation, August 2008. They also wish to thank two anonymous referees for their comments and criticisms.

The ongoing research program that led to this text has been sponsored by the Advanced Studies Group, Production Engineering Program, COPPE–UFRJ, Rio, Brazil. The authors also wish to thank the Institute for Advanced Studies at the University of São Paulo for support of this research project; we wish to acknowledge support from the Brazilian Academy of Philosophy and its chairman Professor J. R. Moderno. Finally FAD wishes to thank Professors R. Bartholo, C. A. Cosenza and S. Fuchs for their invitation to join the Fuzzy Sets Lab at COPPE–UFRJ and the Philosophy of Science Program at the same institution.

The authors acknowledge partial support from CNPq, Philosophy Section.

References

[1] R. Bartholo, C. A. Cosenza, F. A. Doria, C. Lessa, "Can economic systems be seen as computing machines?" *J. Economic Behavior and Organization*, to appear (2009).

[2] B. J. Copeland and R. Sylvan, "Beyond the universal Turing machine," *Austral. J. Philosophy* **77**, 46–67 (1999).

[3] N. C. A. da Costa and F. A. Doria, "Undecidability and incompleteness in classical mechanics," *Int. J. Theor. Phys.* **30**, 1041–1073 (1991).

[4] N. C. A. da Costa and F. A. Doria, "Suppes predicates and the construction of unsolvable Problems in the axiomatized sciences," in P. Humphreys, ed., *Patrick Suppes, Scientific Philosopher*, II, 151–191 Kluwer (1994).

[5] N. C. A. da Costa and F. A. Doria, "Variations on an original theme," in J. Casti and A. Karlqvist, *Boundaries and Barriers*, Addison–Wesley (1996).

[6] N. C. A. da Costa and F. A. Doria, "Computing the future," in K. Vela Velupillai, ed., *Computability, Complexity and Constructivity in Economic Analysis*, Blackwell (2005).

[7] M. Davis, "Hilbert's Tenth Problem is unsolvable," *Amer. Math. Monthly* **80**, 233 (1973).

[8] F. A. Doria, "Informal vs. formal mathematics," *Synthèse* **154**, 401–415 (2007).

[9] S. Feferman, "Transfinite recursive progressions of axiomatic theories," *J. Symbolic Logic* **27**, 259 (1962).

[10] T. Franzen, "Transfinite progressions: a second look at completeness," *Bull. Symbolic Logic* **10**, 367-389 (2004).

[11] R. Havemann, *Dialektik ohne Dogma ?*, Rowohlt (1964).

[12] E. Mendelson, *Introduction to Mathematical Logic*, 4th ed., Chapman & Hall (1997).

[13] D. Richardson, “Some undecidable problems involving elementary functions of a real variable,” *J. Symbol. Logic* **33**, 514 (1968).

[14] H. Rogers Jr., *Theory of Recursive Functions and Effective Computability*, John Wiley (1967).

[15] B. Scarpellini, “Two undecidable problems of analysis,” *Minds and Machines* **13**, 49–77 (2003).

[16] B. Scarpellini, “Comments to ‘Two undecidable problems of analysis,”’ *Minds and Machines* **13**, 79–85 (2003).

[17] I. Stewart, “Deciding the undecidable,” *Nature* **352**, 664–665 (1991).

[18] P. Suppes, personal communication (1990).

[19] A. M. Turing, “On computable numbers, with an application to the Entscheidungsproblem,” *Proc. London Math. Society* **50**, 230 (1937).

[20] A. M. Turing, “Systems of logic based on ordinals,” *Proc. London Math. Society*, Ser. 2 **45**, 161 (1939).

What is a universal computing machine?

Jean-Charles Delvenne

Université Catholique de Louvain,
Department of Mathematical Engineering,
Avenue Georges Lemaître 4, B-1348 Louvain-la-Neuve, Belgium

Abstract

A computer is classically formalised as a universal Turing machine or a similar device. However over the years a lot of research has focused on the computational properties of dynamical systems other than Turing machines, such cellular automata, artificial neural networks, mirrors systems, etc.

In this paper we propose a unifying formalism derived from a generalisation of Turing's arguments. Then we review some of universal systems proposed in the literature and show that are particular case of this formalism. Finally, we review some of the attempts to understand the relation between dynamical and computational properties of a system.

Key words: Turing universality, dynamical systems

1 Introduction

We are interested in *computing machines*, which we informally define as machines able to solve decision problems on integers (or finite objects that can be encoded as integers), such as, for instance, primality. We are especially interested in *universal* computing machines, i.e., those who have the same power as a universal Turing machine.

Note that in this article we are only interested in solving decision problems on *integers*, while computable analysis deals with computable functions and decision problems on the reals (e.g., checking invertibility of a real-valued matrix). See for instance [29,3,25,22] on computable analysis.

Email address: jean-charles.delvenne@uclouvain.be (Jean-Charles Delvenne).

Preprint submitted to Elsevier

Also, we do not consider hypercomputation and systems with super-Turing capabilities, as can be found for instance in [27,4,9].

Note that here we do not define universality as the ability to 'simulate any other system'. See for instance [24] for such a notion of universality in the case of cellular automata.

We start by a quick review of computing machines, such as Turing machines, counter machines, cellular automata. We then observe, with Davis's definition of universality, that a computing machine is always a dynamical system together with an r.e.-complete halting problem. We then ask which problems can be considered as reasonable halting problems for a given dynamical system.

We then generalise Turing's famous argument to the case where a dynamical system, instead of pencil and paper, is available to a human operator. This leads to a general definition of universal computing machine. Then several definitions of universal systems are reviewed and found to be particular cases of this framework.

We finally review some results about the interaction between the computational and dynamical properties of a computing machine.

2 Turing machines

In the beginning of the twentieth century, the question arose, as a consequence of the search for foundations for mathematics, of a mechanical procedure to solve a mathematical problem, or *algorithm*. Several answers, later proved mathematically equivalent, were provided in the thirties and forties, by Post, Church, Kleene, Turing. For the original papers, see [5].

Among those answers, Turing's is perhaps the most thoroughly argued as a model for computation. Here we sketch Turing's argument to construct his machine, as it will be a basis to our definition of computing machine.

In this article, we only consider decision problems, which was not the point of view originally taken by Turing. This is not a loss of generality as computing an integer can reduced to a sequence of decision problems.

The algorithm is performed by a human operator, who applies a series of instructions on the initial data. Intermediate results are written on paper.

Turing essentially argues that the mind of the operator can only be in finitely many states, can distinguish only finitely many symbols on paper, and can write only finitely many different symbols. Hence the operator is essentially a finite-state automaton (as far as the execution of the algorithm is concerned). The sheet of paper can be similarly considered as an unlimited linear tape divided into cells. Every cell contains a symbol out of a finite alphabet. The operator can read or write the symbol in a cell, and translate the tape one cell to the left or right. The initial data is written on finitely many cells on the tape, while the rest of the tape filled with the blank symbol. The computation ends when the operator enters the state of mind 'The computation is finished', which we call the *halting state*. As a result, a human operator executing an algorithm is modeled by a one-tape Turing machine as we know them. For a decision problem, we can also assume that the halting state of mind contains the answer to the problem, e.g., 'The computation is finished and the answer is Yes.' In this case there are two halting states. Note that a computation on a Turing machine can result in three outcomes: 'Yes', 'No' or no answer at all (i.e., the machine does not halt).

Turing confirms that his definition is sensible by showing that slightly different models, such as two-tape Turing machines, have the same power as Turing machines. That means that for every two-tape Turing machine one can construct in an effective way a one-tape Turing machine the solves the same problem, and conversely. Here by 'effective' we mean 'intuitively computable'.

He then finds that there exists a universal Turing machine, i.e., a Turing machine such that every pair (Turing machine, initial data) can be converted in an effective way into an initial data for the universal machine so as to preserve the outcome 'Yes', 'No' or 'Does not halt'.

He then proceeds to show that the *halting problem* is undecidable, more precisely r.e.-complete, as will be said later. The halting problem is the following: Given an initial data for a fixed universal Turing machine, does the universal Turing machine reach a halting state? Equivalently: Given a Turing machine and an initial data for this Turing machine, does the Turing machine reach a halting state?

3 Other universal machines

Other kinds of machines were subsequently devised to formalise computation, such as, for instance, counter machines (or register machines, or Minsky machines); see, e.g., [19]. A k-counter machine is made of k cells, each of which contains a natural integer. At every step, a finite automaton can test if the content of a counter is zero, increment a counter or decrement a counter. Again, the initial data is encoded in the content of the counters, and the computation is considered as finished when we reach a 'halting state'.

It has been proved that there exists a *universal* counter machine U.

'Universal' can be defined in terms of a reduction from the problem solved by a universal Turing machine. This means there is an effective way to encode any initial data for a fixed universal Turing machine into an initial data for the universal counter machine so as to preserve the outcome of the computation ('Yes', 'No', 'Does not halt'). Note once again that we avoid here talking about universality as 'dynamical simulation' of other dynamical systems, which avoids the need to introduce definitions of simulation.

Hence the halting problem for a universal counter machine (i.e., determining if a given initial content of the counters and an initial state of the head will reach a halting state of the head) is r.e.-complete as well.

Other similar machines were defined: Post machines, tag machines (also invented by Post) for instance; see [18]. For those again, a way to use it for computation is defined, and a universal machine is found.

All the machines above are machines with countably many states: the state of a Turing machine, for instance, is a finite sequence of symbols plus the state of the head for a Turing machine. All those machines are dynamical systems. For the moment, we loosely define a dynamical system as an object evolving in time, and completely characterised at any time by its state.

But most dynamical systems studied in mathematics and physics have an uncountable state space, e.g., cellular automata, differential equations, piecewise linear maps, etc. Examples of those systems have been proved universal. Their halting problem is imitated from the Turing machine in the following way. We choose a particular countable family of initial states, and countable family of final states, or final sets of states. Then the halting problem is given an initial state and a final state/set of states, whether the trajectory starting from the initial state will reach the final state/set of states. More specific examples are given in Section 7.

In that case, finding the relevant halting problem becomes not obvious at all, since there are many ways to select a countable family of initial states out of uncountably many. For instance, the cellular automaton of rule 110 is universal for the eventually periodic states (i.e., periodic sequences of symbols up to finitely many) but not for periodic sequences or for finite states (i.e., where all but finitely many symbols are equal to zero).

As observed in [10], some trivial dynamical systems can be also considered universal with an artificial halting problem. For instance, take the full shift on $\{0,1,2\}^{\mathbb{N}}$ is universal with the family of initial states $1^n0^ta^\infty$, where t is the halting time of a universal Turing machine on data n. If the machine does not halt on n, then the initial state is 1^n0^∞. If the machine halts on

‘Yes’, then $a = 1$, if it halts on ‘No’, then $a = 2$. Note that those states are computable: we can compute the bit of any rank for any state. Then the halting problem whether we reach the state 1^∞ from the initial state encodes the halting problem of the Turing machine. Therefore we are bound to conclude that the full shift is a universal computing machine! But it is only so with respect to a certain cooked up halting problem. It is also clear that unreasonable choices of initial conditions (i.e., undecidable) will make even simpler systems, such as the identity, even more powerful than Turing machines and shows that choosing a relevant halting problem requires caution.

As a conclusion, to every universal computing machine, is associated a certain r.e.-complete halting problem.

4 Davis universality

Davis [6], turning things around, proposed an astute definition of universality for a counter machine, Turing machine or any similar kind of object. A machine is said to be *universal* if and only if its halting problem is r.e.-complete. This definition essentially coincides with the former, but it bypasses the the mention of a universal Turing machine and the need for an effective encoding. Instead, the coding is implicit in the r.e.-completeness of the halting problem. Indeed, an r.e.-complete problem is one to which the halting problem of a universal Turing machine, and any other r.e. problem, can be reduced.

Hence a particular dynamical system is said to be universal *with respect to a certain problem*, called the halting problem, when this problem is r.e.-complete. In other terms, a computing machine is composed of a dynamical system together with a halting problem. As seen above, the choice of the natural halting problem for a dynamical system is sometimes obvious, by imitation from known examples, and sometimes more delicate.

Davis’s definition makes the quest for universal computing machines a particular case of the quest for undecidable mathematical problems. It happens in mathematics that a problem occurs, that one would like to solve, but turns out to be undecidable. For instance, whether a given polynomial in several variables with integral coefficients has an integral zero is r.e.-complete (Hilbert’s tenth problem, solved by Matyasevitch in 1970 [17]); whether a finitely presented group is the trivial group is r.e.-complete as well [26]. Those problems have been raised for their mathematical interest, not in order to define a new kind of computing machine. It seems difficult indeed to interpret them as the halting problem of some dynamical system.

As a conclusion, we have to solve the double question:

- Given a dynamical system, what is a relevant halting problem for it?
- What r.e.-complete problems can be considered as the halting problem of some computing machine?

5 Turing's argument revisited

In this section, we propose a recipe to address the two questions just above. We adapt Turing's argument to get a fine understanding of the interaction between dynamics and computation, and select a relevant halting problem.

Like in Turing's original argument, a human wants to solve decision problem. However, this time she has no paper or pencil, but a physical system. She doesn't necessarily know about the initial state of the system. During the process of computation, she can observe and act on the system. Like Turing, we assume that the human operator's mind can be in finitely many different states. Hence, we assume that the human can be modelled by a finite automaton, with finitely many actions on the system and finitely many possible observations from the system. This finite automaton acts as a controller on the system, in a feedback loop. Finite automaton accepting inputs (here, observations) and producing outputs (here, actuation) are also called Mealy automata, or transducers. See Fig. 1. We use the term 'state' for both the dynamical system and the controller, which models the mind. This justified as the controller is itself a dynamical system. By connecting the controller with the dynamical system, we get a new closed-loop dynamical system.

Hence a computing machine is defined in the following way.

Metadefinition 1 *A* computing machine *is defined by a dynamical system along with a countable family of controllers. Every controller is a finite state automaton with initial states and final states.*

We define the halting problem *as the following. Given a controller, the dynamical system is initialised to an arbitrary state and the controller is initialised to one of the initial states; is there a trajectory of the closed-loop system where the controller starts from an initial state and eventually reaches a final state?*

Some remarks must be made at this point.

The name 'halting problem' does not imply that the dynamical system stops after we have reached a final (or 'halting') state of the automaton. It just stops to be interesting, since we have answered the instance of the problem we wanted to solve.

Contrarily to most examples of universal systems found in the literature, there is no explicit reference to an 'initial state'. This is because a particular initial state x can be encoded in the set of actions: 'Set the state to x'. Or it can be encoded in the set of observations, in which case the computation starts with the following instruction: 'Observe the state; if it is not x then enter an infinite loop (i.e., never reach a final state)', which ensures that only computations starting by x will be processed.

We therefore see that the need for specifying an initial state is not as fundamental as it appears.

Since the controllers are finite state automata, it means that for a given controller, finitely many observations can be made on the system, and finitely many actions can be performed on it. As there are countably many controllers, only countably many possible observations and actions are to be considered for the system.

We speak of metadefinition rather than definition, because we still have to specify what is a dynamical system, what kind of observations and actuations are allowed on them, and how to interconnect the dynamical system with the controller. The answer to all these questions depends on our 'model of the world'. For instance, if we want to model physics by deterministic discrete-time systems, we'll consider a system of this kind. If we believe physical quantities cannot be observed with infinite precision, then we cannot allow the controller to observe if the system is a state x. And so on. Let us review some possibilities.

6 Dynamical systems

A dynamical system is intuitively anything that evolves in time. Many classes of dynamical systems exist, some of which we quickly describe in this section.

The most typical class is deterministic, discrete-time systems, given by an evolution map $f : X \to X$, where X is the *state space*. A state x is transformed into $f(x)$, then $f(f(x))$, and so on.

Examples include Turing machines, cellular automata, subshifts, piecewise affine maps, piecewise polynomial maps and neural networks.

Open dynamical systems (or input/output dynamical systems) allow for actuation. For instance, x is sent to $f(x, u)$, where u is the input (or actuation).

An observation on a dynamical system is most often a map $y = g(x)$, where x is the state of the system. As we want finitely many values for y, an observation is a partition of the state space of the system into finitely many sets. In principle, we could also consider a nondeterministic relation between x and y (for instance, to model an uncertain observation), but this seems to be unexplored in the literature of computational universality.

We may also consider a nondeterministic system; for instance the state x is sent into a ball of radius ϵ around $f(x)$. This is used to model perturbations that we know are bounded by ϵ.

Continuous-time systems are usually defined by a differential equation $\dot{x} = f(x)$ on (a part of) $\mathbb{R}^n$, or $\dot{x} = f(x, u)$, where $u(t)$ is the input function. In that case, the closed-loop system is a hybrid system: a mix of continuous and discrete dynamics.

Here we do not consider quantum universal systems; see for instance [8].

7 Reachability problems

Most definitions of universality rely on a reachability problem. The *reachability problem* for a discrete-time deterministic system $f : X \to X$ goes as follows: we are given two points x and y ('point-to-point reachability') or a point x and a set Y ('point-to-set'), and the question is whether there is a t such that $f^t(x) = y$ or $f^t(x) \in Y$.

A reachability problem is modelled according to Metadefinition 1 as follows. The controller first stets the inital condition to x; then it lets the system evolves according to f; when the state of the system is y or belongs to Y, the controller jumps to its final state.

Of course, the halting problem for a universal Turing machine, counter machines and many others is a (point-to-set) reachability problem.

In cellular automata, point-to-point reachability with almost periodic configurations (made of a same pattern indefinitely repeated except for finitely many cells) is usually considered. For instance the automaton 110 and the Game of Life are universal according to this definition. Why almost periodic configurations and not a wider, or smaller, countable family of points? This is discussed in [28].

For systems in $\mathbb{R}^n$, points with rational coordinates and sets defined by polynomial inequalities with rational coefficients (e.g., polyhedra or euclidian balls) are usually considered. The choice of rational numbers seems to be guided by simplicity only.

Let us give some examples of universal systems according to this definition.

- A piecewise-affine continuous map in dimension 2 [13]. This map is defined by a finite set of affine maps of rational coefficients, on domains delimited by lines with rational coefficients.
- Artificial neural networks for several kinds of saturation functions [27].
- A closed-form analytic map in dimension 1 [14].

We can define in a very similar way universal systems in continuous time. Examples of such systems are:

- A piecewise-constant derivative system in dimension 3 [2]. The state space is partitioned on finitely domains delimited by hyperplanes with rational coefficients, and the vector field is constant with a rational value on every domain.
- A ray of light between a set of mirrors [21].
- Black hole computation, which is the interaction of signals in space-time [11].

Despite its popularity and apparent simplicity, we believe that the reachability problems as a basis to define universality suffer from several drawbacks.

First, it is possible to reach unpleasant conclusions, such as the full shift being universal, by choosing artificial initial conditions for the system, as already highlighted in Section 3.

Second, the possibility offered to the controller to set up or observe a state of the system with infinite precision seems unphysical. The least uncertainty on the initial condition can *a priori* completely destroy the computation; ensuring that a physical system is, e.g., in a rational state is obviously an impossible task in practice. It has been shown that many reachability problems become decidable when perturbation is added to the dynamics, thus killing universality; see for instance [1,16,12].

Lastly, it is difficult to prove find interesting necessary or sufficient conditions of universality base on the dynamical properties of the system, as emphasized below in Section 10.

In next sections, we see two other halting problems that generalise the halting problem of Turing machine, and avoid some of the pitfalls mentioned above.

8 Digital computing machines

Suppose we have an arbitrary symbolic dynamical system. A symbolic dynamical system is one whose state is a sequence of symbols from a finite alphabet. In other terms, the state space is $A^{\mathbb{N}}$ or $A^{\mathbb{Z}}$, for a finite alphabet A, or a closed subset of it. Remember that such a set can be endowed with the product topology. The dynamical system is given by a continuous map on the state space.

Of course, the state space could also be $A^{\mathbb{Z}^d}$, for instance, which can be recoded into $A^{\mathbb{N}}$ by reading content of the cells in arbitrary order.

Symbolic dynamical systems include Turing machines, cellular automata and subshifts. No actuation is needed, the same map is applied at every step, the controller here is only an observer.

What is the most natural halting problem for such a dynamical systems? In other words, what is a universal *digital computing machine*?

A *cylinder* is a set of $A^{\mathbb{N}}$ of the form $wA^{\mathbb{N}}$, for any word $w \in A^*$, or a set of the form ${}^{\mathbb{N}}AwA^{\mathbb{N}}$ in $A^{\mathbb{Z}}$. Boolean combinations of cylinders are exactly the clopen (closed open sets) of the space.

We choose clopen sets as observation sets. This is a natural choice because it means that finitely many symbols are observed at every steps, and it means that a finite precision measurement is required. The only initial set is the full space; in other terms, nothing is known about the initial state of the dynamical system. Any deterministic finite automaton can be chosen as controller. This halting problem was proposed in [7]. We therefore say that a symbolic system is a universal digital computing machine if this halting problem is r.e.-complete.

It was shown [7] that a universal Turing machine is also universal for this definition, with some mild modifications. It also has the advantage to lead to non trivial conditions on the dynamical properties of the system for universality to emerge; see Section 10.

The digital computing machines show some robustness to perturbation, because a small enough perturbation on the initial condition of a successful trajectory (i.e., leading the controller to a final state) will keep the trajectory successful.

9 Actuation of dynamical systems

Turing machines are interpreted very simply as closed loop systems. Given a finite set A of symbols (including a blank symbol), we consider the set $A^{\mathbb{Z}}$ (or its restriction to finite configurations if we want a countable state space; finite configurations are those that are entirely blank except for finitely many symbols). On this set we have the following possible actuations: shift to the left, shift to the right or change the symbol in position zero to another

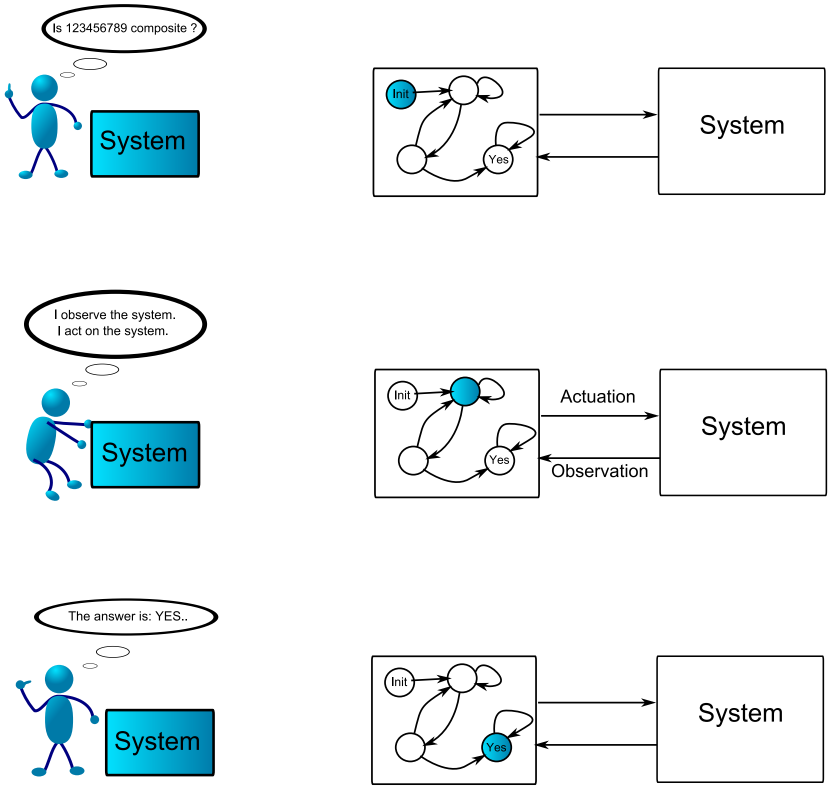


Fig. 1. Turing's argument revisited: How to compute with a dynamical system.

symbol. The only possible observation is the symbol in position zero. This is a simple open (input/output) dynamical system. If we now control it with an arbitrary finite automaton, what we get is exactly a Turing machine. This open dynamical system is therefore universal.

Despite this fundamental example, most set-ups proposed in the literature make no use of the actuation we let the dynamical system evolve by itself, without influence, except possibly at the very step to set up the initial condition. A slightly more elaborate system is proposed in [23], in which we can act on the system with three maps: f_0, f_1, f. The dynamical system starts at the origin (or any point fixed once for all), then we apply a sequence of f_0 and f_1 to introduce a binary word encoding the data; for instance, the number 100111 is encoded by the state $f_1 f_1 f_1 f_0 f_0 f_1(0)$. Then we apply f repeatedly.

10 Dynamical properties of universal systems

What is the link between the dynamics of a system and its computational capabilities?

Wolfram proposed a loose classification of 1-D cellular automata based on the patterns present in the space-time diagram of the automaton; see [30]. He then conjectured that the universal automata are in the so-called ‘fourth class’, associated to the most complex patterns.

Langton [15] advocated the idea of the ‘edge of chaos’, according to which a universal cellular automaton is likely to be neither globally stable (all points converging to one single configuration) nor chaotic. See also [20] for a discussion. Other authors argue that a universal system may be chaotic; see [27].

However it seems difficult to prove any non-trivial result of this kind with the point-to-point or point-to-set reachability definition of universality. Moreover a countable set of points can be ‘hidden’ in a very small part of the state space (nowhere dense, with zero measure for instance), so the link between this set and the global dynamics is unclear in general.

Digital computing machines are analysed in [7], where it is shown that a universal system according to this definition has at least one proper closed subsystem, must have a sensitive point and can be Devaney-chaotic.

11 Conclusions

A framework has been proposed to unify many of the definitions of computing machines found in the literature. A universal computing machine is defined as a dynamical system together with a suitable r.e.-complete problem; this definition is flexible with respect to the kind of dynamical systems we consider.

In particular it appears that reachability problems, despite their mathematical interest, are not the only generalisation of the Turing machine’s halting problem, and perhaps not the most natural.

In particular, it appears to us that universality for open dynamical systems is an almost blank field waiting to be explored.

12 Acknowledgements

The author is indebted to José Felix Costa for pointing some references.

This paper presents research results of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office. It has been also supported by the ARC (Concerted Research Action) "Large Graphs and Networks", of the French Community of Belgium. The scientific responsibility rests with its authors. The author is holding a FNRS fellowship (Belgian Fund for Scientific Research).

References

[1] E. Asarin and A. Bouajjani. Perturbed turing machines and hybrid systems. In *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science (LICS-01)*, pages 269–278, Los Alamitos, CA, June 16–19 2001. IEEE Computer Society.

[2] E. Asarin, O. Maler, and A. Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138(1):35–65, 1995.

[3] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21:1–46, 1989.

[4] O. Bournez and M. Cosnard. On the computational power of dynamical systems and hybrid systems. *Theoretical Computer Science*, 168:417–459, 1996.

[5] M. Davis, editor. *The Undecidable, Basic Papers on Undecidable Propositions, Unsolvable Problems And Computable Functions*. Raven Press, 1965.

[6] M. D. Davis. A note on universal Turing machines. In C.E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 167–175. Princeton University Press, 1956.

[7] J.-Ch. Delvenne, P. Kůrka, and V. D. Blondel. Computational universality in symbolic dynamical systems. *Fundamenta Informaticae*, 71:1–28, 2006.

[8] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London Ser. A*, A400:97–117, 1985.

[9] Francisco Dória and José Felix Costa, editors. *Special Issue of Applied Mathematics and Computation on Hypercomputation*, volume 178. Elsevier, 2006.

[10] B. Durand and Z. Róka. The Game of Life: universality revisited. In M. Delorme and J. Mazoyer, editors, *Cellular Automata: a Parallel Model*, volume 460 of *Mathematics and its Applications*, pages 51–74. Kluwer Academic Publishers, 1999.

[11] Jérôme Durand-Lose. Abstract geometrical computation 1: embedding black hole computations with rational numbers. *Fund. Inf.*, 74(4):491–510, 2006.

[12] Peter Gács. Reliable cellular automata with self-organization. In *38th Annual Symposium on Foundations of Computer Science*, pages 90–99, Miami Beach, Florida, 20–22 October 1997. IEEE.

[13] P. Koiran, M. Cosnard, and M. Garzon. Computability with low-dimensional dynamical systems. *Theoretical Computer Science*, 132(1-2):113–128, 1994.

[14] P. Koiran and C. Moore. Closed-form analytic maps in one and two dimensions can simulate universal Turing machines. *Theoretical Computer Science*, 210(1):217–223, 1999.

[15] C. G. Langton. Computation at the edge of chaos. *Physica D*, 42:12–37, 1990.

[16] W. Maass and P. Orponen. On the effect of analog noise in discrete-time analog computations. *Neural Computation*, 10(5):1071–1095, 1998.

[17] Y. V. Matiyasevich. *Hilbert's Tenth Problem*. MIT Press, 1993.

[18] M. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.

[19] M. L. Minsky. Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines. *Annals of Mathematics (2)*, 74:437–455, 1961.

[20] M. Mitchell, P. T. Hraber, and J. P. Crutchfield. Dynamic computation, and the "edge of chaos": A re-examination. In G. Cowan, D. Pines, and D. Melzner, editors, *Complexity: Metaphors, Models, and Reality*, Santa Fe Institute Proceedings, Volume 19, pages 497–513. Addison-Wesley, 1994. Santa Fe Institute Working Paper 93-06-040.

[21] C. Moore. Unpredictability and undecidability in dynamical systems. *Physical Review Letters*, 64(20):2354–2357, 1990.

[22] C. Moore. Recursion theory on the reals and continuous-time computation. *Theoretical Computer Science*, 162(1):23–44, 1996.

[23] C. Moore. Dynamical recognizers: real-time language recognition by analog computers. *Theoretical Computer Science*, 201:99–136, 1998.

[24] N. Ollinger. The intrinsic universality problem of one-dimensional cellular automata. In H. Alt and M. Habib, editors, *Symposium on Theoretical Aspects of Computer Science (Berlin, Germany, 2003)*, volume 2607 of *Lecture Notes in Computer Science*, pages 632–641. Springer, Berlin, 2003.

[25] M. Boykan Pour-El and I. Richards. Computability and noncomputability in classical analysis. *Transactions of the American Mathematical Society*, 275:539–560, 1983.

[26] M. O. Rabin. Recursive unsolvability of group theoretic problems. *Annals of Math.*, 67:172–194, 1958.

[27] H. T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Progress in Theoretical Computer Science. Springer-Verlag, 1999.

[28] K. Sutner. Almost periodic configurations on linear cellular automata. *Fundamenta Informaticae*, 58(3–4):223–240, 2003.

[29] K. Weihrauch. *Computable Analysis*. Springer-Verlag, 2000.

[30] S. Wolfram. *A new kind of science*. Wolfram Media, Inc., Champaign, IL, 2002.

Black hole computation: implementations with signal machines

Jérôme Durand-Lose
Laboratoire d'Informatique Fondamentale d'Orléans,
Université d'Orléans,
B.P. 6759,
F-45067 ORLÉANS Cedex 2.

1. Introduction

No position is taken on the theoretical and practical feasibility of using any potentially existing particular black hole for hyper-computing. The reader is referred to other contributions in this issue as well as Etesi and Németi (2002); Németi and Dávid (2006); Németi and Andréka (2006) for clear introductions and surveys on this topic. All the work presented here in done the context of so-called "Malament-Hogarth" space-times and not slowly rotating Kerr black hole. The main differences are that, one, the observer remains out of the black hole and second, nested black holes are considered.

A black hole computation works in the following way. At some location[⋆] one observer starts a computing device which he sends on a different world-line such that: at some point, after a finite time on his own world-line the observer has the whole future of the device infinite world-line in its causal past and the device can send some limited information to the observer. The device has an infinite time for computing ahead of it and can send a single piece of information to the observer. Not only can the observer *perceive* this piece of information, but after a finite duration which he is aware of, the observer knows for certain that if he has not received anything then the machine never sends anything during its whole, possibly infinite, computation. So that, for example, if the piece of information is sent only

[⋆] Throughout the article *location* is to be understood as a position in space and time and *position* as only spatial.

when the computation stops, at some point, the observer knows for certain whether the computation ever stops.

From the computer scientist point of view, this allows to have some piece of information on a whole, potentially infinite, computation. This way the Halting problem or the consistency of many first order logic theory (e.g. Set theory or Arithmetic theory) can be decided. This clearly falls out of the classical recursion/computability theory since it allows to decide semi-decidable subsets (Σ_1^0 in the arithmetic hierarchy).

Malament-Hogarth space-times allow to have nested black holes and so-called *arithmetical sentence deciding (SAD) space-times* (Hogarth, 1994, 2004). With each infinite level of nested black-holes, the computing power climbs up a level of the arithmetic hierarchy (and even the hyper-arithmetic hierarchy in second order number theory (Welch, 2006)). The arithmetic hierarchy is formed by recursive predicates preceded by alternating quantifiers on natural numbers. The level in the hierarchy is defined by the first quantifier and the number of alternations.

In computer science, considering and continuing a computation after its potential infinite span exist in various form. Infinite time Turing machines (Copeland, 2002; Hamkins and Lewis, 2000; Hamkins, 2002, 2007) consider ordinal times (and values); the main difference is that the limit of the tape is available, whilst with black holes only a finite and bounded piece of information is available. Infinite computations also provide limits, for example computable analysis (Weihrauch, 2000) (type-2 Turing machines) manipulates infinite inputs and generates infinite outputs, each one representing a real number. In analog computations, limit operators (as limits of sequences of real numbers) are also considered (Chadzelek and Hotz, 1999; Kawamura, 2005; Mycka, 2003b,a, 2006; Mycka and Costa, 2007; Bournez and Hainry, 2007).

The setting in which the black hole effect is simulated here is not the solution to any relativity equations. It is rather something that is constructed inside a Newtonian space-time and provide the desired effect whereas black holes involves non-Euclidean geometry (like the Schwarzschild one). Any initially spatially bounded computation can be embedded in a shrinking structure resulting in the same computation happening in a spatially and temporally bounded room, even if it was not the case before. This structure provide the black hole, outside of it, some information may be received to gain some insight on what happen inside.

Abstract geometrical computation (AGC) deals with dimensionless signals with rectilinear and uniform movement in a (finite dimension) Euclidean space. It is both a continuous and an analog model. The way it is analog is quite unusual because there are finitely many values but continuum many variables. Time is evolving equally everywhere. What brings forth the accelerating power is that space and time are continuous so that Zeno effect can happen and indeed, everything heavily relies on it. Throughout the article, only dimension 1 spaces are considered, thus space-time diagrams are 2 dimensional.

Signals are defined as instances of meta-signals which are of finite number. They move with constant speeds uniquely defined by their meta-signals. When two or more signals meet, a collision happen and they are replaced other signals according to collision rules. A *signal machine* defines the available meta-signals, their speeds and the collisions rules.

A configuration at a given time is a mapping from $\mathbb{R}$ to a finite set (containing the meta-signals) defining signals and their positions. Signals are supposed to be away one from another. The void positions (i.e. position with nothing) are supposed to form an open set. The rest of the configuration contains only singularities. They are many ways to continue a space-time diagram after an isolated singularity, the ones used here are: let it disappear or turn it into signals.

This model originates in discrete signals in Cellular Automata (Durand-Lose, 2008b). This explains the choice of finitely many meta-signals and constant speeds. There are other models of computation dealing with Euclidean spaces.

Huckenbeck (1989, 1991) developed a model based on finite automata able to draw lines and circles and to compute intersections. Not only are the primitives different, but also it is sequential and depends on an external operator to perform the construction whereas in signals in AGC operate on their own.

Jacopini and Sontacchi (1990) developed an approach where a computation results in a polyhedron. They only encompass finite polyhedron (i.e. bounded with finitely many vertices) and allow surfaces and volumes of any dimensions while AGC has only lines.

Another model worth mentioning is the Piecewise Constant Derivative (PCD) system (Bournez, 1999b). There is only one signal, but its speed changes each time it enters a different polygonal region. Not only is AGC parallel but also there is no such things as boundaries. Nevertheless, PCD are able to hyper-compute and climb up the hyper-arithmetic hierarchy (Bournez, 1999a) while the other two cannot.

In AGC, since signals dwell in a continuous space ($\mathbb{R}$) and time ($\mathbb{R}^+$) and there is no absolute scale, it is possible to rescale a configuration. Rescaling a configuration rescales the rest of the space-time. An automatic procedure to freeze the computation, scale it down and unfreeze it, is provided. When this construction is made to restart it-self forever, a singularity is reached. Any AGC-computation starting with only finitely many signals can be embedded in this structure so that the corresponding computation is cut in countably many bounded pieces geometrically shrunk and inserted. This brings a general scheme to transform any AGC-computation into one that can do the same computation but in a piece of the space-time diagram bounded in both space and time. This is the desired black hole effect. Another shrinking scheme is presented in Durand-Lose (2006a), but it works only for spatially bounded AGC-computations while the one presented here does not impose this condition. The construction is detailed since AGC is not so well-known and it is the cornerstone of the results in the article.

Simulating a Turing machine (TM) is easy using one signal to encode each cell of the tape and an extra signal for the location of head and the state of the finite automaton. Any Turing machine simulation can be embedded in the shrinking structure making it possible to decide semi-recursive problems in finite time.

As long as speeds and initial positions are rational and there is no singularity, the location of each collision is rational (as the intersection of lines with rational coefficients). This can be computed in classical discrete computability (and have been implemented to generate figures). The model is also perfectly defined with irrational speeds and positions and can thus be also considered as an analog model. It has recently been connected to the Blum, Shub and Smale model of computation (which can perform algebraic operations as well as test the sign on real numbers with exact precision) (Durand-Lose, 2007, 2008a). In the present paper, it is only recalled how real numbers are encoded and how a singularity is used to provide the multiplication. Our shrinking scheme can be used in this analog context to answer analog decision problems.

The way a signal machine is transformed to have shrinking capability can be iterated so that singularities/accumulations of any order can be generated. This allows to decide a formula formed by a total (resp. BSS) recursive predicate preceded by a finite alternation of quantifiers on ℕ (i.e. to climb the corresponding arithmetical hierarchies). For the analog case, this remains a quantification over a countable set[**].

[**] This is why we do not talk about an analytic hierarchy.

In Section 2, the model and the shrinking structure are presented. In Section 3, the way discrete computation can be done and shrunk is presented. In Section 4, analog computation is considered. Section 5 explains how to have nested black holes so as to climb the arithmetic hierarchies. Section 6 gathers some concluding remarks.

2. Model and Mechanics

The underlying time is $\mathbb{R}^+$; there is no such thing as a next configuration. The underlying space is $\mathbb{R}$. A configuration is a mapping from $\mathbb{R}$ to a finite set (yet to be defined). A space-time diagram is a function from $\mathbb{R} \times \mathbb{R}^+$ to the same finite set.

What lies on $\mathbb{R}$ are signals, collisions between signals or singularities (created by accumulations). Signals are moving with constant speed depending only on their nature. When they meet, an instantaneous collision happens and the signals are replaced by others according to some collision rules. A singularity happens when and where infinitely many collisions, signals and/or singularities accumulate. Ways to continue a space-time diagram beyond an isolated singularity are proposed at the end of this section.

2.1. *Signal machines and space-time diagrams*

Definition 1 A *signal machine (SM)* is defined by (M, S, R) where M (*meta-signals*) is a finite set, S (*speeds*) a function from M to $\mathbb{R}$, and R (*collision rules*) a partial function from the subsets of M of cardinality at least two into subsets of M (all these sets are composed of signals of distinct speeds).

Each signal is an instance of a meta-signal. Its speed is constant and only depends on its meta-signal (given by S). When two or more signals meet, R indicates the signals to replace them. Meeting signals must have distinct speeds otherwise they are parallel and never meet. Signals are not allowed to be superposed, so that all signals emitted by a collision must also have distinct speeds.

Definition 2 The *extended value set*, V, is the union of M and R plus two special values: one for void, $\oslash$, and one for singularity ❊. An *(valid) configuration* is a total function from $\mathbb{R}$ to V such that all the accumulation points of its *support* (the set of non void location, $supp(c) = \{\, x \in \mathbb{R} \mid c(x) \neq \oslash \,\}$) have the value ❊. A configuration is *finite* if its support is finite. It is *simple* if it is finite and ❊ is not reached. It is *rational* if it is simple and its support is included in $\mathbb{Q}$. A SM is *rational* if all the speeds are rational numbers and only rational configurations are used.

As long as there is no singularity, a finite configuration is only followed by finite ones.

To be rational is *robust*. Signals at rational positions with rational speeds can only meet at rational location. All collisions happen at rational dates and at these dates the positions of signals are all rational. Since rational numbers can be encoded and manipulated exactly with any computer, rational SM can be handled inside classical computability theory.

Two results limit the interest and extend of rational SM. First, predicting whether a singularity ever happens is Σ_2^0-complete (in the arithmetical hierarchy, which means not even semi-decidable) (Durand-Lose, 2006b) and second, a singularity can happen at an irrational position (Durand-Lose, 2008a). On the other hand, as long as Turing-computability is involved, rational SM are enough (as shown in Sect. 3). But if analog computations are to be considered, then it is not the case anymore as in Sect. 4.

Let S_{min} and S_{max} be the minimal and maximal speeds. The *causal past*, or *backward light-cone*, arriving at position x and time t, $I^-(x,t)$, is defined by all the positions that might influence the information at (x,t) through signals, formally:

$$I^-(x,t) = \{\ (x',t') \mid x - S_{max}(t{-}t') \leq x' \leq x - S_{min}(t{-}t')\ \}\ .$$

Definition 3 The *space-time diagram* issued from an initial configuration c_0 and lasting for T, is a function c from $[0,T]$ to configurations (i.e. a function from $\mathbb{R} \times [0,T]$ to V) such that, $\forall (x,t) \in \mathbb{R} \times [0,T]$:

(i) $\forall t{\in}[0,T]$, $c_t(.)$ is valid, and $c_0(.) = c_0$;

(ii) if $c_t(x){=}\mu \in M$ then $\exists t_i, t_f{\in}[0,T]$ with $t_i{<}t{<}t_f$ or $0{=}t_i{=}t{<}t_f$ or $t_i{<}t{=}t_f{=}T$ s.t.:

  (a) $\forall t' \in (t_i, t_f)$, $c_{t'}(x + S(\mu)(t'-t)) = \mu\,$,

  (b) $(\,t_i{=}0$ and $c_0(x_i) = \mu\,)$ or $(\,c_{t_i}(x_i) = \rho^-{\to}\rho^+{\in}R$ and $\mu \in \rho^+\,)$ where $x_i{=}x + S(\mu)(t_i - t)\,$,

  (c) $(\,t_f{=}T$ and $c_{t_f}(T) = \mu\,)$ or $(\,c_{t_f}(x_f) = \rho^-{\to}\rho^+{\in}R$ and $\mu \in \rho^-\,)$ or $c_{t_f}(x_f){=}❊$ where $x_f{=}x + S(\mu)(t_f - t)\,$;

(iii) if $c_t(x){=}\rho^-{\to}\rho^+ \in R$ then $\exists\varepsilon$, $0{<}\varepsilon$, $\forall t'{\in}[t{-}\varepsilon, t{+}\varepsilon] \cap [0,T]$, $\forall x'{\in}[x - \varepsilon, x + \varepsilon]$,

  (a) $(x',t') \neq (x,t) \Rightarrow c_{t'}(x') \in \rho^-{\cup}\rho^+ \cup \{\oslash\}$,

  (b) $\forall \mu{\in}M$, $c_{t'}(x'){=}\mu \Leftrightarrow \vee \begin{cases} \mu \in \rho^- \text{ and } t' < t \text{ and } x' = x + S(\mu)(t'-t) \\ \mu \in \rho^+ \text{ and } t < t' \text{ and } x' = x + S(\mu)(t'-t) \end{cases}$;

  and

(iv) if $c_t(x) = ✻$ then $\forall\varepsilon{>}0$, there are infinitely many collisions in $I^{-}(x,t) \cap \mathbb{R} \times [t{-}\varepsilon, t)$ or infinitely many signals in $[x-\varepsilon, x+\varepsilon] \times [t{-}\varepsilon, t)$.

The definition naturally extends to the case $T = +\infty$. A space-time diagram is *rational* if it is correspond to the one of a rational SM. As long as no singularity is reached, the evolution is deterministic; the space-time diagram only depends on c_0 and the SM.

2.1.1. *Encompassing singularities.*

When a singularity is reached isolated (i.e. there is nothing round it), there are various ways to *continue* the space-time diagram beyond it.

Definition 4 (Schemes to handle isolated singularities) A singularity at (x,t) is *isolated* if there is nothing but void around it in the configuration (i.e. $\exists\varepsilon, \forall x' \in [x-\varepsilon, x+\varepsilon], x' \neq x \Rightarrow c_t(x') = \oslash$). There are various schemes to continue a space-time diagram at an isolated singularity.

(i) *Wall.* It remains there forever;
(ii) *Vanish.* It disappears as if it where on another dimension;
(iii) *Simple trace.* It disappears but sends a μ_s signal; or
(iv) *Conditional traces.* It disappears but sends signals according to which signals join in (i.e. signals that are not interrupted by any collision –not an infinite succession of signals).

In the two last schemes, μ_s or the singularity rules have to be added to the definition of the SM. In such a case, one talks about an *extended SM (ESM).* Next sections consider ESM and the schemes used are indicated.

The first scheme is not considered here; although it makes sense since singularities generally do not just disappear, in dimension one, it unfortunately produces an unbreakable frontier definitively splitting the configuration in two. The second one is considered in the section on discrete computation while the third is considered in the section on analog computation. The reason in the analog case is that its position is an important piece of information. The last case is also used to consider the case of a singularity happening exactly on a higher level structure (as the one presented below) with nested structures in Sect. 5.

The last two schemes impose a modification of the rules of space-time diagrams. One is to allow a signal to start from a singularity to add to b "or $(\, c_{t_i}(x_i){=}✻ \wedge \mu \,=\, \mu_s \,)$" and to iv something amounting for signals emitted like the rules for collisions (iii) which is not given since it depends on the scheme and for the last scheme of the singularity rules.

In case of non-isolated singularities, the fist two schemes remain possible while the two last ones would make it necessary to define signals with a dimension (which might be not an integer since the accumulation set can be a Cantor set).

Accumulations of second (or more) order can be considered, as long as all the singularities are isolated. In the last case, more distinction could be made according to the level of singularity. As long as finitely many levels are considered or distinguished, the description remains finite.

2.2. *Shrinking and space and time bounding*

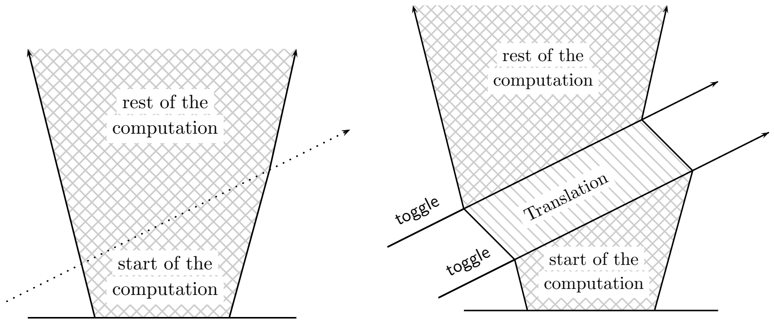
All the constructions work in the following way: starting from a SM, new meta-signals and rules are added in order to generate another SM that works identically with original meta-signals but the new ones provide an extra layer with some kind of meta instructions. For a computation to be altered, extra signals have to be added to the initial configuration.

2.2.1. *Freezing and unfreezing.*

A new meta-signal, toggle, with a speed strictly greater than any present speed, say s_0 is added. If one instance of toggle is set on the left, and is re-generated by every collision, then a succession of toggle (signals) crosses the entire computation. It is the freezing signal. Freezing is done in the following way: each time it meet some signal μ, it replaces it by some frozen counterpart ${}^{\mathsf{F}}\mu$. All these new frozen meta-signals have the same speed, say f_0 (which should be less that s_0 to ensure that they are above the freezing line on the space-time diagram). It might happen that the freezing signal passes exactly on a collision, say ρ. Then the collision is frozen too, i.e. it is replaced by a frozen signal amounting for it, ${}^{\mathsf{F}}\rho$. The signals resulting from the collision are generated when the collision is unfrozen.

The unfreezing scheme is the same, in reverse: signals and collisions replace their frozen counterparts at the passing of another toggle signal.

This scheme is illustrated on Fig. 1. In all the figures, time is elapsing upwards. On the left, one computation is showed unaltered but the intended trace of toggle is indicated with dotted lines. On the right, the computation is frozen, the frozen signals (the ${}^{\mathsf{F}}.$) are left to move for some time and then they are unfrozen by a second toggle. Signal are shifted on one side, they could be shifted on the other side by symmetry, or shifted anywhere (as long as it is in the future) by moving the toggling signals and changing the inner slope.



(a) Regular evolution.

(b) Frozen, translated and unfrozen.

Fig. 1. Freezing principle.

The algorithm to modify a SM is given on Fig. 2 where object oriented notations are used. Each time new identities are created, renaming is used if necessary. There is nothing particular about it and every modification works on the same pattern.

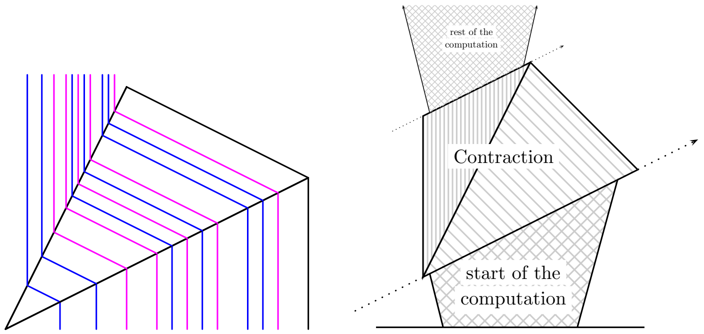
```
Input:
    $M$: signal machine
    $\beta$: real number { speed of the toggle }
    $\theta$: real number { speed of frozen signals }
Assert: ( $S_{max} < \beta$) $\wedge$ ( $\theta < \beta$)
Do:
    { Create the toggle }
    toggle $\leftarrow$ $M$.add_new_meta-signal_of_speed( $\beta$ )
    { For the meta-signals }
    for each $\mu$ original meta-signal from $M$ do
        ${}^{\mathsf{F}}\mu \leftarrow M$.add_new_meta-signal_of_speed( $\theta$ )
        $M$.add_rule( {toggle, $\mu$ } $\rightarrow$ {toggle, ${}^{\mathsf{F}}\mu$} )
        $M$.add_rule( {toggle, ${}^{\mathsf{F}}\mu$} $\rightarrow$ {toggle, $\mu$ } )
    end for
    { For the rules }
    for each $\rho = \rho^- \rightarrow \rho^+$ original rule from $M$ do
        ${}^{\mathsf{F}}\rho \leftarrow M$.add_new_meta-signal_of_speed( $\theta$ )
        $M$.add_rule( { toggle } $\cup\, \rho^- \rightarrow$ { toggle, ${}^{\mathsf{F}}\rho$ } )
        $M$.add_rule( { toggle, ${}^{\mathsf{F}}\rho$ } $\rightarrow$ { toggle } $\cup\, \rho^+$ )
    end for
Output: toggle: meta-signal { the freezing/unfreezing one }
Side effect: New signals and rules added to $M$
```

Fig. 2. Algorithm to add the freezing capability.

2.2.2. *Scaling parallel signals and any computation.*

When signals are parallel, they remain so and do not interact, their structure is quite loose. So that a group of parallel signals can be manipulated quite easily as long as they remain parallel they remain *synchronized*. They just have to be unfrozen with the same slope.

Using a Thales based construction, or prismatic if thought of as light beams, it is easy to scale parallel signals as depicted on Fig. 3(a). The idea is to change the direction twice to be scaled and recover the original direction. The speed of the signals crossing the triangle is carefully selected in order to insure the wanted ratio.



(a) Scaling parallel signals. (b) Freeze, Scale and unfreeze.

Fig. 3. Scaling.

This construction can be set inside the freezing/unfreezing construction. This leads to the scheme of Fig. 3(b). The specially added signal for the structure are only a part of the ones in the next subsection. The algorithms to modify SM are so plain that they are not presented anymore.

2.2.3. *Iterating forever.*

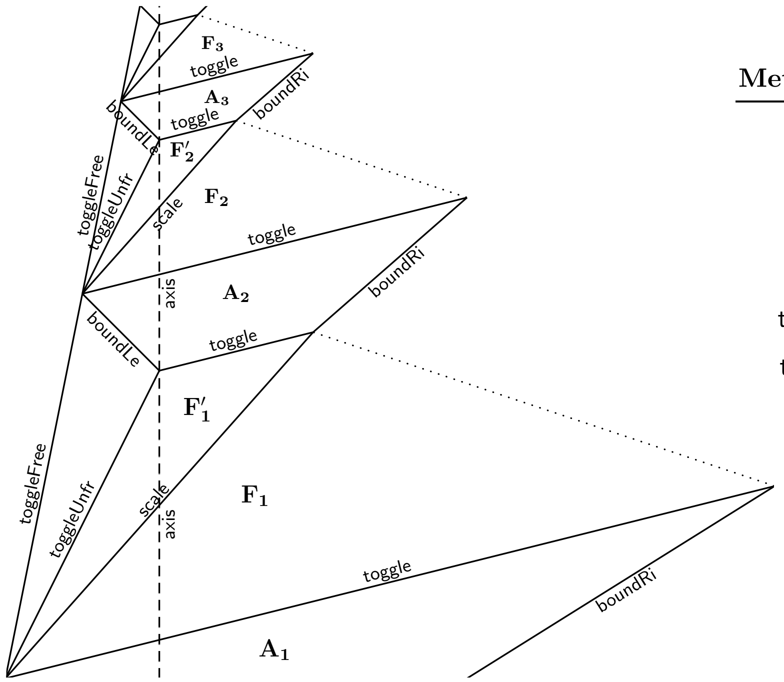
The idea is to iterate *ad infinitum* above construction. More signals have to be added in order to restart the freeze, scale down and unfreeze process on and on. The structure and some basic properties are presented before the embedding of computations.

The structure is presented on Fig. 4. The collision rules are not given because they can be read directly from the figure; for example

$$\{\mathsf{toggleUnfr}, \mathsf{axis}\} \to \{\mathsf{boundRi}, \mathsf{axis}, \mathsf{toggle}\} \ .$$

The dashed axis signals are part of the construction, while the dotted lines do not correspond to any signal and are only there to delimit some regions. Signal axis is dashed because it does not delimit any region.

The toggleUnfr and toggleFree signals are used to start the freezing and unfreezing toggle. The scale signals are used both to change the direction of parallel signals (and achieve scaling down the embedded configuration) and also to start boundRi. The signals boundLe and boundRi are used to bound the potential room used by the embedded computation when active. The



Meta-signal	**Speed**
boundLe	$-\nu_0$
boundRi	ν_0
scale	$\frac{8}{9}\nu_0$
toggle	$4\nu_0$
toggleUnfr	$\frac{1}{2}\nu_0$
toggleFree	$\frac{1}{5}\nu_0$
axis	0

Fig. 4. iterated scaling.

initial position of axis is at one third of the distance between the four on the left and boundRi.

The construction works because the speed parameter ν_0 used to compute the various speeds is equal to the maximum absolute value of the speeds in the original SM. The speeds given on Fig. 4 are computed such that:

– at each iteration, the structure is scaled by half; and
– the length of the unfreezing toggle signal is one fourth of the preceding freezing one.

The structure is twice as small and twice as fast each time but the initial computation is scaled by one fourth. Relatively to the structure the computation is two times faster each time. This is wanted because not only should the structure collapse in finite time, but meanwhile the embedded computation should have infinite time ahead of it. This is to be understood considering the regions.

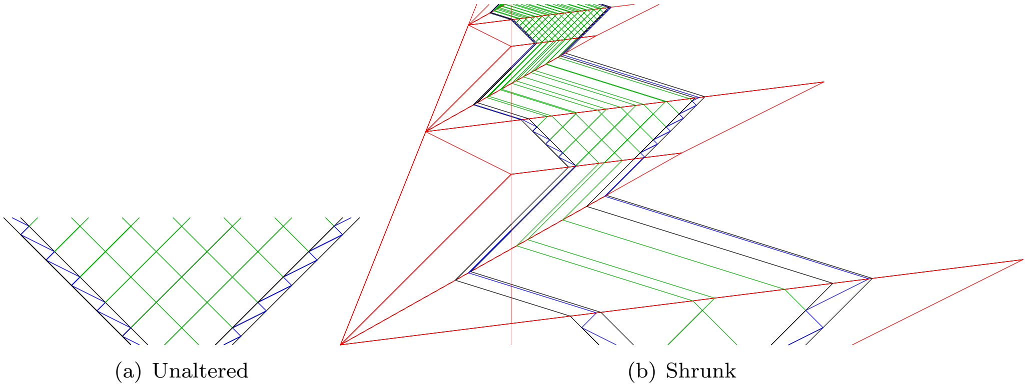
The $\mathbf{A_i}$ regions (on Fig. 4) are the ones where the embedded computation is active. The first one is triangular while all the other ones are trapezoidal. The other two regions are triangular. The $\mathbf{F_i}$ regions are the ones where the embedded computation is frozen and signals follow the direction of the dotted line. The $\mathbf{F'_i}$ regions are the ones where the embedded computation is also frozen but signals follow the direction of toggleUnfr. The frontiers between $\mathbf{A_i}$ and $\mathbf{F_i}$ (as well as $\mathbf{F'_i}$ and $\mathbf{A_{i+1}}$) are toggle signals thus the correct freeze (and unfreeze). The frontiers between $\mathbf{F_i}$ and $\mathbf{F'_i}$ are scale signals which correspond to a change of direction of frozen signals.

On the frozen regions $\mathbf{F_i}$, all frozen signals have to be parallel to the dotted line in order that the lower toggle is "mapped bijectively" onto scale. So that their speed is $-\frac{8}{5}\nu_0$. On $\mathbf{F'_i}$ all frozen signals have to be parallel to the scaleLo signal in order that scale is "mapped bijectively" onto the upper toggle. So that their speed is $\frac{1}{2}\nu_0$ (toggleUnfr).

The embedded configuration is scaled by one fourth but the piece is only one half in size of the previous one. Each time the duration, relatively to the original computation, is halved by 2 (for the structure size) but multiplied by 4 for the scale. Altogether, the ratio is 2. So that each time, the elapsing time for the original computation is doubled. This ensures that the original computation is entirely embedded, i.e. has infinite time ahead of it.

Figure 5 shows a simple space-time diagram that is growing on both side on the left and its embedded version on the right. The active part is where the lattice is visible, otherwise it is frozen, shift and scaled.

The location of the singularity can be computed easily as the intersection of two lines (but also as a geometrical sum) and is rational as long as it is stated with rational positions.



(a) Unaltered

(b) Shrunk

Fig. 5. Iterated scaling example.

The shrinking structure serves as a black hole. The configuration embedded inside is trapped. In the following sections, small modifications of already modified SM show how to let some information leave.

Definition 5 A *Turing machine* (TM) is defined by $(Q, q_i, \Gamma, \hat{}, \texttt{#}, \delta)$ where Q is a finite set of *states*, q_i is the *initial state*, Γ is a finite set of *symbols*, ^, *tape head*, and #, *blank*, are two distinguished symbols, and $\delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \to\}$ is the *transition function*.

A *TM-configuration* is defined by (q, w, i) such that q is a state, w is a finite sequence of symbols –or word over Γ– and i is a natural number. The automaton is in state q, w (completed by #'s) is written on the tape and the head is over the ith cell of the tape. The TM is *self-delimiting* when there is only one ^ on the tape which is written on the first cell, and there is nothing but # right of the first #. If it is not the case, the TM is in an illegal configuration.

The *next configuration* is defined by the transition function as follows. If $\delta(q, w_i) = (r, \texttt{a}, \to)$ then the next configuration is defined by $(r, w', i + 1)$ where w' is obtained from w by replacing the ith symbol by a. If $\delta(q, w_i) = (r, \texttt{a}, \leftarrow)$ then, if $i = 0$ then the TM stops otherwise the next configuration is defined by $(r, w', i - 1)$.

The TM *computes* in the following way: some input (without ^ and #) is written on the tape preceded by ^ and followed by potentially infinitely many #. The result of the computation, if any, is what is written on the tape (^ and all # are discarded) when the TM stops.

The TM halts normally when the head tries to leave the tape. For example, the machine defined on Fig. 6(b), computes on Fig. 6(a) with the entry ab. The output is bbab. If the computation does not halt or enters an illegal configuration, the output is undefined.

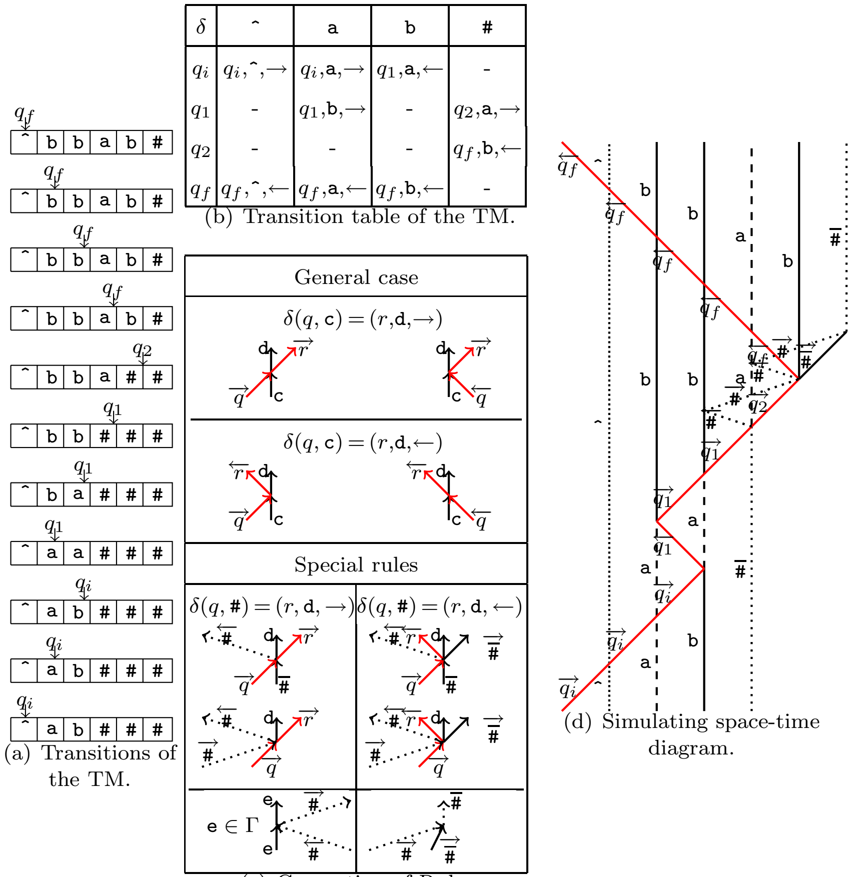
When a TM is used for a decision (yes/no output), the states are partitioned into accepting and refusing one. The answer is then given by the state in which it halts.

3.1. *Turing-machine simulation*

The simulation goes as follows: there are finitely many parallel signals encoding the content of the tape in-between which zigzags a signal mimicking the movement of the head and encoding the state. This is depicted with an example on Fig. 6(d).

The simulating SM is defined by the following meta-signals:

- one *symbol signals* for each value in Γ, with null speed, to encode the cells of the tape;



(a) Transitions of the TM.

δ	ˆ	a	b	#
q_i	q_i,ˆ,→	q_i,a,→	q_1,a,←	-
q_1	-	q_1,b,→	-	q_2,a,→
q_2	-	-	-	q_f,b,←
q_f	q_f,ˆ,←	q_f,a,←	q_f,b,←	-

(b) Transition table of the TM.

(c) Generation of Rules.

(d) Simulating space-time diagram.

Fig. 6. Example of a TM computation and its simulation by a SM.

- $\overrightarrow{q}$ (of speed 1) and $\overleftarrow{q}$ (of speed -1) *head signals* for each state q, to encode the state and the location and movement of the head;
- $\overline{\#}$ (of null speed), $\overleftarrow{\#}$ (of speed -3), $\overrightarrow{\#}$ (of speed 3), and $\overset{\rightrightarrows}{\#}$ (of speed 1) which are used to denote the end of the tape and manage the enlargement of the tape.

The initial SM-configuration for a TM-computation on the entry $w = w_1w_2\ldots w_k$ is generated by putting one $\overrightarrow{q_i}$-signal at position -1, one ˆ-signal at position 0, one w_i-signal at position i ($1 \leq i \leq k$), and one $\overline{\#}$-signal at position $k+1$,

A SM-configuration encodes a TM configuration if it corresponds to the same sequence of symbols (up to extra #'s at the end) closed by a $\overline{\#}$ with an extra signal encoding the state and moving to meet the signal corresponding to the position of the head.

The collisions rules ensure that the evolution of the SM corresponds to the computation of the TM. When the head encounters a symbol signal, it performs the update and move to its next position on the tape. When it meets the right special signal, the configuration is automatically enlarge.

The generated collision rules are given on Fig. 6(c). When a rule is not defined, the signal just cross unaffected. From top to bottom, they correspond to the following cases. For each TM-transition, two collision rules are generated, they correspond to the cases where the head would come from the left or from the right. The special rules are made in order to ensure a correct enlargement of the configuration in the case a head signal meet $\overline{\texttt{\#}}$, the signal marking the last cell. In such a case, two things have to be done: normally do the TM-transition (as if it were a # signal) and enlarge the configuration. The latter means generate one $\overline{\texttt{\#}}$ one position on the right if the head goes left (left side rules). If the head goes right (right side rules), then it should be taken care that the head meets something on the right (lower row rules). This is illustrated in the middle of Fig. 6(d). In each first case, a $\overleftarrow{\texttt{\#}}$ is sent on the left. It bounces on the symbol signal on the left (bottom rule of Fig. 6(c)) and is replaced by $\overrightarrow{\texttt{\#}}$, cross whatever signal present to get where would be the head if it would have gone right. If indeed the head went right, it is met and the TM-transition is done and the configuration enlargement starts again. If it is not the case (left rules of Fig. 6(c)), then $\overrightarrow{\overline{\texttt{\#}}}$ is sent in order to meet $\overrightarrow{\texttt{\#}}$ and place the new $\overline{\texttt{\#}}$. Signals $\overleftarrow{\texttt{\#}}$ and $\overrightarrow{\texttt{\#}}$ are three time faster in order to ensure that the meeting happens exactly where the next symbol signal should have been.

At the end of the TM-computation simulation, the signal encoding the state goes out on the left. The process is robust, the positions does not have to be exact as long as the order of the signals is preserved. It can be implemented with a rational SM.

3.2. *Infinite Turing computation in bounded time*

The construction described in Sect. 2 rightfully apply to the above construction. It is considered that the singularity vanishes leaving no signal (scheme ii of Def. 4). The infinite acceleration is granted. But the information leaving the black hole is missing. The ESM has to undergo some modifications. The first one is to generate the escaping signal, the second to retrieve it.

The result of a deciding TM is given by the new state whenever it performs a transition on ˆ and sends the head on the left. These transitions are clearly identified in the transition table of the TM. So there are also clearly identified in the rules of the simulating ESM.

Once the shrinking mechanics have been added, the rules can be changed so that to generate new signals amounting for accept and refuse. These signals are unaffected by other signals including the structure ones. Their speed is -1 so that they exit on the left.

The last thing is to add two signals orizonLe and orizonRi of speed ($\frac{1}{5}\nu_0$ and $\frac{-16}{15}\nu_0$) on the left and right of the shrinking structure. Their speeds ensure that they remain at constant distance from the structure. The way these signals are intended to work is displayed on Fig. 7

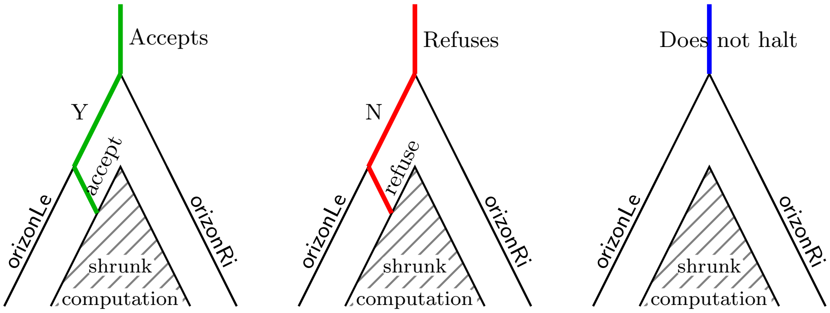


Fig. 7. Encapsulating the shrinking structure to retrieve any leaving signal.

The desired effect is achieved. Recursively enumerable problems (Σ^0_1) can be decided and computation can carry on after the answer is known.

4. Analog computations

In this section, analog computations in the understanding of the Blum, Shub and Smale model (Blum et al., 1989, 1998) (BSS for short) are considered. After briefly presenting the BSS model on $\mathbb{R}$ and its linear restriction, results on its simulation in AGC are recalled before the shrinking construction is applied.

4.1. *BSS Model*

BSS machines (BM) operate like TM on unbounded arrays/tapes. Each cell of the array hold a real number and all the computations are made with exact precision. The BM has a head allowing to access a finite portion of the tape called a *window*. It has finitely many states and to each one corresponds an instruction among the following:

(i) compute a polynomial function of the window and place the result in it;

(ii) test for the sign of a given cell in the window and branch accordingly;
(iii) shift the window one cell on the left or on the right; and
(iv) halt.

The window is shifted by one cell so that consecutive windows overlaps. This is used to carry information around, since the BM has no real value storage on its own.

A BSS machine is *linear* if instruction i is replaced by "compute a linear function [...]". Thus multiplication is allowed only if it is by a constant.

Like for TM, the input (resp. output) is the content of the array when the BM is started (resp. halts). Comparing to TM, trying to leave the array on the left is an error but halting states are provided. If BM are used for decision, then there are distinct halting states for acceptance and refusal.

The simulation is not presented since details are not relevant to the following. Only the encoding of real values and two main results are given.

A real number is encoded as the distance between two signals. But as it has already been guessed by the reader, time and distance are very relative concepts. The distance between two scale signals is thus used as a scale. The same scale is used for all the real numbers so that each one can be encoded by a just pair base and value (or just nul for 0 since superposition of signals is not allowed). This is illustrated for various values on Fig. 8.

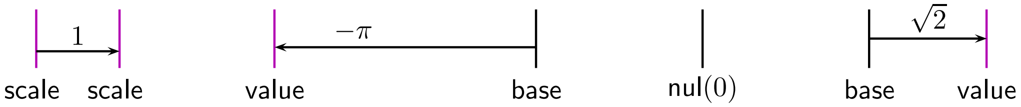


Fig. 8. Encoding: scale and values $-\pi$, 0 and $\sqrt{2}$.

Theorem 6 ((Durand-Lose, 2007)) *AGC with finitely many signals and no singularity is equivalent to the linear BSS model.*

Simulations have been established in both directions. In the constructions for SM simulation, the encoding pairs are manipulated so that they are clearly identified and everything is scaled down whenever they might mess up.

Theorem 7 ((Durand-Lose, 2008a)) *With the Simple trace scheme (iii of Def. 4) for singularities, AGC is able to carry out any multiplication achieving the simulation of the whole BSS.*

In this case AGC is strictly more powerful than classical BSS since square root can be computed.

The meaning of the shrinking structure on a BM simulation is considered before entering into technical issues.

4.2.1. *Interpretation.*

Assuming that everything works well as expected, the ESM can be made so that if the computation stops then some information leaves the singularity. But what kind of information? Usually it is considered that only finitely many values can be used***, not countably many, not continuum many! Mechanisms as in Subsect. 3.2 can be used to send a few bits, but it is not clear how to send four signals (to encode a real number) ensuring that they are all at the same scale (i.e. are emitted from the same active region).

With the same construction as previously, using a universal BM, the BSS halting problem, can be decided.

Instance HALT$_{\text{BSS}}$

- $\overrightarrow{X}$, $\overrightarrow{Y}$: vectors of real numbers

Question

Does the BM represented by $\overrightarrow{X}$ stop on entry $\overrightarrow{Y}$?

Let us consider another example. BSS machines can simulate any TM and any recursive function. There is a BM with one entry that tries all the rational numbers sequentially and stops if it is equal to the entry. Since halting can be decided, the characteristic function of $\mathbb{Q}$ is computable (which is not usually the case).

The general form of decision problems that can be solved is of the form $\exists n \in \mathbb{N}, \phi(n, \overrightarrow{X})$ where ϕ is decision BM that stops for all entries. This definition is the same as the one for classical recursion except that total recursive predicate is replaced by total BSS predicate. The quantification is on $\mathbb{N}$ (that can encode $\mathbb{Q}$ or many other countable sets) but not on $\mathbb{R}$. This does not correspond to the first and second level of an analytical hierarchy but an arithmetical one. It corresponds to BCSS-Σ_1 in Ziegler (2007). Please note that this is only a lower bound on the computing capability.

If one wants the analog output if the computation stops and just the information that is does not stop otherwise, it uses a singularity to know whether the computation stops and if it is the case, then it starts the computation normally to get the result in finite time.

***For example think about the so-called blue-shift effect and how it might be countered (Németi and Dávid, 2006, Sub. 5.3.1).

If one would like to have the limit of say first cell of the array if any, not only does the shrinking might turn the scale and the distance between encoding signals to zero but moreover, if the BM-computation uses bigger and bigger real numbers, its accumulated re-scaling also turns them to zero. So that one would have to find how to generate the value in a way that does not lead to zero divided by zero and neither prevents the shrinking.

This has been achieved to get internal multiplication out of linear BSS: three of the four real numbers encoding signals are constant outside of the singularity and the singularity happens at the correct position for the last signal. A general scheme still have to be found.

4.2.2. *Technical issues.*

In each active region, the configuration only undergoes a regular scaling. Up to scaling, the active regions perfectly assemble. The computation is exact and the real values are preserved even though the encoding can be split between various active regions (retrieving the value is not obvious).

The difference from the discrete case is that singularities are already used for multiplication. These are handled with the simple trace scheme (iii of Def. 4).

With the structure, both type of singularity should be distinguished particularly because the singularity for multiplication could happen exactly on toggle so that with the simple trace scheme, the structure would be damaged. In this case, one extra frozen meta-signal must exist to encode the singularity as well as a toggle for the structure. The case iv –which is more general– is used. Since the speed of toggle is greater than the any of the other signal present for the multiplication, it gets straight into the singularity without getting involved in any collision and thus distinguishes between the singularities.

If there are infinitely many multiplications, then the singularity is of second order. This is not a problem for the structure nor the definition of space-time diagrams. Rule iv of Def. 3 only asks for infinitely many collisions or signals which is also ensured by the accumulating multiplications.

5. Nested singularities

Previous Section ends with a second order singularity. It is possible to built higher order singularities.

Section 2.2 explains how to shrink any space-time diagram that has no singularity. It is quite easy to set signals to automatically start a singularity (in the spirit of what is done in Durand-Lose (2006b)) and according to the result to start another. There can be finitely or countably many isolated singularities (there is a rational location in each open).

The interesting part is when singularities are nested inside another one. In Sect. 4.2.2, it is explained how, with the conditional traces scheme (iv of Def. 4), to handle sub-singularities. The second order structure works because it is built on top of the previous one (after renaming the meta-signals to avoid any conflict). So that the outer one can handle the inner ones while the latter proceed on their own.

It is thus natural to reiterate the construction. This can be done safely a finite number of times (more would yield an infinite number of signals). Inside any structure, it is only possible to start a lesser level structure so that the first one is the topmost and that the number of nested levels inside is bounded by construction of the ESM.

For discrete computations, in the setting presented in Subsect. 3.2, singularities are only used to decide and leave no trace (ii of Def. 4). But since a singularity could happen on a toggle of a higher level structure, for the same reason as before, the conditional traces scheme has to be used. For analog computations, this scheme is already used. It is thus the natural scheme to use in both cases.

In the discrete case, like for the SAD computers of Hogarth (1994, 2004), each order brings the capability to decide an extra level of the arithmetical hierarchy by deciding an extra alternating quantifier (on $\mathbb{N}$). As shown, first order decides Σ_1^0. If nth order singularity allow to decided Σ_n^0, then it can be used as oracles inside the top level of a $n+1$th order singularity.

In the analog case, each level of singularity identically allows to decide an extra alternation of quantifiers (on $\mathbb{N}$) and to climb the BSS arithmetical hierarchy. This is not a quantification on $\mathbb{R}$, this is not an algebraic hierarchy.

6. Conclusion

Abstract geometrical computation provides a setting where Black hole-like computation can be implemented: a finite portion is infinitely accelerated while a computation can continue outside and get a finite insight on what happens inside. For SAD computers, the (nested) black holes have to be found while in AGC the computation construct them (signals are the "fabric" of space and time) but the maximum level of nesting is limited by the construction of the ESM.

In our constructions, there are two levels of space-time: one absolute of the SM and one due to the embedding inside a shrinking structure; singularities are created by the fabric of signals. The following have been proved.

Theorem 8 *With proper handling of accumulation, for any level of the arithmetic hierarchy, a rational ESM can be built to decide it and for any level of the BSS arithmetic hierarchy, an ESM can be built to decide it.*

These are just decisions, but they can be used as sub-computations in any computation.

In the discrete case, the ESM and space-time-diagrams remain rational (irrational coordinate can only be generated as singularity location, but, by construction, the singularities built happen only at rational position).

A general construction that would allow all orders with the same ESM is still lacking. One important achievement would be to provide a system for transfinite order singularity together with the possibility to nest inside recursive computations. Considering spaces with more dimensions, could help getting hyper-arithmetic (Bournez, 1999a,b). But our scheme is only in one dimension, signals with dimension 1 or more could be used to shrunk in such spaces.

Comparing to infinite time Turing machine (Hamkins and Lewis, 2000; Hamkins, 2002, 2007), the content of the tape is lost since it all accumulates in one point. So a way to preserve the resulting tape and a limit mechanism still have to be found to relate to infinite time Turing machines and recursive analysis (Weihrauch, 2000). The same problem arise with the analog counterpart; providing limits would link to the hierarchy of Chadzelek and Hotz (1999).

References

Lenore Blum, Michael Shub, and Steve Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bull. Amer. Math. Soc.*, 21(1):1–46, 1989.

Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. *Complexity and real computation.* Springer, New York, 1998.

Olivier Bournez. Achilles and the Tortoise climbing up the hyper-arithmetical hierarchy. *Theoret. Comp. Sci.*, 210(1):21–71, 1999a.

Olivier Bournez. Some bounds on the computational power of piecewise constant derivative systems. *Theory Comput. Syst.*, 32(1):35–67, 1999b.

Olivier Bournez and Emmanuel Hainry. On the computational capabilities of several models. In Jérôme Durand-Lose and Maurice Margenstern, editors, *Machines, Computations, and Universality, MCU '07*, volume 4664 of *LNCS*, pages 12–23. Springer, 2007.

Thomas Chadzelek and Günter Hotz. Analytic machines. *Theoret. Comp. Sci.*, 219(1-2):151–167, 1999.

B. Jack Copeland. Hypercomputation. *Minds & Machines*, 12(4):461–502, 2002.

Jérôme Durand-Lose. Abstract geometrical computation 1: embedding black hole computations with rational numbers. *Fund. Inf.*, 74(4):491–510, 2006a.

Jérôme Durand-Lose. Forcasting black holes in abstract geometrical computation is highly unpredictable. In J.-Y. Cai, S. B. Cooper, and A. Li, editors, *Theory and Appliacations of Models of Computations (TAMC '06)*, number 3959 in LNCS, pages 644–653. Springer, 2006b.

Jérôme Durand-Lose. Abstract geometrical computation and the linear Blum, Shub and Smale model. In S.B. Cooper, B. Löwe, and A. Sorbi, editors, *Computation and Logic in the Real World, 3rd Conf. Computability in Europe (CiE '07)*, number 4497 in LNCS, pages 238–247. Springer, 2007.

Jérôme Durand-Lose. Abstract geometrical computation with accumulations: beyond the Blum, Shub and Smale model. In Arnold Beckmann, Costas Dimitracopoulos, and Benedikt Löwe, editors, *Logic adn Theory of Algorithms, CiE 2008 (abstracts and extended abstracts of unpublished papers)*, pages 107–116. University of Athens, 2008a.

Jérôme Durand-Lose. The signal point of view: from cellular automata to signal machines. In Bruno Durand, editor, *Journées Automates cellulaires (JAC '08)*, pages 238–249, 2008b. URL `http://www.lif.univ-mrs.fr/jac/`.

Gábor Etesi and Istvan Németi. Non-Turing computations via Malament-Hogarth space-times. *Int. J. Theor. Phys.*, 41(2):341–370, 2002. gr-qc/0104023.

Joel David Hamkins. Infinite time Turing machines: supertask computation. *Minds & Machines*, 12(4):521–539, 2002. arXiv:math.LO/0212047.

Joel David Hamkins. A survey of infinite time Turing machines. In J. Durand-Lose and M. Margenstern, editors, *Machines, Computations and Universality (MCA '07)*, number 4664 in LNCS, pages 62–71. Springer, 2007.

Joel David Hamkins and Andy Lewis. Infinite time Turing machines. *J. Symb. Log.*, 65(2):567–604, 2000. arXiv:math.LO/9808093.

Mark L. Hogarth. Deciding arithmetic using SAD computers. *Brit. J. Philos. Sci.*, 55:681–691, 2004.

Mark L. Hogarth. Non-Turing computers and non-Turing computability. In *Biennial Meeting of the Philosophy of Science Association*, pages 126–138, 1994.

Ulrich Huckenbeck. Euclidian geometry in terms of automata theory. *Theoret. Comp. Sci.*, 68(1):71–87, 1989.

Ulrich Huckenbeck. A result about the power of geometric oracle machines. *Theoret. Comp. Sci.*, 88(2):231–251, 1991.

G. Jacopini and G. Sontacchi. Reversible parallel computation: an evolving space-model. *Theoret. Comp. Sci.*, 73(1):1–46, 1990.

Akitoshi Kawamura. Type-2 computability and Moore's recursive functions. *Electr. Notes Theor. Comput. Sci.*, 120:83–95, 2005.

Jerzy Mycka. Infinite limits and $\mathbb{R}$-recursive functions. *Acta Cybern.*, 16 (1):83–91, 2003a.

Jerzy Mycka. μ-recursion and infinite limits. *Theoret. Comp. Sci.*, 302(1-3): 123–133, 2003b.

Jerzy Mycka. Analog computation beyond the Turing limit. *Appl. Math. Comput.*, 178(1):103–117, 2006.

Jerzy Mycka and José Félix Costa. A new conceptual framework for analog computation. *Theoret. Comp. Sci.*, 374(1-3):277–290, 2007.

István Németi and Hajnal Andréka. Can general relativistic computers break the Turing barrier? In Arnold Beckmann, Ulrich Berger, Benedikt Löwe, and John V. Tucker, editors, *Logical Approaches to Computational Barriers, 2nd Conf. on Computability in Europe, CiE '06*, volume 3988 of *LNCS*, pages 398–412. Springer, 2006.

István Németi and Gyula Dávid. Relativistic computers and the Turing barrier. *Appl. Math. Comput.*, 178(1):118–142, 2006.

Klaus Weihrauch. *Introduction to computable analysis.* Texts in Theoretical computer science. Springer, Berlin, 2000.

Philip D. Welch. The extentent of computation in malament-hogarth space-time. *Brit. J. Philos. Sci.*, 2006. to appear.

Martin Ziegler. (Short) Survey of real hypercomputation. In S. Barry Cooper, Benedikt Löwe, and Andrea Sorbi, editors, *Computation and Logic in the Real World, 3rd Conf. Computability in Europe, CiE '07*, volume 4497 of *LNCS*, pages 809–824. Springer, 2007.

Information processing with structured excitable medium[*]

J. Gorecki[1,2], J. N. Gorecka[3] and Y. Igarashi [1]

[1] Institute of Physical Chemistry, Polish Academy of Science,
Kasprzaka 44/52, 01-224 Warsaw, Poland

[2] Faculty of Mathematics and Natural Sciences,
Cardinal Stefan Wyszynski University,
ul. Dewajtis 5, 01-815 Warsaw, Poland

[3] Institute of Physics, Polish Academy of Sciences,
Al. Lotnikow 36/42, 02-668 Warsaw, Poland

July 22, 2008

Abstract

There are many ways in which a nonlinear chemical medium can be used for information processing. Here we are concerned with an excitable medium and the straightforward method of information coding: a single excitation pulse represents a bit of information and a group of excitations forms a message. On the basis of such assumptions information can be coded or in the number of pulses or in the times between subsequent excitations.

The properties of excitable medium provide us with a pleasant environment for information processing. Pulses of excitation appear as the result of external stimuli and they propagate in a homogeneous medium with a constant velocity and a stationary shape. This is achieved by dissipating medium energy.

Our attention is focused on a quite specific type of nonhomogeneous medium that has an intentionally introduced geometrical structure of regions characterized by different excitability levels. In information processing applications the geometry plays equally important

[*] Paper still under refereeing due to late submission.

role as the dynamics of the medium and allows one to construct devices that perform complex signal processing operations even for a relatively simple kinetics of reactions involved. The ideas of information processing with structured excitable medium are tested in numerical simulations based on simple reaction-diffusion models and in experiments with Bielousov-Zhabotinsky reaction. Considering a chemical signal diode as an example we demonstrate a kind of balance between the geometry and the chemical kinetics. A diode action can be observed for a wide class of reactions if a complex geometry of excitable and non-excitable areas is fixed. On the other hand, the geometrical construction of the diode can be much simplified, but at the cost of restricted class of reaction parameters allowed.

We present chemical realizations of simple information processing devices like logical gates, signal comparers or memory cells and we show that by combining these devices as building blocks the medium can perform complex signal processing operations like for example counting of arriving excitations. We also discuss a few ideas for programming a structured information processing medium with excitation pulses.

Keywords: excitability, information processing, Oregonator, BZ-reaaction

1 Introduction

The history of human civilization can be divided into a number of periods when a given type of material like stone, bronze or iron played a dominant role in technology. During such periods a large proportion of tools, weapons or other goods were manufactured using the dominating material. Nowadays, thanks to the progress in material science and engineering, the materials used are optimized for the final product application. The authors think that the domination similar to that mentioned above can be observed today in the field of information processing. The silicon technology and the conventional von Neumann computer architecture [1], based on the clock controlling the sequence of executed instructions and the data flow, has undisputable leadership in the field. However, the situation can change in the future, because we discover more and more examples of unconventional information processing devices, frequently inspired by biology [2]. These examples play an important, inspiring role suggesting new solutions, realizations or algorithms [3] going beyond the conventional computer science. It may be expected that the new, unconventional strategies of information processing will be very important for the future applications especially in the fields of sensing, visual recognition and orientation in space, high density data storage or artificial intelligence. These applications match well with the common expectations on the development of nanorobotics, where the proper miniaturizations of tracks responsible for sensing the environment and communication are the crucial factors for potential applications.

Among many branches of unconventional computation one can recognize the field called reaction-diffusion computing [4]. The name comes after the mathematical description of time evolution of the computing medium. In the most standard case we consider information processing with a spatially distributed chemical reactor. Its state at a given point is defined by local concentrations of reagents involved. The interactions between reactions proceeding in different places occur via diffusion of molecules, so the evolution equations include both reaction and diffusive terms. The practical applicability of a uniform medium resting in its stable state for information processing seems to be null. In order to do something useful we should consider a nonequilibrium medium with the right balance between the characteristic time scales for chemical reactions and spatial relaxation. It has been found that the medium with Bielousov-Zhabotynsky (BZ) reaction [5] is one of interesting candidates for the investigation, because its state is characterized by color and so it can be easily observed and recorded. The first successful applications of reaction-diffusion computing were image processing with an oscillatory chemical reaction [6] and methods of finding the shortest path in a maze using propagating spikes [7]. In both cases a membrane filled with reagents of Bielousov-Zhabotynsky reaction was used a computing medium.

Uniform medium was used for image processing operations and the image was introduced as the space dependent phase of chemical oscillations, fixed by the initial illumination level generated by the projected image. Illumination level transforms directly into the state of reagents or, from the mathematical point of view, into the chemical oscillator phase. The effects of image processing can be observed because the phase of BZ- reaction is related to the color of the solution. The fact that in BZ reaction rapid changes of color are separated by long periods when colors slowly evolve are especially useful for image processing. Chemical systems can perform such operations as contour smoothing, contrast enhancement or detail removing as illustrated in Fig. 1. Of course the medium processes an image in a highly parallel way, transforming all points of the image at the same real time [8].

Excitability is a wide spread type of nonequilibrium behavior [5, 9] observed in numerous chemical systems including BZ reaction [10], CO oxidation on Pt [11], or combustion of gases [12]. The excitable systems share common properties. They have a stable stationary state (the rest state) they remain in when they are not perturbed. The rest state is stable so its small perturbation uniformly decays in time. However, if a perturbation is sufficiently large then system response is strongly nonlinear and it is accompanied by large changes in variables that characterize the system. The response corresponds to an excitation peak. After an excitation the system is refractory, which means that it takes a certain recovery time before another excitation can take place. If an excitable medium is spatially distributed then an excitation can propagate in space in a form of a pulse. Unlike mechanical waves that dissipate the initial energy and finally decay, traveling pulses use the energy of medium to propagate and finally dissipate it. In a typical homogeneous excitable medium an excitation pulse converges to this stationary shape after some time and the shape is independent of the initial condition. The interests in applications of excitable medium for information processing is motivated by similarities with signaling in neural systems based on excitable biochemical reactions [13]. In particular, spatially distributed excitable medium if not perturbed resides in its rest state and shows no activity. When activated, pulses of excitation appear, move and interact. Stronger excitations bring more pulses or generate signals with higher frequencies.

In the following we focus our attention on a quite specific type of nonhomogeneous excitable medium that has an intentionally introduced geometrical structure of regions characterized by different excitability levels. Here yet again one can see analogy with the structure of nerve system composed of cells linked by nerve channels. Historically, one of the first applications of structured chemical medium for information processing was the solution of the problem of finding the shortest path in a labyrinth [7]. The idea is illustrated in Fig. 2. The labyrinth is build of excitable channels (dark) separated by non-excitable medium (light - illuminated) that does not allow for interactions between pulses propagating in different channels. Using an excitable system it is easy to verify if the distance between two selected points (A and B) in a labyrinth is shorter than an assumed value d. In order to do it one can excite the medium at the point A and observe if excitation appears at the point B before time t_{AB}. The excitation spreads out through the labyrinth, separates at the junctions and pulses enter all possible paths. During the time evolution pulses of excitation can collide and annihilate, but the one that propagates along the shortest path has always unexcited medium in front. The speed s of a pulse propagating through medium in the stationary state can be regarded as constant if the labyrinth is large enough and if the influence of corners can be neglected when compared with the time of propagation in the straight channels. If the length of the shortest path linking two points A,B in a labyrinth is shorter than d then the time between the pulse initialization at A and its arrival at B should be shorter than d/s. The algorithm described above is called the "prairie fire" algorithm and it

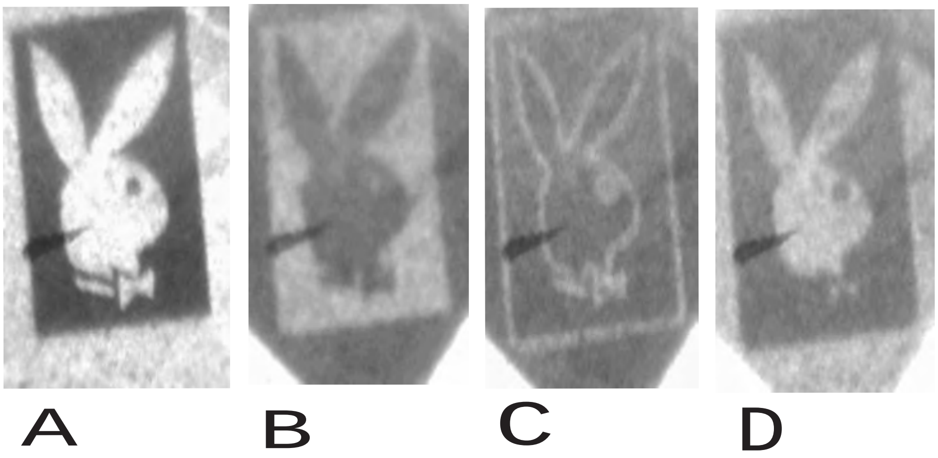


Figure 1: Image transformations during the time evolution of oscillating medium with Ru-catalyzed BZ reaction. (A) - initially projected image, (B-D) three snapshots showing image processing during a typical time evolution [14].

is automatically executed by an excitable medium. It finds the length of the shortest path in a highly parallel manner scanning all possible routes at the same time. It is quite remarkable that the time required for verification if there is a path shorter than d does not depend on the the complexity of labyrinth structure, but only on the distance between the considered points.

In the following we present a few examples illustrating that in information processing applications the medium geometry plays equally important role as its dynamics. As the result one can construct devices that perform complex signal processing operations even for a relatively simple kinetics of reactions involved. We discuss chemical realizations of simple information processing systems like signal diodes, logical gates or memory cells and we show that by combining these devices as building blocks the medium can perform complex signal processing operations like for example counting of excitations arriving to a selected point.

2 The principles of information processing with structured excitable medium

In this section we discuss a number of properties of excitable chemical systems that seem to be useful for processing information coded in excitation pulses. The most straightforward method of information coding is related to

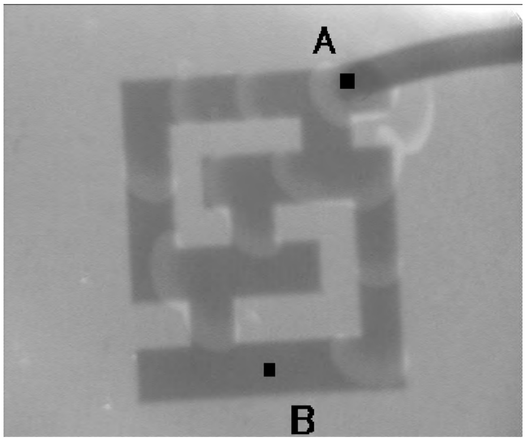


Figure 2: Pulses of excitation propagating in a labyrinth observed in an experiment with Ru-catalyzed BZ-reaction. The excitable areas are dark, the non-excitable light. The source of a train of pulses (a tip of a silver wire) is placed at the point A

the presence of pulses: a single excitation pulse represents a bit of information and a group of excitations forms a message. Information coded in the presence of pulses can be processed via interaction of pulses with the medium of via pulse-to-pulse interaction. Our discussion is mainly based on results of numerical simulations of medium time evolution. Numerical simulations of excitable chemical systems play an important role, because the models are relatively simple and easy to compute, but still accurate enough to give a correct qualitative agreement with experimental results. In the case of Ru-catalyzed BZ reaction simulations can be done with different variants of the Oregonator model [15, 16, 17, 18]. For example, for three variable model, used to obtain most of the results quoted below, the evolution equations are:

$$\varepsilon_1 \frac{\partial u}{\partial t} = u(1-u) - w(u-q) + D_u \nabla^2 u$$

$$\frac{\partial v}{\partial t} = u - v$$

$$\varepsilon_2 \frac{\partial w}{\partial t} = \phi + fv - w(u+q) + D_w \nabla^2 w$$

Here u, v and w denote dimensionless concentrations of the following reagents: $HBrO_2$, $Ru(4,4'\text{-dm-bpy})_3^{3+}$ and Br^-, respectively. In the considered system of equations u is an activator and v is an inhibitor. In the equations given above the units of space and time are dimensionless and they have been chosen to scale the reaction rates to a simple, rate constant free form. The diffusion of ruthenium catalytic complex is neglected because it is much smaller than those of the other reagents. Reaction dynamics is described by a set of parameters: q, f, q, ε_1, ε_2 and ϕ. Among them ϕ represents the rate of bromide production caused by illumination and it is proportional to the applied light intensity. Illumination is an inhibiting factor of photosensitive BZ reactions so, by adjusting the proper ϕ as a function of space variables, we can easily define regions with the required excitability level, like for example excitable stripes surrounded by non-excitable neighborhood.

Let us consider a medium composed of excitable areas, where pulse propagation is stable at the cost of medium energy dissipation and non-excitable regions where, due to different reaction regime, activations are rapidly dumped. We assume unperturbed diffusion of mobile reagents between areas of both types, so a pulse propagating in the excitable region can penetrate into a non-excitable part and disappears after some distance. The properties of excitable medium offer a number of generic properties that seem to be useful for information processing.

One of them is angle dependent penetration of non-excitable barriers. Let us consider two pieces of the active medium separated by a non-excitable stripe. It can be easily shown that the maximum width of non-excitable stripe for which a pulse propagating on one side of a stripe generates an excitation on the other side depends on the angle between the wave vector of the pulse and the normal to the stripe. As it is expected a pulse with wave vector perpendicular to the stripe can excite a medium separated by a wider stripe than a pulse that propagates along the stripe [19]. Thus the width of a gap separating excitable regions can be adjusted such that it is transparent to perpendicular pulses, but not to those that propagate along the gap. This property is frequently used in chemical realizations of information processing devices. For example, such gaps appear in a junction of two excitable channels that automatically excludes interactions of pulses arriving from one input channel on the second one. If we just join to input channels I1 and I2 into one output O (see Fig. 3A) then a pulse arriving form one of input channels would separate at the junction and resulting excitations enter both the output channel and the other input channel perturbing arriving signals. However, if we consider input channels separated by non-excitable gaps as illustrated in Fig. 3B then the propagation of pulses from the inputs to the output is not perturbed, but there is no interference between inputs because a signal arriving from one input channel always propagate parallel to the gap separating it from the other input channel.

A typical excitable dynamics is characterized by a refractory period - an interval of time after excitation when the system evolves towards the stable

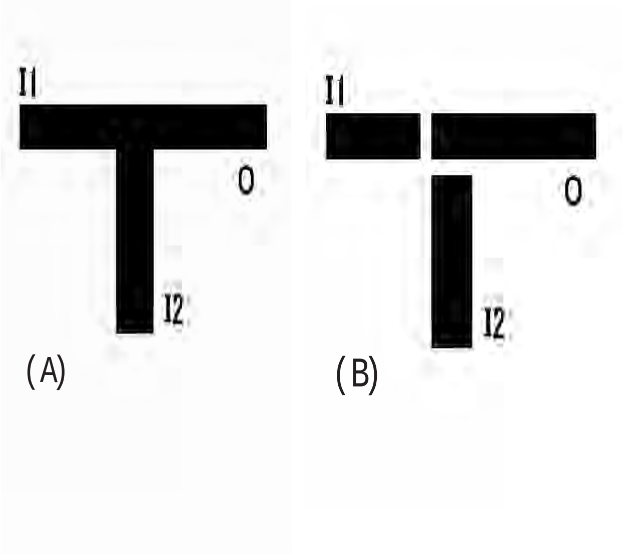


Figure 3: The junction of two input channels I1, I2 into a single output O (a logical OR gate). The black regions are excitable, the white space is non-excitable. (A) - simple junction, (B) - a junction in which interactions between inputs are excluded.

state and its repeated excitation is difficult due to a large concentration of inhibitor. For propagating pulses it means that the region behind a pulse is in the refractory regime characterized by a large amplitude of inhibitor and it cannot be re-excited. One of the consequences of a long refractory period is annihilation of colliding counterpropagating pulses. Another interesting example of behavior resulting from refractory region behind a pulse can be observed in a cross-shaped structure build of excitable channels, separated by gaps penetrable for perpendicular pulses [20, 21] shown in Fig. 4. The answer of cross-shaped junction to a pair pulses arriving from two perpendicular directions has been studied as a function of the time difference between pulses. Of course if the time difference is large pulses propagate independently along one line. If the time difference is small then the cross-junction acts like the AND gate and the output excitation appears in one of the corner areas. However, for a certain time difference the first arriving pulse is able to redirect the second and force it to follow. The effect is related with uncompleted relaxation of the central area of the junction at the moment when the second pulse arrives. Pulse redirection seems to be an interesting effect from the point of programming with excitation pulses, but in practice it requites a high precision in selecting the right time difference.

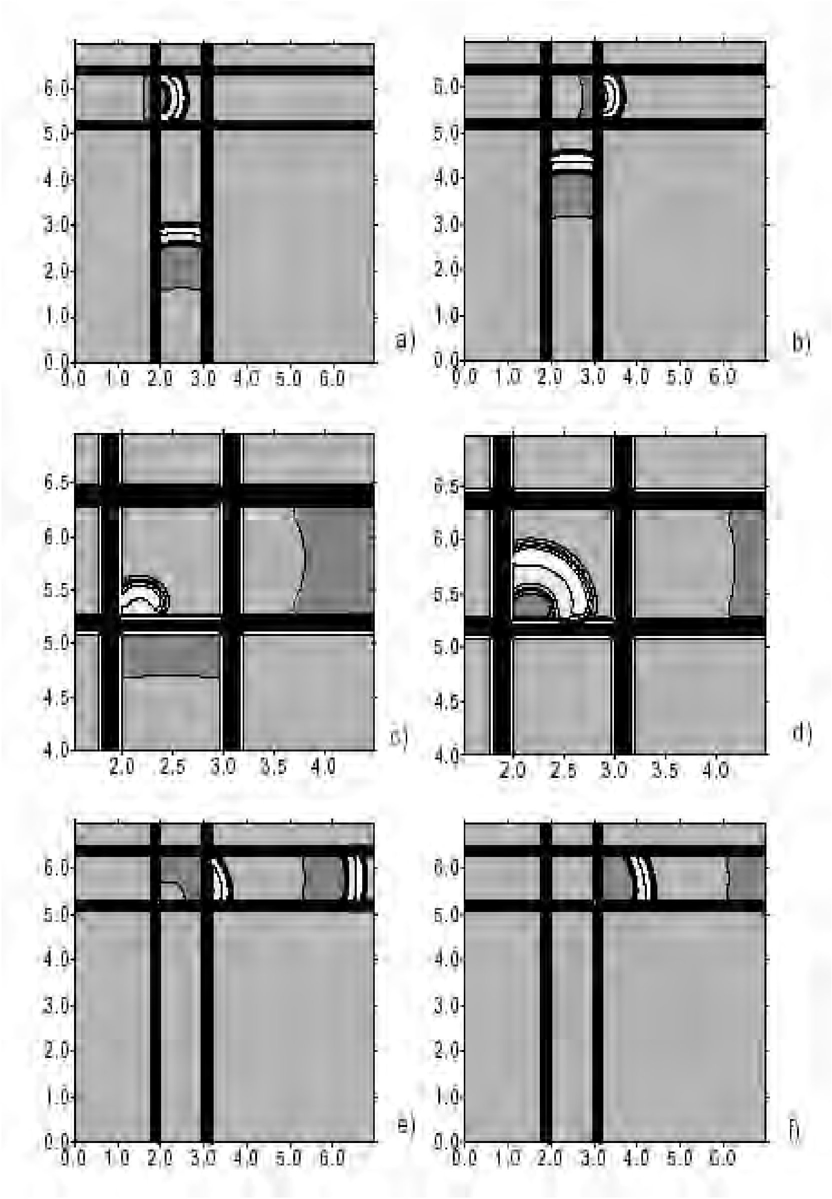


Figure 4: The distribution of excitable and non-excitable regions in a cross-shaped junction. Here the excitable regions are gray and the non-excitable black. Consecutive figures illustrate an interesting type of time evolution caused by interaction of pulses. Two central figures are enlarged in order to show how uncompleted relaxation influences the shape of the second arriving pulse [20].

Another interesting and useful property of structured excitable medium is related to its answer to combined excitations. The perturbations introduced by multiple excitations combine and generate a stronger perturbation than this resulting from a single spike. For example we can consider two parallel channels separated by a gap nonpenetrable for a single propagating excitation. It can be shown [22] that the width of such gap can be selected to allow for cross excitation of one channel by two counterpropagating spikes in the other one. This feature of structured excitable medium allows for easy realization of the AND gate. This result can be generalized to multiple excitations. It can be shown that for properly adjusted strengths of input signals one can get excitation of the output channel only when the required number of excitations arrive from inputs at the same time [23]. The geometry of such chemical artificial neuron is inspired by a the structure of biological neuron [24]. Topologically in both structures we find a number of narrow input channels (dendrites) that transmit excitations to the larger cell body connected with output channels. One of the studied realizations is illustrated on Fig. 5. Here the output channel is perpendicular to figure plane. Another geometry of an artificial chemical neuron with input and output channels on the same plane has been discussed in [23]. In the artificial chemical neuron, like in real neurons, dendrites (input channels 1-4) transmit weak signals which are added together through the processes of spatial and temporal integration inside the cell body. If the aggregate excitation is larger than the threshold value the cell body gets excited. The results illustrated on Fig 5B come from numerical simulations based on the Oregonator model where the illumination of the surrounding non-excitable region was considered as the control parameter. It can be seen that by applying the proper illumination level the structure shown in Fig. 5 can work as a four input McCulloch-Pitts neuron with any required integral threshold.

The amplitude of a pulse propagating in an excitable channel (and thus the strength of excitation) depends on channel width. In the case of wide channels this amplitude is defined by the dynamics of chemical processes only and it is close to that for plane pulses. In narrow channels the diffusion of activator towards non-excitable neighborhood plays the dominant role. If the channel is very narrow then the amplitude of activator may drop below the critical value and the propagating pulse dies. In the studied neuron the amplitudes of spikes in input channels have been adjusted by channel width and by the illumination of surrounding non-excitable medium.

Non-excitable barriers in structured medium can play more complex role in information processing then that described above. The problem of barrier crossing by a periodic train of pulses can be seen as an excitation via a periodic perturbation of the medium [25]. The answer of the medium is quite

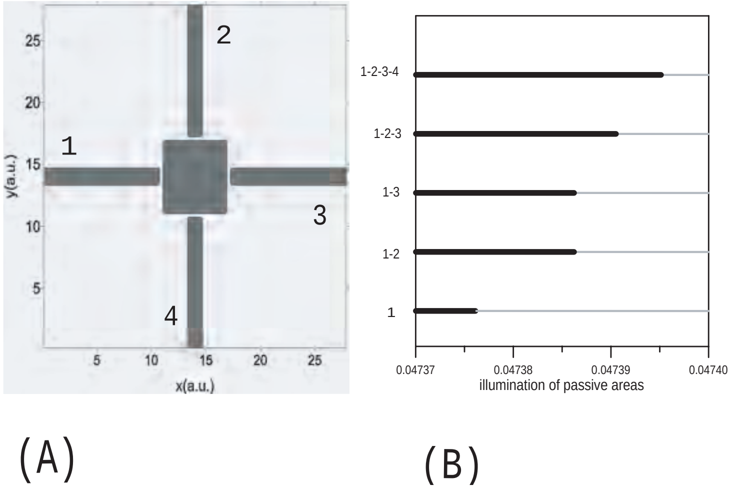


Figure 5: Artificial chemical neuron constructed with structured excitable medium. (A) - the geometry of excitable (dark) and non-excitable areas (gray); (B) - the response of the neuron to different types of combined excitations as the function of illumination of non-excitable regions. The numbers given on the left list the excited channels. The illuminations for which the neuron body gets excited are marked by a thick line.

characteristic. The firing number as a function of perturbation strength has a devil-staircase-like form. In the case of barrier crossing the strength of excitation behind a barrier generated by an arriving pulse depends on the character of non-excitable medium, on barrier width and on the frequency of the incoming signal (usually, due to uncompleted relaxation of the medium the amplitude of spikes decreases with frequency). A typical, complex frequency transformation after barrier crossing is illustrated in Fig. 6. Experimental and numerical studies on firing number of a transmitted signal have been published [26, 27, 28, 29]. It is interesting that the shape of regions characterized by the same firing number in the space of two parameters: barrier width and signal frequency is not generic and depends on type of excitable medium. For example in the case of FitzHugh - Nagumo dynamics trains of pulses with small periods can cross wider barriers than trains characterized by low frequency, for the Rovinsky - Zhabotinsky model the dependence is reversed.

High sensitivity of transmitted signal frequency with respect to the parameters of input perturbation can be used to construct a sensor estimating the distance separating the source of periodic excitations from the observer. The idea is illustrated in Fig. 7 [30]. The sensor is build with a number of similar excitable signal channels (in Fig.7A they are numbered 1-4) that are

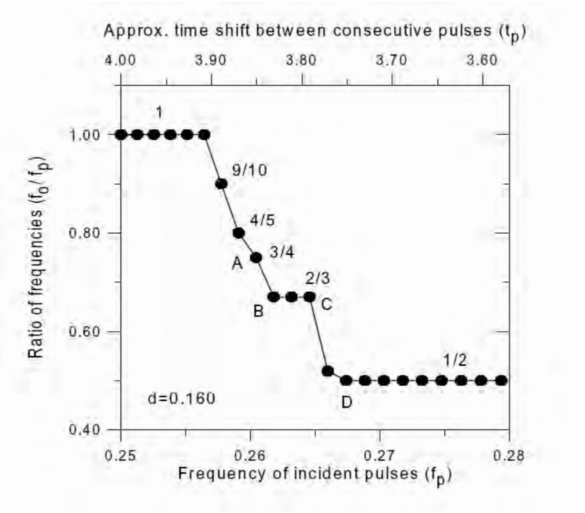


Figure 6: A typical dependence of firing number as a function of frequency of arriving pulses. The ratio between frequency of output signal (f_o) after crossing a non-excitable gap separating excitable channel and the input one (f_p). Results obtained for the FitzHugh - Nagumo model.

wide enough to ensure a stable propagation of pulses. These sensor channels are separated one from another by parallel non-excitable gaps that do not allow for the interference between pulses propagating in the neighboring channels. They are also separated from the active medium M by a non excitable sensor gap G. The width of this gap is crucial for sensor sensitivity. If the gap is too wide then no excitation of the medium M can generate a pulse in the sensor channels. If the gap is narrow then any excitation in front of a sensor channel can pass G and create a pulse in the channel so signals in every sensor channel are identical. The width of the gap should be selected such that the firing number (defined as the ratio between the number of pulses that crossed the gap G to the number of pulses of excitation that were created in the medium M) depends on the wave vector characterizing a pulse at the gap in front of channel. If the source of periodic excitations S is close to the array of sensor channels then the wave vectors characterizing excitations in front of various channels are different. Thus, different frequencies of excitations in various channels are expected. On the other hand, if the source of excitations is far away from the gap G then the wave vectors in front of different channels should be almost identical and so the frequencies of excitations in sensor channels would not differ. Therefore, the system shown in Fig. 7A can sense the distance separating it from the source of excitations.

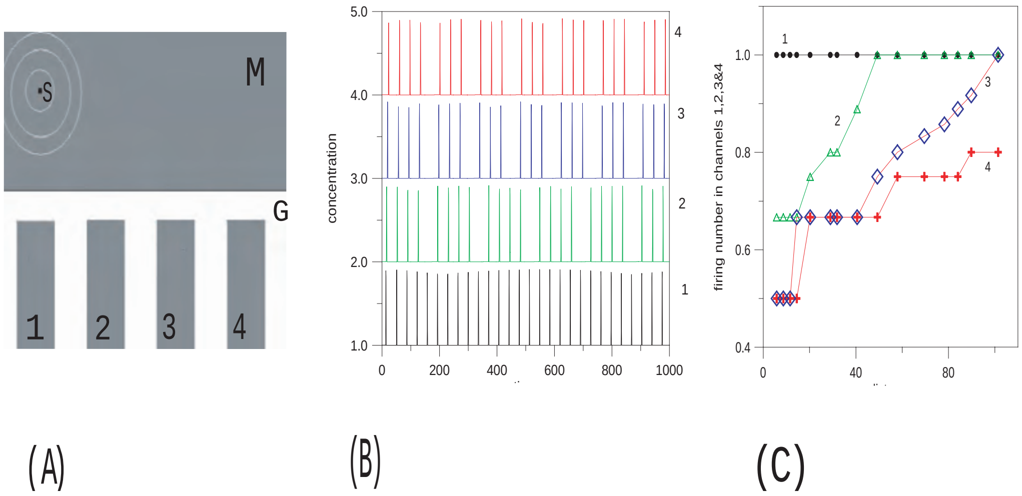


Figure 7: Schematic illustration of a device sensing the distance separating it from a source of periodic excitations S. (A) - the geometry of excitable (dark) and non-excitable (light) regions. (B) - the signals in sensor channels for the source close to the sensor. (C) - the firing numbers as function of distance separating sensor and the source [30].

If this distance is small the firing numbers in neighboring sensor channels are different and these differences decrease when the source of excitations moves away. The results of simulations (Figs. 7B,C) and experiments [30] confirm it.

The sensitivity to temporal changes in medium properties is another feature of an excitable system that can have important potential applications in information processing. It has been recently observed [31] that survival of a propagating excitation pulse in a medium with decreasing excitability level depends on the rate of changes. For example let us assume that the illumination of the system described by Oregonator model increases linearly in time from ϕ_1 to ϕ_2. Moreover, for both values ϕ_1 and ϕ_2 stable pulses of excitation can propagate in the system. It has been shown [31] that there is a range of values of (ϕ_1, ϕ_2) for which pulse propagation is sensitive to the rate of illumination changes. If illumination slowly increases in time from ϕ_1 to ϕ_2 than a pulse initiated in the system characterized by ϕ_1 is continuously propagating with a gradual adjustment of its shape to time dependent illumination. However, when the changes are rapid then excitation disappears.

3 The chemical signal diode

A chemical signal diode is a device that forces unidirectional pulse propagation. The recent studies on the diode nicely illustrate an interplay between the geometrical complexity of the medium and chemical dynamics. The same function can be achieved for relatively wide class of chemical dynamics parameters and a complex geometry or for much simpler geometry, but within a narrow margin for system dynamics where the diode operates. The classical construction of a chemical signal diode is shown in Fig. 8A [32]. The black areas are excitable and the white parts are non-excitable. The asymmetry required for unidirectional signal transmission is introduced by a non-symmetrical junction formed by a rectangular excitable channel on one side (Y) and a triangular excitable channel on the other (X). The distance between the top of the triangle and the side of the rectangle is selected such that a pulse of excitation propagating in the rectangular channel and terminating at its left end gives sufficiently strong perturbation to excite the triangle tip and thus the excitation is transmitted from Y to X. For the same distance the excitation of rectangular channel by a pulse moving towards the top of X channel is too small to excite the channel Y because the amplitude of the pulse moving in the triangular part decreases as the result of diffusion to the neighboring non-excitable areas. Therefore, the propagation of a pulse moving right is terminated. The construction of such chemical diode is generic (it can be adopted for any excitable chemical system), but the excitability of the medium is described by a non-trivial function of two spatial variables.

An alternative construction of a chemical signal diode has been suggested in [33]. Using numerical simulations based on the Oregonator model it has been shown that the parameters of illumination described by a triangular function (cf. Fig. 8B) can be adjusted such that a pulse of excitation is transmitted in one direction only. If the catalyst is immobilized then excitation pulses that enter illuminated area from the highly illuminated end (right end in Fig. 8B) are not transmitted, so the triangular illumination profile makes a signal diode. Quite recently [34] this result has been simplified and it has been shown that an illumination profile, composed of two regions characterized by different, but uniform illumination levels (Fig. 8C) can also work as a signal diode. Finally it has been also shown that if the excitable input channels are not symmetrical then a single, non-excitable barrier (Fig. 8D) with precisely adjusted parameters also works as a diode. Various designs illustrated on Fig. 8 have different features. The diode shown in Fig. 8A is very robust and in order to reverse the direction of signal transmission one should change the positions of channels. The other designs allow for easier

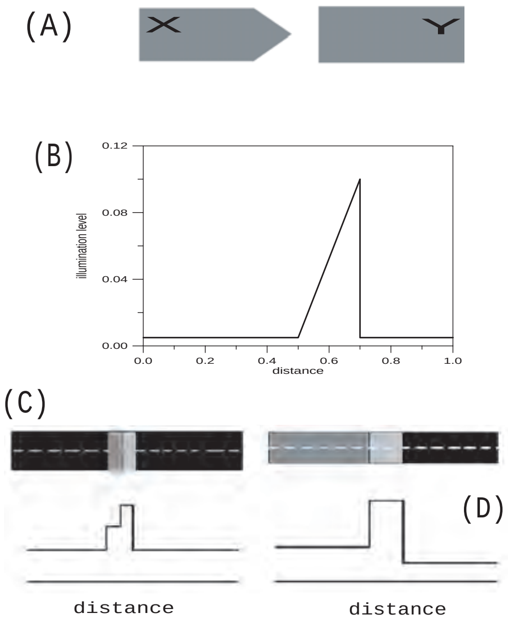


Figure 8: Different chemical realizations of a signal diode with BZ- reaction inhibited by light: (A) - the classical design based on nonsymmetrical junction [32], (B) - a triangular profile of illumination [33], (C) - the illumination profile in a diode composed of two non-excitable barriers [34], (D) - the illumination profile in a single barrier diode with nonsymmetrical excitable inputs.

control. Fig. 9 shows the functional diagram of the device illustrated in Fig. 8D as the function of illumination of the left input channel and illumination of the gap. The illumination of the right output channel is fixed ($\phi_{right} = 0.007$) and the width of the gap is constant (3.75). The open circles indicate transmission in both directions and the closed ones show the pairs of (ϕ_{left}, ϕ_{gap}) for which the gap is stops all arriving pulses. The triangles mark unidirectional transmission in the direction indicated by the triangle tip. It is interesting that both possible directions of diode transmission are present on the functional diagram and quite small changes in illuminations (for example from ($\phi_{left} = 0.0098$, $\phi_{gap} = 0.04494$) to ($\phi_{left} = 0.01$, $\phi_{gap} = 0.04490$) can change the transmission direction.

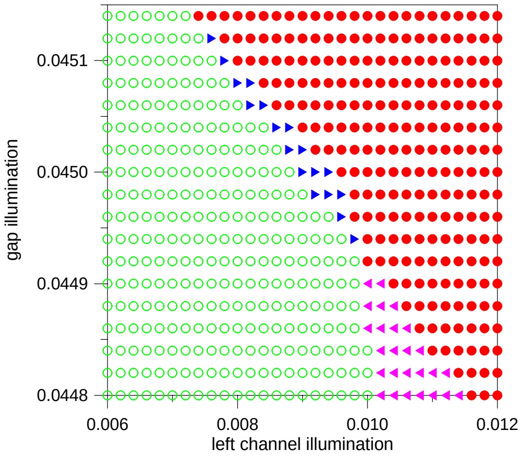


Figure 9: The functional diagram of the device illustrated in Fig. 8D as the function of illumination of the left input channel and the illumination of the gap. Open and closed circles mark bi-directional transmission and nonpenetrable gaps. Triangles show the conditions where the gap works as a signal diode with transmission direction indicated by the tip.

4 Chemical memory and its applications

The devices discussed in the previous section can be classified as instant machines [8] capable of performing just the task they have been designed for. A memory where information coded in excitation pulses can be written-in, kept, read-out and, if necessary, erased significantly increases information processing potential of structured excitable medium. Moreover, due to the fact that the state of memory can be changed by a spike, the memory allows for programming with excitation pulses. One possible realization of a chemical memory is based on observation that a pulse of excitation can can rotate on a ring-shaped excitable area as long as the reactants are supplied and the products removed [35, 36, 37]. Therefore, a ring with a number of spikes rotating on it can be regarded as a loaded memory cell. Such memory can be erased by counterpropagating pulses. The idea of memory with loading pulses rotating in one direction and erasing pulses in another has been discussed in [38]. Our later studies [39] were oriented on increase of reliability of ring shaped memory. A large ring can be used to memorize a large amount of information because it has many states corresponding to different numbers of rotating pulses. However, in such case loading the memory with subsequent pulses may fail because the input can be blocked by the refractory tail left by one of already rotating pulses. Therefore, a memory capable of storing just a single bit seems to be more reliable. Such memory has two states: loaded when there is a rotating pulse the ring and erased when the ring is in the rest state. An example of ring memory is shown in Fig. 10. The red areas define the memory ring composed of two L-shaped excitable areas and Z-shaped erasing channel in the mid of the ring. The blue regions are non-excitable. The excitable areas are separated by gaps selected such that a pulse of excitation propagating perpendicularly to the gap excites the active area on the other site of the gap, but the gap is impenetrable for pulses propagating parallel to the gap. Such choice breaks the symmetry and only excitations rotating counterclockwise are stable. The rotating pulse does not affect the erasing channel because it always propagates parallel to it. The erasing excitation is generated in the center of Z-shaped area and it splits into two erasing pulses (Fig. 10B). These pulses can cross the gaps separating the erasing channel from the memory ring and create a pair of pulses rotating clockwise (Figs. 10 C/D). A spike that propagates clockwise on the memory ring is not stable because it is not able to cross any of the gaps and dies. Therefore, if the memory has not been loaded then an erasing excitation does not load it. On the other hand, if the memory is loaded then pulses rotating clockwise resulting from excitation of the erasing channel annihilate with the loading pulse and the memory is erased (Fig. 10D). The idea of using two places where erasing pulses can enter memory ring is used to ensure that at least one of those places is fully relaxed and so one of erasing pulses can always enter the ring. We have confirmed reliable work of the ring memory in a number of computer experiments with random positions of rotating and erasing pulses. A few experiments gave qualitative agreement with simulations: the loaded memory ring preserved information for few minutes and it was erased after excitation of the erasing channel.

Using the structured excitable medium and simple circiuits described above one can construct devices, that perform more complex signal processing operations. As an example we present a simple chemical realization of excitation counter. The system returns their number in any chosen positional representation [22]. Such counter can be assembled from single digit counters. The construction of a single digit counter depends on the representation used. Here, as an example, we consider the positional representation

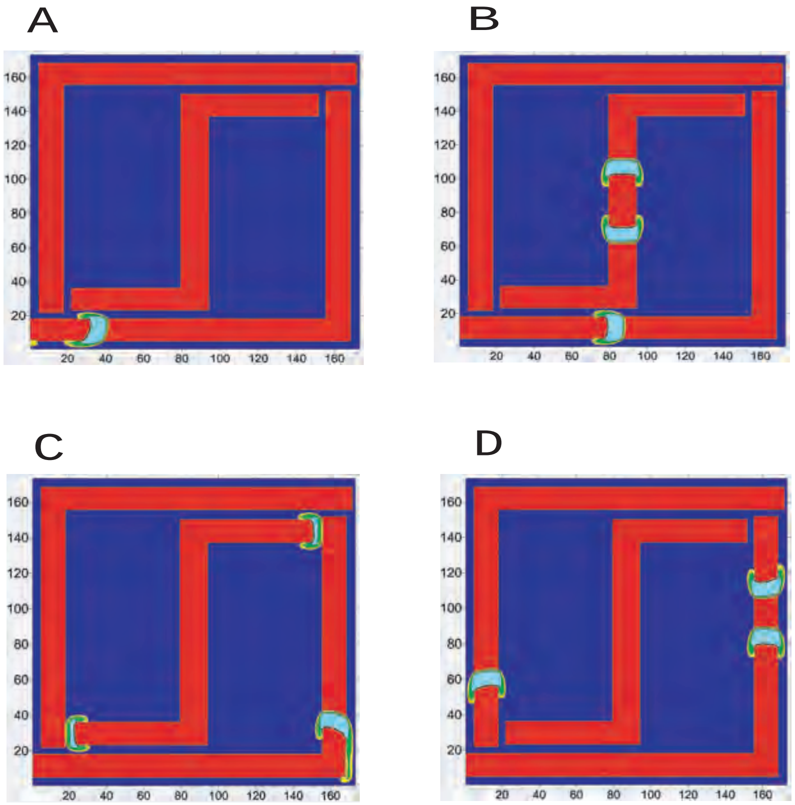


Figure 10: Four snapshots illustrating simulations of memory erasing by an excitation coming from erasing channel. The memory ring is formed by two L-shaped excitable channels, the Z-shaped erasing channel is inside the ring.

with the base 3. The geometry of a single digit counter is schematically shown in Fig. 11. Its main elements are two memory cells M_1 and M_2 and two coincidence detectors C_1 and C_2. At the beginning let us assume that none of the memory cells is loaded. When the first pulse arrives through the input channel I_0, it splits at all junctions and excitations enter segments B_0, B_1 and B_2. The pulse that has propagated through B_0 loads the memory cell M_1. The pulses that have propagated through B_1 and B_2 die at the bottom diodes of segments C_1 and C_2 respectively. Thus, the first input pulse loads the memory M_1 and does not change the state of M_2. When M_1 is loaded then pulses of excitation are periodically sent to segments B_0 and C_1 via the bottom channel. Now let us consider what happen when the second pulse arrives. It does not pass through B_0 because it annihilates with the pulses arriving from the memory M_1. The excitations generated by the second pulse can enter B_1 and B_2. The excitation that propagated through B_2 dies at the bottom diode of the segment C_2. The pulse that has propagated through B_1 enters C_1, annihilates with a pulse from memory M_1 and activates the coincidence detector. The output pulse from the coincidence detector loads the memory M_2. Therefore, after the second input pulse both memories M_1 and M_2 are loaded. If the third pulse arrives the segments B_0 and B_1 are blocked by spikes sent from the memory rings. The generated excitation can enter channel B_2 and its collision with a pulse coming from the memory cell M_2 activates the output channel of C_2. The output signal is directed to the counter of responsible for the digit at next position (I1) and it is also used to erase all memory cells. Thus after the third pulse both memory cells M1 and M2 are erased. The counter shown in Fig. 11 returns a digit in a representation with the base 3 : here 0 is represented by the $(M_1, M_2) = (0, 0)$, 1 by $(1, 0)$, 2 by $(1, 1)$ and the next pulse changes the state of memory cell into $(M_1, M_2) = (0, 0)$. Of course, using $n - 1$ memory cells in a single digit counters we can represent digits of the system with base n. A cascade of single digit counters gives a positional representation of the number of arriving pulses.

Discussion

We have presented a number of examples that should convince the reader that structured excitable medium can be successfully used for processing information coded in excitation pulses. All the considered systems transform information in an unconventional (non-von Neumann) way, i.e. without an external clock or synchronizing signal that controls the sequence of operations. On the other hand, in many cases the right timing of performed operations is hidden in the geometrical distribution and sizes of excitable regions. The described devices can be used as building blocks for more complex systems that process signals formed of excitation pulses. Some of them

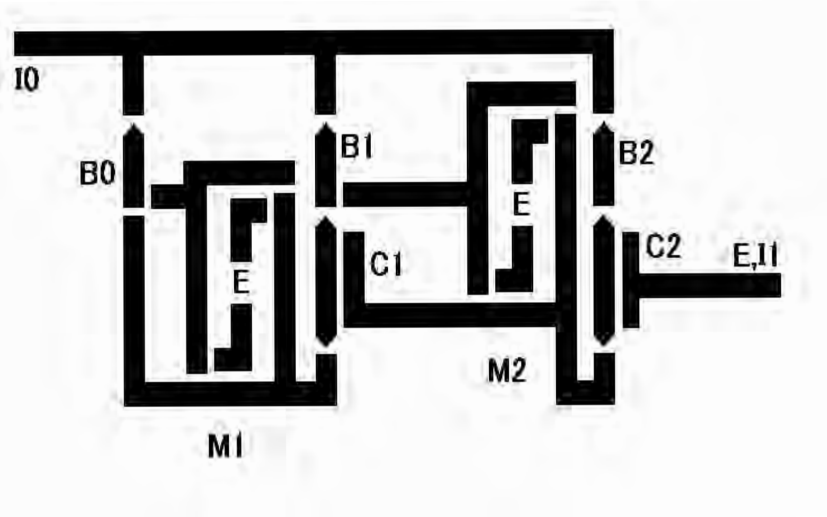


Figure 11: The counter of excitation pulses that arrive at the input I0. Figure shows the geometry of excitable channels (black) in a single digit counter for the positional representation with the base 3.

like the memory cell or pulse counter can be controlled with spikes. Therefore, there is a room for programming and learning. However, for the further development of information processing with structured excitable medium we have to solve two basic problems.

First is the problem of creating a structure that performs required functions. It seems that the answer to this problem is suggested by a biological analog of structured signal processing medium - a brain. The structure should appear as a thermodynamically stable phase under carefully selected nonequilibrium conditions. We know that much simpler but yet interesting structures come as metastable phases in multicomponent systems. For example, the diamond structure in oil-water-surfactant system, that spontaneously appears at certain thermodynamic conditions has a form of centers linked with the four nearest neighbors. If the reactants corresponding for excitability are soluble in water, but not in oil then the water rich phase forms the structure of excitable channels and processing elements just as the result of thermodynamic conditions. Within a certain range of parameters, such structure is thermodynamically stable. This means that the network has auto-repair ability and robustness against unexpected destruction. Moreover, the structure is three dimensional, what allows for higher density of processing elements than that obtained with classical two-dimensional techniques like for example lithography.

The second is the problem of continuous operation. The energy needed for pulse propagation is comes from the energy of reactants in the medium. The reagents should be continuously delivered in order to ensure unperturbed operation. In the case of very simple systems reagents can be delivered by hydrodynamic flows caused by pressure gradients. We doubt if such simple transport is sufficient for complex operations. And yet again the living organisms have solved this problem by creating a parallel structure that keeps the signal processing components at steady nonequilibrium conditions.

References

[1] R. P. Feynman, R. W. Allen, T. Heywould, *Feynman Lectures on Computation*, New York: Perseus Books, (2000).

[2] C.S. Calude and G. Paun *Computing with cells and atoms*, Taylor and Francis, London and New York, (2002).

[3] A. Tero, R. Kobayashi and T. Nakagaki, Physarum solver: A biologically inspired method of road-network navigation, *Physica* A 363: 115?119 (2006).

[4] A. Adamatzky, B. De Lacy Costello and T. Asai. (2005). *Reaction-Diffusion Computers*, UK: Elsevier Science.

[5] R. Kapral and K. Showalter,*Chemical Waves and Patterns*, Kluwer Academic, Dordrecht , (1995).

[6] L. Kuhnert, K.I. Agladze and V.I. Krinsky, Image processing using light-sensitive chemical waves, *Nature* 337:244-247 (1989).

[7] O. Steinbock, A. Toth, and K. Showalter, Navigating complex labyrinths - optimal paths from chemical waves, *Science* 267:868-871, (1995). .

[8] N. G. Rambidi and A. V. Maximychev, Towards a Biomolecular Computer. Information Processing Capabilities of Biomolecular Nonlinear Dynamic Media, *BioSystems* 41:195-211 (1997).

[9] Y. Kuramoto, *Chemical Oscillations, Waves, and Turbulence*, Springer-Verlag, Berlin, (1984).

[10] A. S. Mikhailov and K. Showalter, Control of waves, patterns and turbulence in chemical systems, *Phys. Rep.* 425: 79-194, (2006).

[11] K. Krischer, M. Eiswirth, and G.J. Ertl, Oscillatory CO oxidation on Pt(110): modelling of temporal self-organization, *J.Chem. Phys.* 96:9161-9172 (1992).

[12] J. Gorecki, A.L. Kawczynski, Molecular dynamics simulations of a thermochemical system in bistable and excitable regimes, *J Phys. Chem.* 100:19371-19379 (1996).

[13] J. D. Murray, (1989), *Mathematical Biology*, Springer-Verlag, Berlin.

[14] J. Szymanski , private information (2008).

[15] R.J. Field, and R.M. Noyes, Oscillations in chemical systems. IV. Limit cycle behavior in a model of a real chemical reaction, *J. Chem. Phys.* 60:1877-1884 (1974).

[16] V. Gaspar, G. Bazsa and M.T. Beck, The influence of visible light on the Belousov–Zhabotinskii oscillating reactions applying different catalysts, *Z. Phys. Chem.(Leipzig)* 264:43-48 (1983).

[17] H.J. Krug, L. Pohlmann and L. Kuhnert, Analysis of the modified complete Oregonator accounting for oxygen sensitivity and photosensitivity of Belousov–Zhabotinskii systems, *J. Phys. Chem.* 94:4862-4866 (1990).

[18] T. Amemiya, T. Ohmori, T. Yamaguchi. An Oregonator-class model for photoinduced Behavior in the $\mathrm{Ru(bpy)}_3^{2+}$–Catalyzed Belousov–Zhabotinsky reaction, *J. Phys. Chem. A.* 104: 336-344 (2000).

[19] I. Motoike and K. Yoshikawa, Information Operations with an Excitable Field, *Phys. Rev. E* 59:5354-5360 (1999).

[20] J. Sielewiesiuk and J. Gorecki, Chemical impulses in the perpendicular junction of two channels, *Acta Phys. Pol. B* 32:1589-1603 (2001).

[21] J. Sielewiesiuk and J. Gorecki, Logical functions of a cross junction of excitable chemical media, *J. Phys. Chem. A* 105:8189 - 8195, (2001),

[22] J. Gorecki, K. Yoshikawa and Y. Igarashi, On chemical reactors that can count, *J. Phys. Chem. A* 107:1664-1669, (2003).

[23] J. Gorecka and J. Gorecki, Multiargument logical operations performed with excitable chemical medium, *J. Chem. Phys.* 124: 084101 (2006).

[24] H. Haken, *Brain Dynamics* Springer Series in Synergetics, Springer-Verlag Berlin and Heidelberg, (2002).

[25] M. Dolnik, I. Finkeova, I. Schreiber, and M. Marek, Dynamics of forced excitable and oscillatory chemical-reaction systems, *J. Phys. Chem.* 93:2764-2774 (1989); I. Finkeova, M. Dolnik, B. Hrudka, and M. Marek, Excitable chemical reaction systems in a continuous stirred tank reactor, *J. Phys. Chem.* 94:4110-4115 (1990); M. Dolnik and M. Marek, Phase excitation curves in the model of forced excitable reaction system. *J. Phys. Chem.* 95:7267-7272 (1991); M. Dolnik, M. Marek, and I.R. Epstein, Resonances in periodically forced excitable systems, *J. Phys. Chem.* 96:3218-3224 (1992).

[26] K. Suzuki, T. Yoshinobu and H. Iwasaki, Unidirectional propagation of chemical waves through microgaps between zones with different excitability, *J. Phys. Chem. A* 104:6602-6608 (2000).

[27] J. Sielewiesiuk and J. Gorecki, On complex transformations of chemical signals passing through a passive barrier, *Phys. Rev. E* 66, 016212 (2002).; J. Sielewiesiuk and J. Gorecki, Passive barrier as a transformer of chemical signal frequency, *J. Phys. Chem. A* 106:4068-4076 (2002).

[28] A. F. Taylor, G. R. Armstrong, N. Goodchild, S. K. Scott, Propagation of chemical waves across inexcitable gaps, *Phys. Chem. Chem. Phys.*5:3928-3932 (2003).

[29] G. R. Armstrong, A. F. Taylor, S. K. Scott, V. Gaspar, Modelling wave propagation across a series of gaps, *Phys. Chem. Chem. Phys.* 6:4677-4681 (2004).

[30] J. Gorecki, J. N. Gorecka, K. Yoshikawa, Y. Igarashi and H. Nagahara, Sensing the distance to a source of periodic oscillations in a nonlinear chemical medium with the output information coded in frequency of excitation pulses, *Phys. Rev. E* 72:046201 (2005).

[31] M. Tanaka, H. Nagahara, H. Kitahata, V. Krinsky, K. Agladze and K. Yoshikawa, Survival versus collapse: Abrupt drop of excitability kills the traveling pulse, while gradual change results in adaptation, *Phys. Rev. E* 76:016205(2007),

[32] K. Agladze, R. R. Aliev, T. Yamaguchi and K. Yoshikawa. Chemical diode, *J. Phys. Chem.* 100:.13895-13897 (1996).

[33] A. Toth, D. Horvath and K. Yoshikawa, Unidirectional wave propagation in one spatial dimension, *Chem. Phys. Lett.* 345: 471-474 (2001).

[34] . J. Gorecka, J. Gorecki and Y. Igarashi, One dimensional chemical signal diode constructed with two non-excitable barriers, *Journal of Physical Chemistry A* 111:885–889(2007).

[35] A. Lázár, Z. Noszticzius, H. -D. Försterling, and Z. Nagy-Ungvárai. Chemical pulses in modified membranes I. Developing the technique, *Physica* D, 84:112-119 (1995).; A. Volford , P. L. Simon, H. Farkas, Z. Noszticzius, Rotating chemical waves: theory and experiments, *Physica* A 274:30-49. (1999).

[36] Y. Nagai, H. Gonzalez, A. Shrier and L. Glass,. Paroxysmal Starting and Stopping of Circulatong Pulses in Excitable Media, *Phys. Rev. Lett.* 84:4248-4251(2000).

[37] Z. Noszticzuis, W. Horsthemke, W. D. McCormick, H. L. Swinney and W. Y. Tam. Sustained chemical pulses in an annular gel reactor: a chemical pinwheel, *Nature* 329:619?620 (1987).

[38] I. N. Motoike, K. Yoshikawa, Y. Iguchi and S. Nakata, Real–Time Memory on an Excitable Field, *Phys. Rev. E* 63:036220 (2001).

[39] J. Gorecki and J.N. Gorecka, On mathematical description of information processing in chemical systems, in *Mathematical Approach to Nonlinear Phenomena; Modeling, Analysis and Simulations, GAKUTO International Series, Mathematical Sciences and Applications.* pp 73-90 vol. 23, ISBN 4762504327 (2005).

Computational bounds on polynomial differential equations

Daniel S. Graça [a,b] , Jorge Buescu [c,d] ,
Manuel L. Campagnolo [e,b] ,

[a] *DM/FCT da Universidade do Algarve, Portugal*
[b] *SQIG/Instituto de Telecomunicações, Lisboa, Portugal*
[c] *DM/FCUL, University of Lisbon, Portugal*
[d] *CMAF, Lisbon, Portugal*
[e] *DM/ISA, Technical University of Lisbon, Portugal*

Abstract

In this paper we study from a computational perspective some properties of the solutions of polynomial ordinary differential equations.

We consider elementary (in the sense of Analysis) discrete-time dynamical systems satisfying certain criteria of robustness. We show that those systems can be simulated with elementary and robust continuous-time dynamical systems which can be expanded into fully polynomial ordinary differential equations in $\mathbb{Q}[\pi]$. This sets a computational lower bound on polynomial ODEs since the former class is large enough to include the dynamics of arbitrary Turing machines.

We also apply the previous methods to show that the problem of determining whether the maximal interval of definition of an initial-value problem defined with polynomial ODEs is bounded or not is in general undecidable, even if the parameters of the system are computable and comparable and if the degree of the corresponding polynomial is at most 56.

Combined with earlier results on the computability of solutions of polynomial ODEs, one can conclude that there is from a computational point of view a close connection between these systems and Turing machines.

Email addresses: `dgraca@ualg.pt` (Daniel S. Graça), `jbuescu@ptmat.fc.ul.pt` (Jorge Buescu), `mlc@math.isa.utl.pt` (Manuel L. Campagnolo).

Preprint submitted to Elsevier *22 July 2008*

1 Introduction

Differential equations are a powerful tool to model a diversity of phenomena in fields ranging from basic natural sciences like physics, chemistry or biology to social sciences or economics. Among these, initial value problems (IVPs) of the form $x' = f(t,x)$, with $x(t_0) = x_0$, where f is a vector field and t is the independent variable, play a predominant role. In this paper we consider the large class of polynomial IVPs (PIVPs for short) in which f is a vector of polynomials. Many well known models, like the Lorenz equations in meteorology, the Lotka-Volterra equations for predator-prey systems, or Van der Pol's equation in electronics [1] fall into this category. In Section 2 we show that in fact all the elementary functions of Analysis are solutions of PIVPs. This is a stronger version of the well established fact that all elementary functions are differentially algebraic [2]. It is also worth noticing that the solutions of PIVPs are precisely the set of functions definable with Shannon's General Purpose Analog Computer (GPAC) [3] as proved in [4].

While the qualitative behavior of linear systems (i. e. where f is linear) and planar systems (where $f : \mathbb{R}^3 \to \mathbb{R}^2$) is completely understood [5], it is not known for all but a few cases how to predict the behavior of the solutions of general PIVPs from the expression of f, which is the reason why many fundamental questions about PIVPs (e.g. Hilbert's 16th problem) are still open.

Since most nonlinear differential equations cannot be solved exactly, one has to resort to numerical methods to obtain approximate solutions. This leads to a range of questions about computational properties of PIVPs. In particular, one can ask if PIVPs have computable approximations, if the domain of the solution is computable, or even if deciding whether the maximal interval of existence is bounded is computable. Such questions have been answered for analytic IVPs (where f is analytic) in [6]. In Section 2 we point out that the results in [6] imply that the domain of existence of PIVP functions (i.e. solutions of PIVPs) is in general recursively enumerable and that the solution is computable on its domain. This last result sets an upper bound on the computability of PIVP functions since it ensures that as long as f is polynomial and computable they can be arbitrarily approximated wherever they are defined.

To obtain computational lower bounds for PIVPs, one can show that any computable function can be approximated by some PIVP function. In [7] it was proved that under a simple (and unbounded) encoding in $\mathbb{N}^3$, the evolution of Turing machines can be simulated with PIVPs. In this paper, we extend that result and show that any computable discrete dynamical system on $\mathbb{N}^m$ which admits a robust extension (to be defined) can be simulated with a PIVP.

The iteration of maps with IVPs is not new and can be found, for instance, in [8], [9]. However, those constructions are in some sense not satisfactory since they involve functions with some degree of discreteness (e.g. functions which are not analytic or even have discontinuous derivatives) which can be used to build "exact clocks" that simulate the discrete steps of the iteration.

In Sections 3 and 4 we state and prove the main result of the paper. We show that given a map ω on $\mathbb{N}^m$, one can construct a PIVP with coefficients in $\mathbb{Q}[\pi]$ that simulates the iteration of ω as long as ω is extendable to a "robust" map Ω on $\mathbb{R}^m$, in a sense to be defined in Section 3, and Ω is composition of polynomial and PIVP functions with parameters in $\mathbb{Q}[\pi]$. The simulation is robust, which is a necessity for our construction, but is also a natural requirement for a continuous-time physical system described by an IVP. The constructions in Section 4 will also provide the necessary tools to address the issues discussed in the remainder of the paper.

Finally, in Section 5, we review and extend some undecidability results on PIVPs. Our results in [7] imply that reachability for PIVPs is undecidable, i.e., given a PIVP and some open set in phase space, there is no algorithm to decide if the solution of the PIVP crosses the open set. This contrasts with the decidability of the reachability for linear differential equations [10]. In [11] we showed that the boundedness of the domain of existence for PVIPs is undecidable as long as f is polynomial of sufficiently high degree and computable. At first sight, this result might seem trivial, since one can easily construct simple PIVPs which, upon varying one parameter, exhibit a critical value where the solution is bounded on the left of this parameter value and unbounded on the right. For instance, the PIVP $x' = \alpha(x^2 - 1)t$, $x(0) = 3$ has a maximal interval which is bounded for $\alpha > 0$ and unbounded if $\alpha \leq 0$. Since we cannot compare exactly two arbitrary computable reals [12] the boundedness problem for the PIVP above is undecidable. However, in Section 5 we show that if we consider that all input parameters are "comparable", the boundedness problem remains undecidable. We also prove the claim in [11] that those undecidability results hold for PIVPs where the degree of the polynomial is less or equal to 56.

2 The GPAC, Polynomial Differential Equations, and Computable Analysis

In this section we introduce some useful definitions and results that will later be used in this paper.

Definition 1 *Let $I \subseteq \mathbb{R}$ be a non-empty open interval and let $t_0 \in I$. We say that $g : I \to \mathbb{R}$ is a PIVP function on I if it is a component of the solution of* *the initial-value problem*

$$x' = p(t, x), \quad x(t_0) = x_0 \tag{1}$$

where p is a vector of polynomials and $t_0 \in I$. We say that g is a PIVP function with parameters in $S \subseteq \mathbb{R}$ if the coefficients of p in (1), t_0, and the components of x_0 belong to S.

Similarly we say that a function $g : I \subseteq \mathbb{R} \to \mathbb{R}^k$ is a vector PIVP function if each component of g is a PIVP function.

Example 2 *The following are examples of PIVP functions with parameters in $\mathbb{Z}$: the exponential function e^x, the trigonometric functions* $\cos$, $\sin$ *[7], the inverse function $x \mapsto 1/x$ (solution of $y' = -y^2$; on $(0, +\infty)$ it can be obtained by setting the initial condition $y(1) = 1$).*

The PIVP functions are also closed under the following operations (as far as we know, these properties have only been reported in the literature for the broader case of differentially algebraic functions):

(1) Field operations $+, -, \times, /$. For instance, if $f, g : I \to \mathbb{R}$, where $I \subseteq \mathbb{R}$ is an open interval, are PIVP functions, then so is $f + g$ in I. In fact, if f, g are the first components of the solutions of the (vector) PIVPs

$$\begin{cases} x' = p(t, x) \\ x(t_0) = x_0 \end{cases} \quad \text{and} \quad \begin{cases} y' = q(t, y) \\ y(t_0) = y_0 \end{cases}$$

respectively then, since $f'(t) + g'(t) = p_1(t, x) + q_1(t, x)$, where $p_1(t, x)$ and $q_1(t, x)$ are the first components of $p(t, x)$ and $q(t, x)$ respectively, $f + g$ is the last component of the solution of the PIVP

$$\begin{cases} x' = p(t, x) \\ y' = q(t, y) \\ z' = p_1(t, x) + q_1(t, y) \end{cases} \qquad \begin{cases} x(t_0) = x_0 \\ y(t_0) = y_0 \\ z(t_0) = x_{0,1} + y_{0,1} \end{cases}$$

where $x_{0,1}$ and $y_{0,1}$ are the first components of vectors x_0 and y_0, respectively. Similar proofs apply for the operations $-, \times, /$. It should be noted that the quotient f/g is a PIVP function in intervals which do not contain zeros of g, and that the PIVP which generates f/g is well-defined in such intervals. For instance $\tan(= \frac{\sin}{\cos})$ is a PIVP function on $(-\pi/2, \pi/2)$.

(2) Composition. If $f : I \to \mathbb{R}$, $g : J \to \mathbb{R}$, where $I, J \subseteq \mathbb{R}$ are open intervals and $f(I) \subseteq J$, are PIVP functions, then so is $g \circ f$ on I. To see this,

suppose that f, g are the first components of the solutions of the PIVPs

$$\begin{cases} x' = p(t,x) \\ x(t_0) = x_0 \end{cases} \quad \text{and} \quad \begin{cases} y' = q(t,y) \\ y(t_1) = y_0 \end{cases} \tag{2}$$

respectively, where $t_0 \in I$ and $t_1 \in J$ (no connection is assumed between these values). Then, since $(g \circ f)'(t) = g'(f(t)).f'(t)$, we construct a system that computes $f'(t)$ (just copy the left system of (2) and note that $f'(t) = p_1(t,x)$), and another that computes $g'(f(t))$ (now pick the right system of (2); the first component will give $g'(t)$, so we have to substitute the variable t by $f(t) = x_1$ so that this component yields $g'(f(t))$), obtaining the following PIVP, where $g \circ f$ is the component z_1:

$$\begin{cases} x' = p(t,x) \\ z_1' = q_1(x_1, z)p_1(t,x) \\ \qquad\vdots \\ z_n' = q_n(x_1,z)p_1(t,x) \end{cases} \qquad \begin{cases} x(t_0) = x_0 \\ z(t_0) = f(x_0). \end{cases}$$

(3) Differentiation. If $f : I \to \mathbb{R}$, where $I \subseteq \mathbb{R}$ is an open interval, is a PIVP function, then so is $f' : I \to \mathbb{R}$. To see this, suppose that f is the first component of the solution of the PIVP

$$\begin{cases} x' = p(t,x) \\ x(t_0) = x_0. \end{cases}$$

Then

$$f'(t) = x_1''(t) = \frac{d}{dt}p_1(t,x) = \frac{\partial p_1}{\partial t} + \sum_{i=1}^{n} \frac{\partial p_1}{\partial x_i} x_i' = \frac{\partial p_1}{\partial t} + \sum_{i=1}^{n} \frac{\partial p_1}{\partial x_i} p_i(t,x)$$

which implies that f' is the last component of the solution of the PIVP

$$\begin{cases} x' = p(t,x) \\ z' = \frac{\partial p_1}{\partial t} + \sum_{i=1}^{n} \frac{\partial p_1}{\partial x_i} p_i(t,x) \end{cases} \qquad \begin{cases} x(t_0) = x_0 \\ z(t_0) = f'(t_0). \end{cases}$$

(4) Compositional inverses. If $f : I \to \mathbb{R}$, where $I \subseteq \mathbb{R}$ is an open interval, is a bijective PIVP function, then so is f^{-1}. This case will be shown in the end of this section. In particular, this result implies that log, arcsin, arccos, and arctan are also PIVP functions.

From the preceding examples, we conclude that the following corollary, where closed-form stands for the class of *elementary functions* in Analysis which, informally, correspond to the functions obtained from the rational functions, sin, cos, exp through finitely many compositions and inversions.

Corollary 3 *All closed-form functions are PIVP functions.*

When proving that some function is PIVP, we will find it most convenient to make use of ODEs not only defined with polynomials, but also with other PIVP functions. For this purpose, we have to resort to the next theorem, which can be viewed as a strengthening of the elimination theorem of Rubel and Singer for differentially algebraic functions [13] to the case of PIVPs. Its proof is given in [7] for $S = \mathbb{R}$ but applies to any subfield of $\mathbb{R}$ (a different proof is given implicitly in [14]).

Theorem 4 *Let S be a subfield of $\mathbb{R}$. Consider the IVP*

$$x' = f(t,x), \quad x(t_0) = x_0 \tag{3}$$

where $f : D \subseteq \mathbb{R}^{n+1} \to \mathbb{R}^n$, D is the domain of f, and each component of f is a composition of polynomials with coefficients in S and PIVP functions with parameters in S and $(t_0, x_0) \in D \cap S^{n+1}$. Then there exists $m \geq n$, a polynomial $p : \mathbb{R}^{m+1} \to \mathbb{R}^m$ with coefficients in S and $y_0 \in S^m$ such that the solution of (3) is given by the first n components of $y = (y_1, ..., y_m)$, where y is the solution of the PIVP

$$y' = p(t,y), \quad y(t_0) = y_0.$$

Let us now prove that the inverse function f^{-1} of a bijective PIVP function $f : I \to \mathbb{R}$, where $I \subseteq \mathbb{R}$ is an open interval, is also a PIVP function. We know that $(f^{-1})'(x) = 1/f'(f^{-1}(x))$. Then, between two consecutive (inverse images of) zeros a, b of f', with $a < b$, f^{-1} will be the solution of the IVP

$$y' = \frac{1}{f'(y)}, \quad y(f(d)) = d, \tag{4}$$

where $d \in I$ and $f(d) \in (a,b)$. Since f is a PIVP function, so is f'. Moreover $x \mapsto 1/x$ is also a PIVP function, and since PIVP functions are closed under composition, so is $x \mapsto 1/f'(x)$. Then Equation (4) and Theorem 4 ensure that $f^{-1} : (a,b) \to \mathbb{R}$ is a PIVP function.

The following result, extracted from [4], [14] shows that the General Purpose Analog Computer (GPAC), a model introduced by Shannon in 1941 [3], and later refined in [15, pp. 13-14], [4, p. 647], [14], is equivalent to PIVP functions. This result applies formally to the refined version of the GPAC presented in [4, p. 647], [14].

Proposition 5 *A function is generated by a GPAC iff it is a PIVP function.*

Therefore, all results stated in this paper for PIVP functions are also valid for the GPAC generable functions.

We now recall basic notions from computable analysis. See [16] for an up-to-date monograph on computable analysis from the computability point of view, [12] for a presentation from a complexity point of view, or [17] for a general introduction to the subject.

Definition 6 *A sequence $\{r_n\}$ of rational numbers is called a ρ-name of a real number x if there exist three functions a, b, c from $\mathbb{N}$ to $\mathbb{N}$, such that for all $n \in \mathbb{N}$, $r_n = \frac{b(n)}{c(n)+1}(-1)^{a(n)}$ and*

$$|r_n - x| \leq \frac{1}{2^n}. \tag{5}$$

In the conditions of the above definition, we say that the ρ-name $\{r_n\}$ is given as an oracle to an oracle Turing machine, if the oracle to be used is (a, b, c). The notion of the ρ-name can be extended to $\mathbb{R}^l$: a sequence $\{(r_{1n}, r_{2n}, \ldots, r_{ln})\}_{n\in\mathbb{N}}$ of rational vectors is called a ρ-name of $x = (x_1, x_2, \ldots, x_l) \in \mathbb{R}^l$ if $\{r_{jn}\}_{n\in\mathbb{N}}$ is a ρ-name of x_j, $1 \leq j \leq l$.

Definition 7 *A real number x is called computable if a, b, and c in (5) are computable (recursive) functions.*

Definition 8 *A function $f : D \subseteq \mathbb{R}^m \to \mathbb{R}^k$ is computable if there is an oracle Turing machine such that for any input $n \in \mathbb{N}$ (accuracy) and any ρ-name of $x \in E$ given as an oracle, the machine will output a rational vector r satisfying $\|r - f(x)\|_\infty \leq 2^{-n}$, where $\|(y_1, \ldots, y_l)\|_\infty = \max_{1\leq i\leq l} |y_i|$ for all $(y_1, \ldots, y_l) \in \mathbb{R}^l$.*

In particular, every rational number must be computable and it is not difficult to show that polynomials having computable coefficients are computable functions. The following is a corollary of Theorem 3.1 of [18].

Theorem 9 *Let $f : \mathbb{R} \to \mathbb{R}^m$ be a vector PIVP function with computable parameters defined on an interval (α, β). Then f is computable in (α, β).*

3 Robust Simulations of Discrete Dynamical Systems

One of the purposes of the present paper is to show that a large class of discrete systems can be simulated with vector PIVP functions. Let $\mathcal{D}$ be a discrete dynamical system (both space and time are discrete). We can associate each discrete part of the state space to an integer, so that the evolution of the system is modeled by the iteration of a map $\omega : \mathbb{N}^m \to \mathbb{N}^m$. In general, if f is a function, we denote its kth iterate by $f^{[k]}$, i.e. $f^{[0]}(x) = x$ and $f^{[k+1]} = f \circ f^{[k]}$ for all $k \in \mathbb{N}$. We now present some definitions.

Definition 10 *The map $\Omega : \mathbb{R}^m \to \mathbb{R}^m$ is a (real)* robust extension *of the map $\omega : \mathbb{N}^m \to \mathbb{N}^m$ if there exist δ_{in}, δ_{ev}, $\delta_{out} \in (0, 1/2)$ such that for all $x_0 \in \mathbb{R}^m$, $n_0 \in \mathbb{N}^m$, $\overline{\Omega} : \mathbb{R}^m \to \mathbb{R}^m$one has*

(1) $\Omega(n) = \omega(n)$ *and*
(2) $\|n_0 - x_0\|_\infty \leq \delta_{in}$ *and* $\left\|\Omega - \overline{\Omega}\right\|_\infty \leq \delta_{ev}$ *implies* $\left\|\omega(n_0) - \overline{\Omega}(x_0)\right\|_\infty \leq \delta_{out}$.

The following lemma follows easily from this definition by induction (we can "contract" δ_{out} to δ_{in} using the function σ presented in Lemma 19). For simplicity, we will usually refer to robust extensions of a map as the property described by this lemma instead of Definition 10.

Lemma 11 *If $\Omega : \mathbb{R}^m \to \mathbb{R}^m$ is a robust extension of the map $\omega : \mathbb{N}^m \to \mathbb{N}^m$, then there exist δ_{in}, δ_{ev}, $\delta_{out} \in (0, 1/2)$ such that for all $x_0 \in \mathbb{R}^m$, $n_0 \in \mathbb{N}^m$, $\overline{\Omega} : \mathbb{R}^m \to \mathbb{R}^m$one has*

(1) $\Omega(n) = \omega(n)$ *and*
(2) $\|n_0 - x_0\|_\infty \leq \delta_{in}$ *and* $\left\|\Omega - \overline{\Omega}\right\|_\infty \leq \delta_{ev}$ *implies* $\left\|\omega^{[k]}(n_0) - \overline{\Omega}^{[k]}(x_0)\right\|_\infty \leq \delta_{out}$ *for all* $k \in \mathbb{N}$.

In the continuous-time setting dynamical systems are described by ODEs instead of iteration of maps. Moreover, since time is continuous, we also allow robustness in the time instant where we read the output. Again, we could consider robustness for one time unit steps, and then generalize to give iterates for all $k \in \mathbb{N}$ as we did for robust extension. Here, for simplicity, we omit this two step procedure and present instead the following definition.

Definition 12 *Let $\phi : \mathbb{R} \to \mathbb{R}$ be the unique solution of the initial value problem*

$$x' = f(t, x), \quad x(0) = n_0.$$

We say that ϕ is a robust suspension *of the map $\omega : \mathbb{N}^m \to \mathbb{N}^m$ if there exist δ_{in}, δ_{ev}, δ_{out}, $\delta_{time} \in (0, 1/2)$, such that for all $x_0 \in \mathbb{R}^m$, $n_0 \in \mathbb{N}^m$, $k \in \mathbb{N}$, and $\overline{f} : \mathbb{R}^{m+1} \to \mathbb{R}^m$ one has that*

$$\|n_0 - x_0\|_\infty \leq \delta_{in} \text{ and } \left\|\overline{f} - f\right\|_\infty \leq \delta_{ev}$$

implies that the solution $\overline{\phi}$ of the initial-value problem

$$x' = \overline{f}(t, x), \quad x(0) = x_0$$

satisfies

$$\left\|\omega^{[k]}(n_0) - \overline{\phi}(t)\right\|_\infty \leq \delta_{out}$$

for all $t \in \mathbb{R}_0^+$ such that $|t - k| \leq \delta_{time}$.

These two definitions say that whenever we have a robust extension/suspension of a map, we can perturb the system by some amount, and still obtain a result close to the desired iterate $\omega^{[k]}(n_0)$.

We shall use $\mathbb{Q}[\pi]$, the standard algebraic ring extension of $\mathbb{Q}$ by adjoining the transcendent π, and which is the smallest ring containing $\mathbb{Q} \cup \{\pi\}$:

$$\mathbb{Q}[\pi] := \{a_n\pi^n + \ldots + a_1\pi + a_0 \in \mathbb{R} | a_0, \ldots, a_n \in \mathbb{Q}\}.$$

The following are the main results of this section, to be proved in Section 4. The next theorem shows that if the map is a composition of polynomials and PIVP functions (with parameters in $\mathbb{Q}[\pi]$), then one can constructively obtain a robust suspension of the map which is itself a PIVP function (with parameter in $\mathbb{Q}[\pi]$).

Theorem 13 *If the map $\omega : \mathbb{N}^m \to \mathbb{N}^m$ admits a robust extension $\Omega : \mathbb{R}^m \to \mathbb{R}^m$ whose components are compositions of polynomials and PIVP functions with parameters in $\mathbb{Q}[\pi]$, then ω admits a robust suspension ϕ which is a vector PIVP function with parameters in $\mathbb{Q}[\pi]$.*

The next proposition follows from the proof of Theorem 12 from [7]. There the transition of a Turing machine is coded as a map over the integers in the following manner: we code the state as an integer and, using a representation of numbers in some adequate base, we code the right part of the tape as a second integer, and the left part as a third integer. We denote that encoding by η (see [7, p. 332] for more details).

Proposition 14 *Under the encoding η, the transition function $\omega : \mathbb{N}^3 \to \mathbb{N}^3$ of a Turing machine admits a robust extension $\Omega : \mathbb{R}^3 \to \mathbb{R}^3$. Moreover Ω can be chosen to be a composition of polynomials with coefficients in $\mathbb{Q}[\pi]$ and PIVP functions with parameters in $\mathbb{Q}[\pi]$ (in particular* sin, cos *and* arctan*).*

Actually in [7] we required algebraic numbers as coefficients for the polynomials. But non-rational coefficients are only needed to perform a trigonometric interpolation, and may be well approximated by rationals for the purpose at hand. This approximation will introduce some extra error to the computation of the map, but this is a minor hinderance since the map is robust. From Theorem 13 and Proposition 14, we obtain the following result.

Corollary 15 *With the above encoding, the transition function ω of a given Turing machine admits a robust suspension ϕ. Moreover ϕ is a vector PIVP function with parameters in $\mathbb{Q}[\pi]$.*

4 Proof of Theorem 13

This proof is based on Branicky's construction [8], and many steps are similar to those presented in [7]. So, before presenting the proof of Theorem 13, we will briefly sketch this technique, that constructively shows how a map from integers to integers can be iterated with smooth ODEs. By a smooth ODE we mean an ODE

$$y' = f(t, y) \tag{6}$$

where f is of class C^k, for some $1 \leq k \leq \infty$ (but not necessarily analytic). Instead of using the original approach of Branicky, we will use the one by Campagnolo, Costa, and Moore in [9], [19], [20].

Suppose that $\omega : \mathbb{Z}^m \to \mathbb{Z}^m$ is a map. For better readability, we break down the procedure into two constructions.

Construction 16 *Consider a point $b \in \mathbb{R}$ (the* target*), some $\gamma > 0$ (the* targeting error*), and time instants t_0 (*departure time*) and t_1 (*arrival time*), with $t_1 > t_0$. Then obtain an IVP (the* targeting equation*) defined with an ODE (6), where $f : \mathbb{R}^2 \to \mathbb{R}$, such that the solution y satisfies*

$$|y(t_1) - b| < \gamma \tag{7}$$

independent of the initial condition $y(t_0) \in \mathbb{R}$.

As pointed out in [7, p. 345] this can be done by an ODE

$$y' = c(b - y)^3 \phi(t), \tag{8}$$

where $\phi : \mathbb{R} \to \mathbb{R}_0^+$ is some function satisfying $\int_{t_0}^{t_1} \phi(t)dt > 0$ and $c > 0$ is any constant which is bigger than a constant c_0 depending on γ and ϕ. Note that the only requirement for the construction to hold is that c is large enough. We refer the reader to [7, p. 345] for details.

Construction 17 *Iterate the map $\omega : \mathbb{Z}^m \to \mathbb{Z}^m$ with a smooth ODE (6).*

Let $\Omega : \mathbb{R}^m \to \mathbb{R}^m$ be an arbitrary smooth extension of ω to $\mathbb{R}$ (not necessarily robust). The iteration of ω may be performed [21, Proposition 3.4.2] by the initial-value problem

$$\begin{cases} z_1' = c_1(\Omega(r(z_2)) - z_1)^3 \theta_j(\sin 2\pi t) \\ z_2' = c_2(r(z_1) - z_2)^3 \theta_j(-\sin 2\pi t) \end{cases} \qquad \begin{cases} z_1(0) = x_0 \\ z_2(0) = x_0, \end{cases} \tag{9}$$

where $z_1(t), z_2(t) \in \mathbb{R}^m$, $\theta_j(x) = 0$ if $x \leq 0$ and $\theta_j(x) = x^j$ if $x > 0$, and $r(x)$ is a function that is a solution of an ODE and that satisfies $r(x) = i$ whenever $x \in [i - 1/4, i + 1/4]$ for all $i \in \mathbb{Z}$ (see the proof of Proposition 3.4.2 in [21] for the explicit definition of $r(x)$). Note that c_1 and c_2 depend on j and that all coefficients in (9) are in $\mathbb{Q}[\pi]$ [21]. In the remainder of this section we will show how to replace the non-analytic terms in (9) by PIVP functions with parameters in $\mathbb{Q}[\pi]$. As a result, by Theorem 4, it follows that the iteration can be performed with vector PIVP functions with parameters in $\mathbb{Q}[\pi]$.

However, if our purpose is to prove Theorem 13, we have some problems with the previous constructions:

(1) We have used the nonanalytic functions $\theta_j(x)$ and $r(x)$ which are obviously not PIVP functions. We will remove these functions using the fact that ω admits a robust extension. Therefore we have to study what happens when perturbations are allowed in (9) to prove Theorem 13.
(2) We would like to "read" the value of the iterated function not in time intervals of the form $[k, k+1/2]$ for $k \in \mathbb{N}$ as before, but rather in time intervals of the form $[k-1/4, k+1/4]$ so that we can use $\delta_{time} = 1/4$ for Theorem 13. This may be easily achieved by using a translation that adds 1/4 units of time. Because this construction is simple, in what follows, we will continue to stick to time intervals of the form $[k, k+1/2]$ in order to not overcomplicate our constructions.

In order to solve the previous problems, we need to recall the following two functions, σ and l_2, which were introduced and studied in [7].

Lemma 18 *Let $l_2 : \mathbb{R}^2 \to \mathbb{R}$ be given by $l_2(x,y) = \frac{1}{\pi}\arctan(4y(x-1/2)) + \frac{1}{2}$. Suppose also that $a \in \{0,1\}$. Then, for any $\overline{a}, y \in \mathbb{R}$ satisfying $|a-\overline{a}| \leq 1/4$ and $y > 0$,*

$$|a - l_2(\overline{a}, y)| < \frac{1}{y}.$$

Lemma 19 *Let $\sigma(x) = x - 0.2\sin(2\pi x)$ and $\varepsilon \in [0,1/2)$. Then there is some contracting factor $\lambda_\varepsilon \in (0,1)$ such that for all $n \in \mathbb{Z}$, $\forall \delta \in [-\varepsilon, \varepsilon]$, $|\sigma(n+\delta) - n| < \lambda_\varepsilon \delta$.*

Studying the perturbed targeting equation. (cf. Construction 16) Because the iterating procedure relies on the basic ODE (8), we have to study the following perturbed version of (8)

$$z' = c(\overline{b}(t) - z)^3 \phi(t) + E(t), \tag{10}$$

where $\left|\overline{b}(t) - b\right| \leq \rho$ and $|E(t)| \leq \delta$. This was done in [7], where it is shown that

$$|z(1/2) - b| < \rho + \gamma + \frac{\delta}{2}. \tag{11}$$

Removing the θ_j's from (9). We must remove the θ_j's in two places: in the function r and in the terms $\theta_j(\pm\sin 2\pi t)$. Since in (9) we are using a robust extension $\Omega : \mathbb{R}^m \to \mathbb{R}^m$ of $\omega : \mathbb{N}^m \to \mathbb{N}^m$, we no longer need the corrections performed by r. There may be a problem when Ω is a robust extension of ω with $\delta_{out} > 1/4$, but this can easily be overcome by applying the function σ l times to each component of Ω until one has that $\sigma^{[l]} \circ \Omega$ is a robust extension of ω with $\delta^{\sigma}_{in} \leq 1/4$, and use $\sigma^{[l]} \circ \Omega$ instead of Ω. So, without loss of generality, we assume that $\delta_{out} \leq 1/4$ for Ω.

On the other hand we cannot use this technique to treat the terms $\theta_j(\pm\sin 2\pi t)$. We need to substitute $\phi(t) = \theta_j(\sin 2\pi t)$ with an analytic (PIVP) function $\zeta : \mathbb{R} \to \mathbb{R}$ with the following ideal behavior:

(i) ζ is periodic with period 1;

(ii) $\zeta(t) = 0$ for $t \in [1/2, 1]$;

(iii) $\zeta(t) \geq 0$ for $t \in [0, 1/2]$ and $\int_0^{1/2} \zeta(t)dt > 0$.

Of course, conditions (ii) and (iii) are incompatible for analytic functions. Instead, we approximate ζ using a function ζ_ϵ, where $\epsilon > 0$. This function must satisfy the following conditions:

$(ii)'$ $|\zeta_\epsilon(t)| \leq \epsilon$ for $t \in [1/2, 1]$;

$(iii)'$ $\zeta_\epsilon(t) \geq 0$ for $t \in [0, 1/2]$ and $\int_0^{1/2} \zeta_\epsilon(t)dt > I > 0$, where I is independent of ϵ.

In [7] an example of a PIVP function satisfying both $(ii)'$ and $(iii)'$ is constructed (function $W_0(t, y)$ in p. 346 of that paper). Similarly, $\theta_j(-\sin 2\pi t)$ will be replaced by the PIVP function $\zeta_\epsilon(-t)$. This function is defined by means of a PIVP where all coefficients are in $\mathbb{Q}[\pi]$.

Performing Construction 17 with vector PIVP functions. We are now ready to perform a simulation of an integer map with a system similar to (9), but using only PIVP (and hence analytic) functions. Choose δ_{in}, δ_{ev}, and a targeting error $\gamma > 0$ such that

$$2\gamma + \delta_{ev}/2 \leq \delta_{in} < 1/4. \tag{12}$$

We take $\delta_{time} = 1/4$. We want to determine δ_{out} and present a system of ODEs that satisfies the conditions of Theorem 13. Consider the system of ODEs

$$\begin{cases} z_1' = c_1(\Omega \circ \sigma^{[m]}(z_2) - z_1)^3\, \zeta_{\epsilon_1}(t), \\ z_2' = c_2(\sigma^{[n]}(z_1) - z_2)^3\, \zeta_{\epsilon_2}(-t) \end{cases} \tag{13}$$

with initial conditions $z_1(0) = z_2(0) = \overline{x}_0$, where $c_1, c_2, m, n, \epsilon_1$, and ϵ_2 are still to be defined, and σ is the error-contracting function defined in Lemma 19.

We would like (13) to satisfy the following property: on $[0, 1/2]$,

$$|z_2'(t)| \leq \gamma. \tag{14}$$

This can be achieved by taking $\epsilon_2 = \gamma/K$, where K is a bound for $c_2(\sigma^{[n]}(z_1) - z_2)^3$ in the interval $[0, 1]$. Since $|x|^3 \leq x^4 + 1$ for all $x \in \mathbb{R}$, we can take $\epsilon_2 = \frac{\gamma}{c_2(\sigma^{[n]}(z_1)-z_2)^4} + \frac{\gamma}{c_2}$. Now notice that $z_2(0)$ has an error bounded by δ_{in}. This fact, together with (14) and the fact that z_2' might be subject to perturbations of amplitude not exceeding δ_{ev} imply that

$$|z_2(t) - x_0| \leq \delta_{in} + (\delta_{ev} + \gamma)/2 = \delta_{out} < 1/2 \quad \text{for } t \in [0, 1/2]. \tag{15}$$

Therefore, for m satisfying $\sigma^{[m]}(\delta_{out}) < \gamma$, we have $\left|\sigma^{[m]}(z_2(t)) - x_0\right| < \gamma$ for all $t \in [0, 1/2]$. Hence, from the study of the perturbed targeting equation (10), where $\phi(t) = \zeta_{\epsilon_1}(t)$ and c_1 is obtained accordingly, we have (take $\rho = \gamma$ and consider (12))

$$|z_1(1/2) - \omega(x_0)| < 2\gamma + \frac{\delta_{ev}}{2} \leq \delta_{in}. \tag{16}$$

For the interval $[1/2, 1]$ the roles of z_1 and z_2 are interchanged. Similarly to the reasoning done for z_2 on $[0, 1/2]$, take $\epsilon_1 = \frac{\gamma}{c_1(\Omega \circ \sigma^{[m]}(z_2)-z_1)^4} + \frac{\gamma}{c_1}$ so that on $[0, 1/2]$

$$|z_1'(t)| \leq \gamma.$$

From this inequality, (16), and the fact that z_2' might be subject to perturbations of amplitude not exceeding δ_{ev}, we conclude that

$$|z_1(t) - \omega(x_0)| \leq \delta_{in} + (\delta_{ev} + \gamma)/2 = \delta_{out} < 1/2 \quad \text{for } t \in [1/2, 1].$$

Therefore, for $n = m$, we have $\left|\sigma^{[n]}(z_1(t)) - \omega(x_0)\right| < \gamma$ for all $t \in [1/2, 1]$. Hence, from the study of the perturbed targeting equation (10), where $\phi(t) = \zeta_{\epsilon_2}(t)$ and c_2 is obtained accordingly, we have

$$|z_2(1) - \omega(x_0)| < 2\gamma + \frac{\delta_{ev}}{2} \leq \delta_{in}.$$

Now we can repeat the procedure for intervals $[1, 2]$, $[2, 3]$, etc. to conclude that for all $j \in \mathbb{N}$ and for all $t \in [j, j + 1/2]$,

$$\left|z_1(t) - \omega^{[j]}(x_0)\right| \leq \delta_{out}.$$

Moreover, z_1 is defined as the solution of an ODE written in terms of PIVP functions, and all coefficients of this ODE are in $\mathbb{Q}[\pi]$. Then, by Theorem 4, z_1 is a vector PIVP function with parameters in $\mathbb{Q}[\pi]$.

5 Application – Undecidability for PIVPs with Comparable Parameters

It is well known from the basic existence-uniqueness theory of ODEs [22], [23] that if f is analytic, then the IVP

$$x' = f(t, x), \quad x(t_0) = x_0 \tag{17}$$

has a unique solution $x(t)$ defined on a maximal interval of existence $I = (\alpha, \beta) \subset \mathbb{R}$ that is analytic on I [24]. The interval is maximal in the sense that either $\alpha = -\infty$ or $x(t)$ is unbounded as $t \to \alpha^+$ with similar conditions applying to β (see Proposition 20 for details). Actually, f only needs to be continuous and locally Lipschitz in the second argument for this maximal interval to exist.

A question of interest is the following: is it possible to design an automated method that, on input (f, t_0, x_0), gives as output the maximal interval of existence for the solution of (17)? In computability theory, e.g. [25], [26], it is well known that some problems cannot be answered by the use of an algorithm (more precisely, by the use of a Turing machine). Such problems are labelled *undecidable* and many examples are known. The most prominent undecidable problem is the Halting Problem: given a universal Turing machine and some input to it, decide whether the machine eventually halts or not. To address this kind of questions for IVPs, we use the computable analysis approach [17], [12], [16], which we presented in the end of Section 2. Using that approach, it was shown in [18] that given an analytic IVP (17), defined with computable data, its corresponding maximal interval may be non-computable.

Non-computability results related to initial-value problems of differential equations are not new. For example, Pour-El and Richards [27] showed that if we relax the condition of analyticity in the IVP (17) defined with computable data, it can have non-computable solutions. In [28], [29] it is shown that there is a three-dimensional wave equation, defined with computable data, such that the unique solution is nowhere computable. However, in these examples, non-computability is not "genuine" in the sense that the problems under study are ill-posed: either the solution is not unique or it is unstable [30]. In other words, ill-posedness was at the origin of non-computability in those examples. In contrast, an analytic IVP (17) is classically well-posed and, consequently, the non-computability results do not seem to reflect computational and well-posedness deficiencies inherited by the problems.

Motivated by the non-computability result obtained in [18], this latter paper also addresses the following problem: while it is not possible to compute the maximal interval of (17) is it possible to compute some partial information about it? In particular, is it possible to decide if this maximal interval is bounded or not?

This question has interest on its own for the following reason. In many problems, we implicitly assume that t is defined for "all time". For example, if one wants to compute sinks or limit cycles associated with ODEs, this only makes sense if the solution of the ODE is defined for all times $t > t_0$. This is also implicitly assumed in problems like reachability [31], [32], [33], [34], [35], etc. For this reason, those problems only make sense when associated with ODEs for which the maximal interval is unbounded. So, it would be interesting to know which are the "maximal" classes of functions f for which the boundedness problem is decidable.

In [18], it was shown that for the general class of analytic IVPs, the boundedness problem of the maximal interval is undecidable. Here we will deepen this result: we will show that the boundedness problem is still undecidable for PIVPs of degree greater or equal than 56 with parameters in $\mathbb{Q}[\pi]$. Our result is slightly different in form from the case of the general class of analytic IVPs. Indeed, the coefficients of the polynomials are coded as finite sequences of integers and not as ρ-name satisfying (5), though from these finite sequences of integers one can easily compute ρ-names for the coefficients of the polynomials.

The boundedness problem is decidable for linear differential equations thus implying that the boundary between decidability/undecidability lies in the class of polynomials of degree n, for some $2 \leq n \leq 56$.

This result is shown using methods which differ from those employed in [18]. This result was already stated in [11], but we now present its proof.

The following result introduces the notion of maximal interval for ODEs and follows as an immediate consequence of the fundamental existence-uniqueness theory for the initial-value problem (17), where the analyticity condition is dropped for f [22], [23], [36].

Proposition 20 *Let E be an open subset of $\mathbb{R}^{n+1}$ and assume that $f : E \to \mathbb{R}^n$ is continuous on E and locally Lipschitz in the second argument (i.e. in the last n components). Then for each $(t_0, x_0) \in E$, the problem (17) has a unique solution $x(t)$ defined on a maximal interval (α, β), on which it is C^1. The maximal interval is open and has the property that, if $\beta < +\infty$ (resp. $\alpha > -\infty$), either $(t, x(t))$ approaches the boundary of E or $x(t)$ is unbounded as $t \to \beta^-$ (resp. $t \to \alpha^+$).*

Note that, as a particular case, when $E = \mathbb{R}^{n+1}$ and $\beta < \infty$, $x(t)$ is unbounded as $t \to \beta^-$. This will be the case under study in this section.

We now introduce a definition that allows us to compare real numbers of some given set, to avoid the trivial undecidability of the boundedness problem sketched in Section 1.

Definition 21 *We say that a set $D \subseteq \mathbb{R}$ is effectively comparable if D has a naming system γ, if all elements of D are γ-computable, and if given γ-names of $x, y \in D$, then $x = y$ and $x < y$ are decidable*

In the previous definition, "naming system" is either a (finite) notation or a (infinite) representation of the elements of D according to Weihrauch [16, p. 33 and p. 52]. Next we show that $\mathbb{Q}[\pi]$ is effectively comparable. Indeed, given $a_0, \ldots, a_m \in \mathbb{Q}$ (which can easily be coded as a finite sequence using a finite alphabet A), we can take the notation $f : A^* \to \mathbb{Q}[\pi]$

$$f(a_0, \ldots, a_m) = \sum_{i=0}^{m} a_i \pi^i.$$

Moreover, if $\alpha, \beta \in \mathbb{Q}[\pi]$,

$$\alpha = \sum_{i=0}^{m} a_i \pi^i \quad \text{and } \beta = \sum_{i=0}^{n} b_i \pi^i,$$

where $a_0, \ldots, a_m, b_0, \ldots, b_n \in \mathbb{Q}$. We can decide if $\alpha = \beta$ since $\alpha = \beta$ iff $a_i = b_i$ for all i and a_i and b_i are rationals. We can also compute arbitrarly close approximations of α and β. Therefore, if $\alpha \neq \beta$, we can compare these values: we just need to start computing increasing approximations of α and β until we decide whether $\alpha < \beta$ or $\alpha > \beta$. The following result is similar to Theorem 12 in [11], but here we restrict the parameters of the PIVP to an effectively comparable set. This prevents the trivial undecidability discussed in Section 1.

Theorem 22 *Let D be an effectively comparable set such that $\mathbb{Q}[\pi] \subseteq D$. The following problem is undecidable: "Given $p : \mathbb{R}^{n+1} \to \mathbb{R}^n$ with polynomial components with coefficients in D (these coefficients are given by their names, as described in Definition 21), and $(t_0, x_0) \in \mathbb{Q} \times \mathbb{Q}^n$, decide whether the maximal interval of the IVP (1) is bounded or not".*

Actually, if we are given the description of a universal Turing machine, we can constructively define a set of polynomial ODEs simulating it that encodes the Halting Problem. If we use the small universal Turing machine presented in [37], having 4 states and 6 symbols, we obtain the following theorem.

Theorem 23 *Let D be an effectively comparable set such that $\mathbb{Q}[\pi] \subseteq D$. There is a vector $p : \mathbb{R}^{n+1} \to \mathbb{R}^n$, with $n \geq 1$, defined by polynomials with coefficients in D (these coefficients are given by their names, as described in Definition 21), where each component has degree less than or equal to* 56*, such that the following problem is undecidable: "Given $(t_0, x_0) \in \mathbb{Q} \times \mathbb{Q}^n$, decide whether the maximal interval of the IVP (1) is bounded or not".*

Proof. The idea to prove this theorem is to simulate with a set of polynomial ODEs Rogozhin's small universal Turing machine [37]. We can obtain a set of PIVPs simulating this Turing machine as described by Theorem 13, Proposition 14, and Corollary 15. Then we expand this PIVP system as a polynomial ODE using the techniques introduced in the proof of Theorem 4. Since the entire procedure is constructive and bottom-up, it is possible to determine the degrees of the polynomials appearing in the IVP. This will be done later in the proof.

The important point is that we can obtain a PIVP (1), with solution x, that satisfies for every $k \in \mathbb{N}$

$$\begin{cases} x_q(t) \leq m - \frac{11}{16} & \text{if } M \text{ has not halted at step } k \text{ and } t \leq k \\ x_q(t) \geq m - \frac{5}{16} & \text{if } M \text{ has already halted at step } k \text{ and } t \geq k \end{cases} \tag{18}$$

where the states of the Turing machine are encoded by numbers in $\{1, \ldots, m\}$ and $m = 4$ is the Halting state. Consider the IVP

$$\begin{cases} z_1' = x_q - (m - 1/2) \\ z_2 = \frac{1}{z_1} \end{cases} \iff \begin{cases} z_1' = x_q - (m - 1/2) \\ z_2' = ((m - 1/2) - x_q)z_2^2 \end{cases} \tag{19}$$

where $z_1(0) = z_2(0) = -1$. Since x_q appears as a component, we assume that this IVP is coupled with the PIVP defined by Proposition 14 and Theorem 4. It is easy to see that while M hasn't halted, $x_q - (m - 1/2) \leq -3/16$. Thus z_1 keeps decreasing and the IVP is defined in $(0, +\infty)$, i.e. the maximal interval is unbounded, if M never halts.

On the other hand, if M eventually halts, z_1 starts increasing at a rate of at least 3/16 and will do that forever. So, at some time it will have to assume the value 0. When this happens, a singularity appears for z_2 and the maximal interval is therefore (right-)bounded. For negative values of t just replace t by $(-t)$ in the PIVP (1) and assume t to be positive. It can be shown that the behavior of the system will be similar, and we reach the same conclusions for the left bound of the maximal interval. So M halts iff the maximal interval of the PIVP (19) is bounded, i.e. boundedness is undecidable.

It remains to determine the degree of the polynomials appearing in the definition of (1) and (19). We will now sketch how this is done. In what follows we assume that x and y are variables in an IVP, whose derivatives can be written as a polynomial (possibly involving other variables of the IVP) of degrees k and n, respectively (for short, we will simply say that x and y have degree k and n). Then our task is to know what is the degree of the PIVP giving functions like $\sin x$, etc.

(1) The case of sin and cos. We have

$$\begin{cases} (\sin x)' = x' \cos x \\ (\cos x)' = -x' \sin x \end{cases} \Longrightarrow \begin{cases} y_1' = x' y_2 \\ y_2' = -x' y_1 \end{cases}$$

where y_1 and y_2 substitute $\sin x$ and $\cos x$, respectively. So, if x has degree k, $\sin x$ and $\cos x$ can be replaced by variables having degree $k+1$.

(2) The case of arctan. One has

$$\begin{cases} (\arctan x)' = \frac{x'}{1+x^2} \\ \left(\frac{1}{1+x^2}\right)' = -\frac{2\acute{x}x}{(1+x^2)^2} \end{cases} \Longrightarrow \begin{cases} y_1' = x' y_2 \\ y_2' = -2x' x y_2^2 \end{cases}$$

where y_1 replaces $\arctan x$. So, $\arctan x$ can be replaced by a variable of degree $k+1$, but also introduces another variable of degree $k+3$.

(3) There are other functions that we didn't describe in detail previously, and that are used in our simulation (the reader is referred to [7]). But they are built from polynomials and the functions arctan and sin. So a straightforward application of the proof of Theorem 4 and the cases 1 and 2 above are enough to understand what happens with the degree of variables which derivative is described in terms of these functions.

Carrying out all the steps mentioned above, one can see that 56 is the highest degree for a variable that appears in the polynomial expansion of the ODE simulating Rogozhin's small universal Turing machine. ■

Let us remark that, while the boundedness problem of the maximal interval for unrestricted PIVPs is in general undecidable, this is not the case for some subclasses of polynomials. For instance, the boundedness problem is decidable for the class of linear differential equations (the maximal interval is always $\mathbb{R}$ — see e.g. [36, p. 79]) or for the class of one-dimensional autonomous differential equations where f is a polynomial of any degree (the ODE is separable, yielding an integral of a rational function that can be algorithmically solved). It would be interesting to investigate maximal classes where the boundedness problem is decidable.

6 Conclusion

In this paper we provide further results that establish a bridge between the theory of ODEs and computation (see [38] for an up-to-date review). We focus on polynomial initial value problems with computable and comparable parameters.

With respect to computation, our main result is that the boundness of the maximal interval of definition is undecidable even for PIVPs with comparable parameters and degree up to 56. We can view this result as a ODE analog to the undecidability of the Halting problem for Turing machines.

With respect to polynomial ODEs, we show that they can simulate a large class of dynamical systems – including Turing machines – in the presence of noise.

Based on the previous results we argue that polynomial ODEs, which are a well known model of physical phenomena, are also a powerful, yet realistic, model of continuous time computation.

Acknowledgments. The authors wish to thank Pieter Collins, Kerry Ojakian, and Ning Zhong, and the anonymous referees for useful remarks and comments. DG and MC were partially supported by *Fundação para a Ciência e a Tecnologia* and EU FEDER POCTI/POCI via CLC, SQIG - Instituto de Telecomunicações and grant SFRH/BPD/39779/2007 (DG). Additional support to DG was also provided by the *Fundação Calouste Gulbenkian* through the *Programa Gulbenkian de Estímulo à Investigação.* JB was partially supported by CMAF through FEDER and FCT-Plurianual 2007.

References

[1] M. W. Hirsch, S. Smale, Differential Equations, Dynamical Systems, and Linear Algebra, Academic Press, 1974.

[2] J. F. Ritt, Integration in Finite Terms. Liouville's Theory of Elementary Methods., Columbia Univ. Press, 1948.

[3] C. E. Shannon, Mathematical theory of the differential analyzer, J. Math. Phys. MIT 20 (1941) 337–354.

[4] D. S. Graça, J. F. Costa, Analog computers and recursive functions over the reals, J. Complexity 19 (5) (2003) 644–664.

[5] J. H. Hubbard, B. H. West, Differential Equations: A Dynamical Systems Approach — Higher-Dimensional Systems, Springer, 1995.

[6] D. S. Graça, N. Zhong, J. Buescu, The ordinary differential equation defined by a computable function whose maximal interval of existence is non-computable, in: G. Hanrot, P. Zimmermann (Eds.), Proceedings of the 7th Conference on Real Numbers and Computers (RNC 7), LORIA/INRIA, 2006, pp. 33–40.

[7] D. S. Graça, M. L. Campagnolo, J. Buescu, Computability with polynomial differential equations, Adv. Appl. Math. 40 (3) (2008) 330–349.

[8] M. S. Branicky, Universal computation and other capabilities of hybrid and continuous dynamical systems, Theoret. Comput. Sci. 138 (1) (1995) 67–100.

[9] M. L. Campagnolo, C. Moore, J. F. Costa, Iteration, inequalities, and differentiability in analog computers, J. Complexity 16 (4) (2000) 642–660.

[10] E. Hainry, Reachability in linear dynamical systems, in: A. Beckmann, C. Dimitracopoulos, B. Löwe (Eds.), Computability in Europe 2008 (CiE 2008), Vol. 5028 of LNCS, 2008.

[11] D. S. Graça, J. Buescu, M. L. Campagnolo, Boundedness of the domain of definition is undecidable for polynomial ODEs, in: R. Dillhage, T. Grubba, A. Sorbi, K. Weihrauch, N. Zhong (Eds.), Proceedings of the 4th International Conference on Computability and Complexity in Analysis (CCA 2007), FernUniversität in Hagen, 2007, pp. 127–135.

[12] K.-I. Ko, Computational Complexity of Real Functions, Birkhäuser, 1991.

[13] L. A. Rubel, F. Singer, A differentially algebraic elimination theorem with application to analog computability in the calculus of variations, Proc. Amer. Math. Soc. 94 (4) (1985) 653–658.

[14] D. S. Graça, Some recent developments on Shannon's General Purpose Analog Computer, Math. Log. Quart. 50 (4-5) (2004) 473–485.

[15] M. B. Pour-El, Abstract computability and its relations to the general purpose analog computer, Trans. Amer. Math. Soc. 199 (1974) 1–28.

[16] K. Weihrauch, Computable Analysis: an Introduction, Springer, 2000.

[17] M. B. Pour-El, J. I. Richards, Computability in Analysis and Physics, Springer, 1989.

[18] D. Graça, N. Zhong, J. Buescu, Computability, noncomputability and undecidability of maximal intervals of IVPs, Trans. Amer. Math. Soc.To appear.

[19] M. Campagnolo, C. Moore, Upper and lower bounds on continuous-time computation, in: I. Antoniou, C. Calude, M. Dinneen (Eds.), 2nd International Conference on Unconventional Models of Computation - UMC'2K, Springer, 2001, pp. 135–153.

[20] M. L. Campagnolo, Computational complexity of real valued recursive functions and analog circuits, Ph.D. thesis, Instituto Superior Técnico/Universidade Técnica de Lisboa (2002).

[21] M. L. Campagnolo, The complexity of real recursive functions, in: C. S. Calude, M. J. Dinneen, F. Peper (Eds.), Unconventional Models of Computation (UMC'02), LNCS 2509, Springer, 2002, pp. 1–14.

[22] E. A. Coddington, N. Levinson, Theory of Ordinary Differential Equations, McGraw-Hill, 1955.

[23] S. Lefshetz, Differential Equations: Geometric Theory, 2nd Edition, Interscience, 1965.

[24] V. I. Arnold, Ordinary Differential Equations, MIT Press, 1978.

[25] M. Sipser, Introduction to the Theory of Computation, PWS Publishing Company, 1997.

[26] J. E. Hopcroft, R. Motwani, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, 2nd Edition, Addison-Wesley, 2001.

[27] M. B. Pour-El, J. I. Richards, A computable ordinary differential equation which possesses no computable solution, Ann. Math. Logic 17 (1979) 61–90.

[28] M. B. Pour-El, J. I. Richards, The wave equation with computable initial data such that its unique solution is not computable, Adv. Math. 39 (1981) 215–239.

[29] M. B. Pour-El, N. Zhong, The wave equation with computable initial data whose unique solution is nowhere computable, Math. Log. Quart. 43 (1997) 499–509.

[30] K. Weihrauch, N. Zhong, Is wave propagation computable or can wave computers beat the Turing machine?, Proc. London Math. Soc. 85 (3) (2002) 312–332.

[31] R. Alur, D. L. Dill, Automata for modeling real-time systems, in: Automata, Languages and Programming, 17th International Colloquium, LNCS 443, Springer, 1990, pp. 322–335.

[32] E. Asarin, O. Maler, A. Pnueli, Reachability analysis of dynamical systems having piecewise-constant derivatives, Theoret. Comput. Sci. 138 (1995) 35–65.

[33] T. A. Henzinger, P. W. Kopke, A. Puri, P. Varaiya, What's decidable about hybrid automata?, J. Comput. System Sci. 57 (1) (1998) 94–124.

[34] O. Bournez, Achilles and the Tortoise climbing up the hyper-arithmetical hierarchy, Theoret. Comput. Sci. 210 (1) (1999) 21–71.

[35] V. D. Blondel, J. N. Tsitsiklis, A survey of computational complexity results in systems and control, Automatica 9 (36) (2000) 1249–1274.

[36] J. K. Hale, Ordinary Differential Equations, 2nd Edition, Robert E. Krieger Pub. Co, 1980.

[37] Y. Rogozhin, Small universal Turing machines, Theoret. Comput. Sci. 168 (2) (1996) 215–240.

[38] O. Bournez, M. L. Campagnolo, New Computational Paradigms. Changing Conceptions of What is Computable, Springer-Verlag, New York, 2008, Ch. A Survey on Continuous Time Computations, pp. 383–423.

A New Problem for Rule Following

Mark Hogarth
Girton College, Cambridge

1. *Introduction*

This is part of an extended argument of mine about the Church-Turing thesis (CTT). In Hogarth 1994 I argued that the thesis is a thoroughly empirical claim. In Hogarth 2004, 2008 I rejected that view, arguing instead that the thesis is really a pseudo-proposition like 'Australia is below England', or , better, like 'Euclidean geometry is the true geometry'. I say 'better' because my attitude towards this issue is shaped by an analogy with the concept of geometry. The key idea is that there is no fundamentally privileged computing device (e.g. Turing machine), in just the way that there is no privileged geometry (e.g. Euclidean). From a mathematical viewpoint there are lots of computers, lots of inequivalent ways to execute an algorithm (or procedure or rule). In my previous work I took it that the idea of losing CTT would have important consequences but consequences primarily for computability theory. Here I suggest the consequences extend further and may indeed touch on the nature of pure mathematics itself. Simple counting provides a case study. I also make some remarks about how the analogy with geometry might provide an answer to the stubborn problem of why pure mathematics is applicable to the natural sciences.

2. *The liquefaction of computability*

This line of enquiry began with the discovery of some non-Turing computers within the theory of general relativity (Hogarth 1992,1994, 2004, 2008; Earman and Norton 1993; Etesi and Nemeti 2001; Nemeti and David 2006). But what I have to say here, and indeed what is important, is not about those computers *per se* but rather about what their existence reveals about the concept of computability (written Computability). In that sense these non-Turing computers are like the first non-Euclidean geometries: what matters, conceptually speaking, is not the real-world accuracy of the models but their existence.

Let us remind ourselves how the concept of geometry came to change. Writing in 1897, when the status of non-Euclidean geometries was still controversial, Bertrand Russell began his book, *An Essay on the Foundations of Geometry*, as follows:

> 'When a long established system is attacked, it usually happens that the attack begins at a single point, where the weakness of the doctrine is particularly evident. But criticism once invited, is apt to extend much further than the most daring, at first, would have wished.'
>
> '"First cut the liquefaction, what comes last,
> But Fichte's clever cut at God himself?"'
>
> 'So it has been with Geometry.'

The liquefaction of Euclidean geometry began at the end of the 18th century, when Gauss questioned one of the axioms of the Euclidean system, the so-called Axiom of Parallels or Euclid's Postulate 5 ('Axioms XI' in some older manuscripts). Gauss experimented with systems lacking that axiom, but he never published his results for he feared an 'uproar of the Boeotians'[1]. The first publications to present a non-Euclidean system were due Lobachevsky in 1829 and Bolyai in 1832; others followed.

There are two strands of argument in Russell's book. First, that these new geometries represent real possibilities for the geometry of our world and deciding between them is an empirical matter. Secondly, that any possible geometry must possess constant curvature. This idea Russell held for essentially Kantian reasons. With the arrival in 1915 of Einstein's general theory of relativity this second strand became untenable, a point Russell himself was quick to acknowledge (Torretti, Chapter 7). Space (and spacetime) could possess variable curvature.

The process of liquefaction therefore advanced further than Russell had anticipated in 1897. After Einstein the term 'geometry' became broad and vague. 'Broad' because it encompassed a myriad of systems suggested by the sciences; 'vague' because the lessons of the past were not to attempt to isolate some 'essential' features of geometry (like constant curvature); not, in short, to try to cauterize the concept of geometry. We are now not even tempted to ask what makes a system a 'geometry'. The term, like the term 'game', gets applied because of family resemblances with accepted archetypes. We rightly tend to leave it at that.

One way to phrase the conceptual shift taking us from Euclidean geometry to post-Einstein geometry, is to say that the concept of geometry (written Geometry) became two-sided, with physical geometry on one side and pure geometry on the other. Physical geometry is concerned with modeling the geometry of the physical world; it's part of physics. Pure geometry is concerned with the mathematical structure of each of the many geometrical models now in the offing; it's part of pure mathematics.

What I argued in Hogarth 2004, 2008 is that Computability now also looks two-sided, exactly because it has come to look so like Geometry.[2]

In Table 1 I summarize the main points of contact of the two concepts.

The '*SAD*s' referred to above are relativistic computers that can perform some non-Turing computable tasks. In the Appendix I give a representation of the two simplest, the SAD_1 and SAD_2, alongside representations of a finite Turing machine (*FTM*) and an ordinary Turing machine (*OTM*). The reader is encouraged to consult the references for further details, but the key point here is that 'running' the same algorithm first on the Turing machine and then on the SAD_1 will, in general, produce two different results. Hardware matters.

[1] Meaning 'fools'. The ancient Athenians took a dim view of their neighbours in Boeotia.

[2] This is arguably more than just a metaphor / simile. Computers employ space and time = spacetime = spacetime *geometry*. Thus if Geometry is two-sided, then it is not unreasonable to expect Computability to be two-sided too.

Geometry	Computability
Euclidean	Turing
The various representations of Euclidean geometry by e.g. Euclid, Playfair, Wallis, Saccheri, Riemann.	The various representations of Turing computability by e.g. Church, Kleene, Turing, and Post
‘Euclidean geometry is the true geometry’—call this *Euclid’s thesis*; it has a dual role: as a statement of the ‘truth’ of Euclidean geometry, and as an heuristic to complete ‘proofs’	Church-Turing thesis; it has dual role: as a statement of the ‘truth’ of Turing computability, and as an heuristic to complete ‘proofs’.
Geometries of Lobachevsky, Bolyai, Riemann, Einstein, etc.	Quantum computers, relativistic computers (*SAD*s), Davies’s machine
Geometry is two-sided: pure and physical	Computability is two-sided: pure and physical
From pure view point, there is no e.g. Euclidean *v.* Lobachevsky	From a pure view point, there is no e.g. Turing machine *v.* SAD_1
Subjective terms e.g. ‘intuitive’, ‘natural’ fall out of use.	Subjective/vague terms e.g. ‘intuitive’, ‘natural’, ‘mechanical’ fall out of use.
Without Euclid’s thesis, rigorous proofs can be advanced.	Without CTT, rigorous proofs can be advanced.

Table 1

3. *Counting*

The idea that dispensing with CTT will have implications for only a sliver of mathematics, namely computability theory, is quickly dispelled when one is reminded that pure mathematics is shot through with algorithms (or rules or procedures – I take these to be synonyms).

Take as simple an example as ordinary counting, of the kind a child performs. The procedure (it’s like the Frege successor function) is this:

```
Begin with $n$=1
        Let $n$=$n$+1
                Reveal $n$
Repeat the second step
```

Suppose this algorithm is executed on some machine, and an observer consults it from time to time to see what is happening.

One is inclined to say the observer will see something like:

1,2,17,18,101,1201

(Note: the observer observers only from time to time, so some numbers will be missing. And of course the observer does not see the 'numbers' as such; rather she must interpret the machine's output: perhaps 17 dots on a screen is taken to be '17'.)

But this entirely depends upon which machine is executing the algorithm. A *FTM* (Appendix) will just stop at 101 (say), so our observer will see:

1,2,17,18,101

An *OTM* (Appendix) will never stop but our mortal observer must, at (say) 1739:

1,2,17,18,101,1739

With a SAD_1 our observer sees:

1,2,17,18,101,1201, 1739, ω

This is like the *OTM* case, except now at some point the observer can witness the first ordinal ω (again as naturally interpreted from the machine's output; see Appendix).

SAD_2 goes a step further. Now our observer sees:

1,2,17,18,101,1201, 2015, ω, $\omega+1$, 2ω, ω^2

One tends to think that an algorithm determines the 'output'. Here we see the output depends, non-trivially, upon the machine too. The output is irreducibly a function of two variables: the algorithm and the machine. An algorithm is by itself indeterminate, and those parts of pure mathematics involving algorithms or procedures or rules really are nothing but squiggles until coupled to a machine. The formalists then were right to take the squiggles in mathematics textbooks to be merely squiggles, but they were wrong in thinking the squiggles are pure mathematics.

This idea undermines a deeply held intuition. We find ourselves saying: but surely the algorithm above just is 1,2,3,…?

All one can say is that executing the algorithm on an *OTM* yields 1,2,3,…

The analogy with geometry may help here. Counting is like building a ladder with numbered rungs. The ladder is built in space (no space, no ladder), and the structure of the space, together with the building instructions (add another rung), determines the shape of the final ladder.

The fixation with 1,2,3,… is akin to the fixation with the long Euclidean ladder (in Euclidean space). But – and this is the point – there are other possible ladder structures.

This observation, that algorithm by itself lacks of determinacy, might seem redolent of Wittgenstein's skeptism about rule following:

> '… we get [a] pupil to continue a series (say + 2) beyond 1000 — and he writes 1000, 1004, 1008, 1012 (Wittgenstein 1953).'

Wittgenstein is drawing attention to how we use the '+' sign, and consequently what we *mean* by that sign. The pupil's answer, thinks Wittgenstein, does nothing to conflict with past uses (e.g. 10+2=12) – and so there is nothing ‘wrong’ with the answer.

But Wittgenstein’s problem is of course another problem. For even if the implementation of ‘+’ is unproblematic, transparent, the argument above shows that this mark by itself does no work.

4. *How is pure mathematics applicable to the natural sciences?*

A stubborn problem in the philosophy of mathematics is this: why is pure mathematics applicable to the natural sciences? Related to this is Benacerraf’s problem: how can we come to know the ‘inert’ objects of pure mathematics?

On the view that CTT holds, or indeed that computation is a transparent process, this is a hard problem. Counting (say) clearly is applicable to the real world and yet is comprised of an algorithm (a pure object) and is driven by a Turing machine (a pure object). How can these two pure objects come together to represent part of the physical world? Put like that, it’s a hard problem.

But when Computability is viewed as two-sided, like Geometry, this problem becomes tractable. Counting (say 1,2,3,…) can be treated as a pure system. But that system (algorithm + *OTM*) has, on the other side, a physics, the physics of Turing machines.

This is only a very partial answer to the problem, but it shows where the answer lies (in the physics of machines) – and that was really the problem.

Appendix

Figure 1 below shows representations of four computing devices. The vertical dimension is time, the horizontal space. A filled dot represents an event; an unfilled dot represents spacetime at ‘infinity’. The unattached filled dot is a typical event on a computer user’s worldline (though the worldline itself is not shown). A line is a worldline of a computer. In (i) the computer stops computing after a finite number of operations. This is the finite Turing machine or *FTM.* In (ii) the computer never stops computing (but the user can access only a finite number of computational steps). This is the ordinary Turing machine or *OTM.* In (iii) the computer is underpinned by a so-called Malament-Hogarth spacetime, which permits the user access to an infinite number of steps. This is called a SAD_1 computer (because it can decide arbitrary sentences in arithmetic with one quantifier). In (iv) is a $SAD_{2,}$ that is a ‘string’ of SAD_1s.

The numbers to the left of each computer are the numbers observed from time to time by a computer user. Of course the ‘numbers’ must be interpreted from the signal data. This holds for 1, 2, etc but also, in (iii), for *ω*, which is the interpretation of the absence of a signal. Further details can be found in Hogarth (2004).

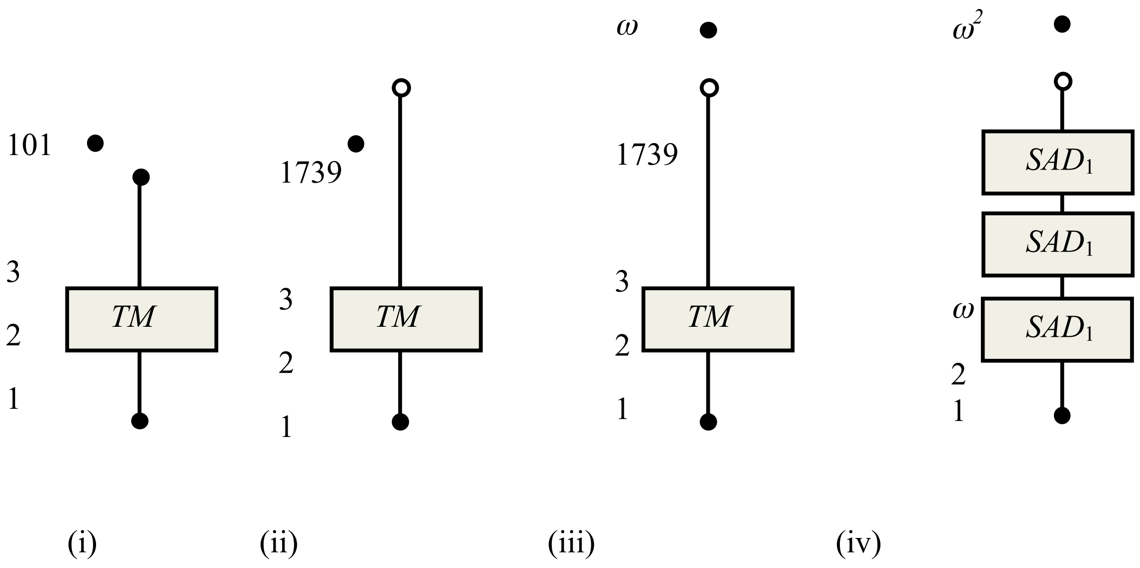


Figure 1. Four different computers, as represented in spacetime.

References

Davies, E.B. (2001) 'Building infinite machines', *British Journal for the Philosophy of Science* 52 671-582.

Earman, J. and Norton, J., (1993), 'Forever is a Day: Supertasks in Pitowsky and Malament-Hogarth Spacetimes', Philosophy of Science, 5, 22-42.

Etesi, G. and Nemeti, I., (2002) 'Non-Turing computations via Malament-Hogarth space-times'. *International Journal of Theoretical Physics* 41, 2, 341-370.

Hogarth, M., (1994), 'Non-Turing Computers and Non-Turing Computability'. In PSA 1994 vol. 1:126-138. East Lansing: *Philosophy of Science Association*. Ed. D. Hull, M. Forbes, and K. Okruhlik.

Hogarth, M. (2004), 'Deciding Arithmetic Using SAD Computers', *The British Journal for the Philosophy of Science* 55: 681-691.

Hogarth, M., (2008) 'Non-Turing Computers are the New Non-Euclidean Geometries', forthcoming, *International Journal of Unconventional Computing*. forthcoming.

Nemeti, I. and David, Gy., (2006) Relativistic computers and the Turing barrier. *Journal of Applied Mathematics and Computation* 178, 118-142.

Russell, Bertrand., (1996) *An Essay on the Foundations of Geometry*, Routledge.

Torretti, Roberto (1978) *Philosophy of Geometry from Riemann to Poincaré*. Dordrecht: D. Reidel Publishing Co.

Wittgenstein, L., (1953) *Philosophical Investigations*, Blackwell Publishing.

General relativistic hypercomputing and foundation of mathematics

Hajnal Andréka, István Németi and Péter Németi

Rényi Institute of Mathematics, Budapest P.O.Box 127, H-1364 Hungary,
andreka@renyi.hu, nemeti@renyi.hu, nemetip@gmail.com

Abstract. Looking at very recent developments in spacetime theory, we can wonder whether these results exhibit features of hypercomputation that traditionally seemed impossible or absurd. Namely, we describe a physical device in relativistic spacetime which can compute a non-Turing computable task, e.g. which can decide the halting problem of Turing machines or decide whether ZF set theory is consistent (more precisely, can decide the theorems of ZF). Starting from this, we will discuss the impact of recent breakthrough results of relativity theory, black hole physics and cosmology to well established foundational issues of computability theory as well as to logic. We find that the unexpected, revolutionary results in the mentioned branches of science force us to reconsider the status of the physical Church Thesis and to consider it as being seriously challenged. We will outline the consequences of all this for the foundation of mathematics (e.g. to Hilbert's programme).
Observational, empirical evidence will be quoted to show that the statements above do not require any assumption of some physical universe outside of our own one: in our specific physical universe there seem to exist regions of spacetime supporting potential non-Turing computations. Additionally, new "engineering" ideas will be outlined for solving the so-called blue-shift problem of GR-computing. Connections with related talks at the Physics and Computation meeting, e.g. those of Jerome Durand-Lose, Mark Hogarth and Martin Ziegler, will be indicated.

1 Introduction

We discuss here the impact of very recent developments in spacetime theory and cosmology on well established foundational issues (and interpretations) of logic and computability theory. The connections between computability theory, logic and spacetime theory (general relativity theory, GR) cut both ways: logic provides a tangible foundation for GR, cf. [1], while GR and its new developments might profoundly influence our interpretation of basic results of computability theory, as we will see in this paper. The new computability paradigms in turn offer feedback to the foundation of mathematics and logic.

Because of the interdisciplinary character of this paper, the first two sections are somewhat introductory, explaining the basic ideas for the nonspecialist. We will start speeding up beginning with section 3.

Two major new paradigms of computing arising from new physics are quantum computing and general relativistic computing. Quantum computing challenges complexity barriers in computability, while general relativistic computing challenges the physical Church-Turing Thesis itself. In this paper we concentrate on relativistic computers and on the physical Church-Turing Thesis (PhCT).

The PhCT is the conjecture that whatever physical computing device (in the broader sense) or physical thought-experiment will be designed by any future civilization, it will always be simulateable by a Turing machine. The PhCT was formulated and generally accepted in the 1930's. At that time a general consensus was reached declaring PhCT valid, and indeed in the succeeding decades the PhCT was an extremely useful and valuable maxim in elaborating the foundations of theoretical computer science, logic, foundation of mathematics and related areas.[1] But since PhCT is partly a physical conjecture, we emphasize that this consensus of the 1930's was based on the physical world-view of the 1930's. Moreover, many thinkers considered PhCT as being based on mathematics + common sense. But "common sense of today" means "physics of 100 years ago". Therefore we claim that the consensus accepting PhCT in the 1930's was based on the world-view deriving from Newtonian mechanics. Einstein's equations became known to a narrow circle of specialists around 1920, but about that time the consequences of these equations were not even guessed at. The world-view of modern black hole physics was very far from being generally known until much later, until after 1980.

Our main point is that in the last few decades there has been a major paradigm shift in our physical world-view. This started in 1970 by Hawking's and Penrose's singularity theorem firmly establishing black hole physics and putting general relativity into a new perspective. After that, discoveries and new results have been accelerating. In the last 10 years astronomers have obtained firmer and firmer evidence for the existence of ever larger, more and more exotic black holes [38],[35] not to mention evidence supporting the assumption that the universe is not finite after all [40]. Nowadays the whole field is in a state of constant revolution. If the background foundation on which PhCT was based has changed so fundamentally, then it is desirable to re-examine the status and scope of applicability of PhCT in view of the change of our general world-picture. A relevant perspective is e.g. in Cooper [9]. Cf. also [19], [15], [30], [36].

Assumption of an absolute time scale is a characteristic feature of the Newtonian world-view. Indeed, this absolute time has its mark on the Turing machine as a model for computer. As a contrast, in general relativity there is no absolute time. Kurt Gödel was particularly interested in the exotic behavior of time in general relativity. Gödel [16] was the first to prove that there are models of GR to which one cannot add a partial order satisfying some natural properties of a "global time". In particular, in GR various observers at various points of

[1] As a contrast, one of the founding fathers of PhCT, László Kalmár, always hoped for a refutation of PhCT and to his students he emphasized that PhCT is meant to be a challenge to future generations, it is aimed at "teasing" researchers to put efforts into attacking PhCT. [21]

spacetime in different states of motion might experience time radically differently. Therefore we might be able to speed up the time of one observer, say C (Cecil, for "computer"), relatively to the other observer, say P (Peter, for "programmer"). Thus P may observe C computing very fast. The difference between general relativity and special relativity is (roughly) that in general relativity this speed-up effect can reach, in some sense, infinity assuming certain conditions are satisfied. Of course, it is not easy to ensure that this speed-up effect happens in such a way that we could utilize it for implementing some non-Turing-computable functions.

In sections 2 and 3 we briefly recall from [30],[29] an intuitive idea of how this infinite speed-up can be achieved and how one can implement a computer based on this idea. More concrete technical details can be found in [15],[30] and to some extent in the remaining parts of this paper. For brevity, we call such thought-experiments *relativistic computers.* We will see that it is consistent with Einstein's equations, i.e. with general relativity, that by certain kinds of relativistic experiments, future generations might find the answers to non-computable questions like the halting problem of Turing machines or the consistency of Zermelo Fraenkel set theory (the foundation of mathematics, abbreviated as ZFC set theory from now on). Moreover, the spacetime structure we assume to exist in these experiments is based in [15],[30] on huge slowly rotating black holes the existence of which is made more and more likely (practically certain) by recent astronomical observations [38],[35].

We are careful to avoid basing the beyond-Turing power of our computer on "side-effects" of the idealizations in our mathematical model of the physical world. For example, we avoid relying on infinitely small objects (e.g. pointlike test particles, or pointlike bodies), infinitely elastic balls, infinitely (or arbitrarily) precise measurements, or anything like these. In other words, we make efforts to avoid taking advantage of the idealizations which were made when GR was set up. Actually, this kind of self-constraint is essential for the present endeavor as can be illustrated by [41, pp.446-447].

In sections 4–6 we discuss some essential questions of principle as well as some technical questions in connection with realizability of a relativistic computer, such as e.g. the so-called blue-shift problem, assuming infinity of time and space. Many of these questions come close to the limits of our present scientific knowledge, provoking new research directions or adding new motivations to already existing ones. We show that, at least, the idea of relativistic computers is not in conflict with presently accepted scientific principles. E.g. we recall that the presently accepted standard cosmological model predicts availability of infinite time and space. We also show that the principles of quantum mechanics are not violated, no continuity of time or space is presupposed by a relativistic computer. Discussing physical realizability and realism of our design for a computer is one of the main issues in [30, §5].

A virtue of the present research direction is that it establishes connections between central questions of computability theory and logic, foundation of mathematics, foundation of physics, relativity theory, cosmology, philosophy, particle physics, observational astronomy, computer science and Artificial Intelligence [44]. E.g. it gives new kinds of motivation to investigating central questions of these fields like "is the universe finite or infinite (both in space and time) and in what sense", "exactly how do huge Kerr black holes evaporate" (quantum gravity), "how much matter is needed for coding one bit of information (is there such a lower bound at all)", questions concerning the statuses of the various cosmic censor hypotheses, questions concerning the geometry of rotating black holes [5], to mention only a few. The interdisciplinary character of this direction was reflected already in the 1987 course given by the present authors [28] during which the idea of relativistic hypercomputers emerged and which was devoted to connections between the above mentioned areas.

Section 6 is also about the impact of general relativistic computing on the foundation of mathematics.

Section 7 is devoted to the impact of the "new computability paradigm" on spacetime theory. There, we discuss a different kind of motivation for studying relativistic computers. Namely, such a study may have applications to theoretical physics as follows. To GR, there is an infinite hierarchy of hypotheses called causality constraints which can be added to GR as outlined in the monograph [11, §6.3, pp.164-167]. Among these occur the various versions of the cosmic censor hypothesis (CCH) of which the basic reference book of relativity theory [42, p.303] writes "whether the cosmic censor conjecture is correct remains the key unresolved issue in the theory of gravitational collapse". On p.305 [42] writes "... there is virtually no evidence for or against the validity of this second version of CCH". These causality hypotheses play a role in GR analogous with the role formulas like GCH independent of ZF set theory play in set theory (or logic). These causality hypotheses are independent of GR (they are not implied by GR), and their status is the subject of intensive study as op. cit. illustrates this. Now, the study of relativistic computers could, in principle, reveal how the physical Church Thesis PhCT is situated in this hierarchy, in a sense which we will discuss in section 7. If we could find out which one of these constraints imply PhCT (or are implied by PhCT), that could be illuminating in why certain issues are difficult to settle about these constraints, cf. e.g. Etesi [14] and [42, p.303].

Tangible data underlying the above interconnections and also more history, references are available in [30]. The textbook Earman [11, p.119, section 4.9] regards the same interdisciplinary perspective as described above to be one of the main virtues of the present research direction. It is the unifying power of logic which makes it viable to do serious work on such a diverse collection of topics. One of the main aims of the research direction represented by [1]–[3], [23]–[25] is to make relativity theory accessible for anyone familiar with logic.

2 Intuitive idea for non-Turing GR computing

In this section we briefly recall from [30, 29] the ideas of how relativistic computers work, without going into technical details. The technical details are elaborated, among others, in [15], [19], [30]. To make our narrative more tangible, we use the example of huge slowly rotating black holes for our construction of relativistic computers. These are called "slow-Kerr" black holes in the physics literature. There are many more kinds of spacetimes suitable for carrying out essentially the same construction for a relativistic computer. We chose rotating black holes because they provide a tangible example for illustrating the kind of reasoning underlying general relativistic approaches to breaking the "Turing barrier". Mounting astronomical evidence for their existence makes them an even more attractive choice for our didactic purposes. In passing we note that some intuitively easy to read fine-structure investigations of slowly rotating Kerr-Newman black holes are found in the recent [5].

We start out from the so-called Gravitational Time Dilation effect (GTD). The GTD is a theorem of relativity theory, it says that gravity makes time run slow. Clocks that are deep within gravitational fields run slower than ones that are farther out. Roughly, GTD can be interpreted by the following thought-experiment. Choose a high enough tower on the Earth, put precise enough (say, atomic) clocks at the bottom of the tower and the top of the tower, then wait enough time, and compare the readings of the two clocks. The clock on the top will run faster (show more elapsed time) than the one in the basement. So, gravity causes the clock on the top ticking faster. Therefore computers there also compute faster. If the programmer in the basement would like to use this GTD effect to speed up his computer, he can just send his computer to the top of the tower and he gets some speed-up effect. We want to increase this speed-up effect to the infinity. Therefore, instead of the Earth, we use a huge black hole. A black hole is a region of spacetime with so big "gravitational pull" that even light cannot escape from this region. There are several types of black holes, an excellent source is Taylor and Wheeler [39]. For our demonstration of the main ideas here, we will use a huge, slowly rotating black hole. These black holes have two *event horizons*, these are bubble-like surfaces one inside the other, from which even light cannot escape. See Figures 1–2.

As we approach the outer event horizon from far away outside the black hole, the gravitational "pull" of the black hole approaches infinity as we get closer and closer to the event horizon. This is rather different from the Newtonian case, where the gravitational pull also increases but remains finite everywhere. For a while from now on "event horizon" means "outer event horizon". Imagine observers suspended over the event horizon. Here, suspended means that the distance between the observer and the event horizon does not change. Equivalently, instead of suspended observers, we could speak about observers whose spaceship is hovering over the event horizon, using their rockets for maintaining altitude. Assume one suspended observer C is higher up and another one, P, is suspended lower down. So, C sees P below her while P sees C above him. Now the gravitational time dilation (GTD) will cause the clocks of C run faster than the clocks of P. They both agree on this if they are watching each other e.g. via photons. Let us keep the height of C fixed. Now, if we gently lower P towards the event horizon, this ratio between the speeds of their clocks increases and, as P approaches the event horizon, this ratio approaches infinity. This means that for any integer n, if we want C's clocks to run n times as fast as P's clocks, then this can be achieved by lowering P to the right position. If we could suspend the lower observer P on the event horizon itself then from the point of view of C, P's clocks would freeze, therefore from the point of view of P, C's clocks (and computers!) would run infinitely fast, hence we would have the desired infinite speed-up upon which we could then start our plan for breaking the Turing barrier. The problem with this plan is that it is impossible to suspend an observer on the event horizon. As a consolation for this, we can suspend observers arbitrarily close to the event horizon. To achieve an "infinite speed-up" we could do the following. We could lower and lower again P towards the event horizon such that P's clocks slow down (more and more, beyond limit) in such a way that there is a certain finite time-bound, say b, such that, roughly, throughout the whole history of the universe P's clocks show a time smaller than b. More precisely, by this we mean that whenever C decides to send a photon to P, then P will receive this photon before time b according to P's clocks. This is possible. See Figure 2.

There is a remaining problem to solve. As P gets closer and closer to the event horizon, the gravitational pull or gravitational acceleration tends to infinity. If P falls into the black hole without using rockets to slow his fall, then he does not have to withstand the gravitational pull of the black hole. (He would only feel the so-called tidal forces which can be made negligibly small by choosing a large enough black hole.) However, his falling through the event horizon would be so fast that some photons sent after him by C would not reach him outside the event horizon. Thus P has to approach the event horizon relatively slowly in order that he be able to receive all possible photons sent to him by C. In theory he could use rockets for this purpose, i.e. to slow his fall (assuming he has unlimited access to fuel somehow). Because P approaches the event horizon slowly, he has to withstand this enormous gravity (or equivalently acceleration). The problem is that this increasing gravitational force (or acceleration) will kill P before his clock shows time b, i.e. before the planned task is completed. At the outer event horizon of our black hole we cannot compromise between these two requirements by choosing a well-balanced route for P: no matter how he will choose his route, either P will be crashed by the gravitational pull (acceleration), or some photons sent by C would not reach him. (This is the reason why we can not base our relativistic computer on the simplest kind of black holes, called Schwarzschild ones, which have only one event horizon and that behaves as we described above.) To solve this problem, we would like to achieve slowing down the "fall" of P not by brute force (e.g. rockets), but by an effect coming from the structure of spacetime itself. In our slowly rotating black hole, besides the gravitational pull of the black hole (needed to achieve the time dilation effect) there is a counteractive repelling effect coming from the rotation of the black hole. This repelling effect (or cushioning effect) is analogous to "centrifugal force" in Newtonian mechanics and will cause P to slow down in the required rate. So the idea is that instead of the rockets of P, we would like to use for slowing the fall of P this second effect coming from the rotation of the black hole. The *inner event horizon* marks the point where the repelling force overcomes the gravitational force. Inside the inner horizon, it is possible again to "suspend" an observer, say P, i.e. it becomes possible for P to stay at a constant distance from the center of the black hole (or equivalently from the event horizons). It is shown in [15] that the path of the in-falling observer P can be planned in such a way that the event when P reaches the inner event horizon corresponds to the time-bound b (on the wristwatch of P) mentioned above before which P receives all the possible messages sent out by C. In fact, the path of P can be chosen (to be a geodesic, i.e.) so that P does not have to use rockets at all, all the "slowing down" is done by the spacetime itself.

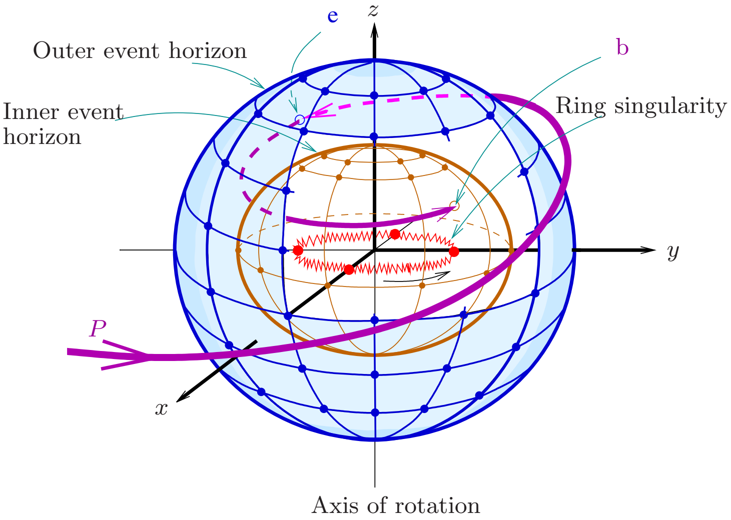


Fig. 1. A slowly rotating (Kerr) black hole has two event horizons and a ring-shape singularity (the latter can be approximated/visualized as a ring of extremely dense and thin "wire"). The ring singularity is inside the inner event horizon in the "equatorial" plane of axes x, y. Time coordinate is suppressed. Figure 2 is a spacetime diagram with x, y suppressed. Rotation of ring is indicated by an arrow. Orbit of in-falling programmer P is indicated, it enters outer event horizon at point e, and meets inner event horizon at point b.

By this we achieved the infinite speed-up we were aiming for. This infinite speed-up is represented in Figure 2 where P measures a finite proper time between its separation from the computer C (this separation point is not represented in the figure) and its touching the inner horizon at proper time b (which point also is not represented in Figure 2). It can be seen in the figure that whenever C decides to send a photon towards P, that photon will reach P before P meets the inner horizon.

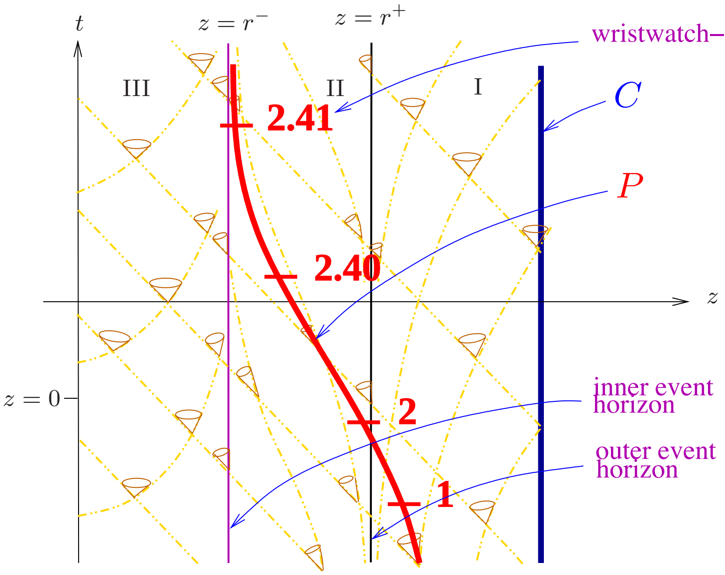


Fig. 2. The “tz-slice” of spacetime of slowly rotating black hole in coordinates where z is the axis of rotation of black hole. The pattern of light cones between the two event horizons r^- and r^+ illustrates that P can decelerate so much in this region that he will receive outside of r^- all messages sent by C. r^+ is the outer event horizon, r^- is the inner event horizon, $z = 0$ is the “center” of the black hole as in Figure 1. The tilting of the light cones indicates that not even light can escape through these horizons. The time measured by P is finite (measured between the beginning of the experiment and the event when P meets the inner event horizon at b) while the time measured by C is infinite.

3 Implementation for a relativistic computer

We now use the above to describe a computer that can compute tasks which are beyond the Turing limit. To break the Turing limit, let us choose the task, for an example, to decide whether ZFC set theory is consistent. I.e. we want to learn whether from the axioms of set theory one can derive the formula FALSE. (This formula FALSE can be taken to be $x \neq x$.) The programmer P and his computer C are together (on Earth), not moving relative to each other, and P uses a finite time-period for transferring input data to the computer C as well as for programming C. After this, P boards a huge spaceship, taking all his mathematical friends with him (like a Noah’s Ark), and chooses an appropriate route towards a huge slowly rotating black hole, entering the inner event horizon when his wrist-watch shows time b. While he is on his journey towards the black hole, the computer that remained on the Earth checks one by one the theorems of set theory, and as soon as the computer finds a contradiction in set theory, i.e. a proof of the formula FALSE from the axioms of set theory, the computer sends a signal to the programmer indicating that set theory is inconsistent. If it does not find a proof for FALSE, the computer sends no signal.

The programmer falls into the inner event horizon of the black hole and after he has crossed the inner event horizon, he can evaluate the situation. If a light signal has arrived from the direction of the computer, of an agreed color and agreed pattern, this means that the computer found an inconsistency in ZFC set theory, therefore the programmer will know that set theory is inconsistent. If the light signal has not arrived, and the programmer is already inside the inner event horizon, then he will know that the computer did not find an inconsistency in set theory, did not send the signal, therefore the programmer can conclude that set theory is consistent. So he can build the rest of his mathematics on the secure knowledge of the consistency of set theory. We will return to the issue of whether the programmer has enough space and time and resources for using the just gained information at the end of section 5.

The above outlined train of thought can be used to show that any recursively enumerable set can be decided by a relativistic computer [15]. Actually, more than that can be done by relativistic computers. Welch [43] shows that the arrangement described in section 3 using Kerr black holes can compute exactly Δ_2 problems in the arithmetical hierarchy (under some mild extra assumptions). Computability limits connected with such relativistic computers are also addressed in [19], [20], [36], [44].

Relativistic computers are not tied to rotating black holes, there are other general relativistic phenomena on which they can be based. An example is anti-de-Sitter spacetime which attracts more and more attention in explaining recent discoveries in cosmology (the present acceleration of the expansion of the universe, cf. [30]). Roughly, in anti-de-Sitter spacetime, time ticks faster and faster at farther away places in such a way that P can achieve infinite speed-up by sending away the computer C and waiting for a signal from her. This scenario is described and is utilized for computing non-Turing computable functions in [19]. This example shows that using black holes (or even singularities) is not inherent in relativistic computers.

Spacetimes suitable for an implementation of relativistic computation like described in this section are called Malament-Hogarth spacetimes in the physics literature. A relativistic spacetime is called Malament-Hogarth (MH) if there is an event (called MH-event) in it which contains in its causal past a worldline of infinite proper length. The spacetime of ordinary Schwarzschild black hole is not MH, the spacetime of rotating Kerr black hole is MH and any event within the inner event horizon is MH, in anti-de-Sitter spacetime every event is an MH-event, the spacetime of an electrically charged BH (called Reissner-Nordström spacetime) is MH and there are many other examples for MH.

We note that using MH spacetimes does not entail faith in some exotically "benevolent" global property of the whole of our universe. Instead, most of the MH spacetimes, like rotating BH's, can be built by a future, advanced civilization inside our usual "standard" universe of high precision cosmology. Namely, such MH spacetimes do not necessarily refer to the whole universe, but instead, to some "local" structure like a rotating ring of gravitationally collapsed matter in a "spatially finite part" of a more or less usual universe involving no particular global "witchcraft", so-to-speak. We are writing this because the word "spacetime" in the expression "MH spacetime" might be misleading in that it might suggest to the reader that it is an exotic unlikely property of the whole of God's creation, namely, the whole universe. However, in most MH spacetimes this is not the case, they are (in some sense) finite structures that can be built, in theory, by suitably advanced civilizations in a standard kind of universe like the one which is predicted by the present-day standard version of cosmology. In other words, nothing fancy is required from the whole universe, the "fancy part" is a structure which can, in theory, be manufactured in an ordinary infinite universe. Therefore in the present context it would be more fortunate to talk about MH regions of spacetime than about MH spacetimes.

4 Two sides of the coin and the blue-shift problem

A relativistic computer as we described in section 3 is a team consisting of a Computer (C, for Cecil) and a Programmer (P, for Peter).

How does the computer C experience the task of this computing? C will see (via photons) that the programmer P approaches the black hole (BH), and as he approaches it, his wristwatch ticks slower and slower, never reaching wristwatch time b. C will see the Programmer approaching the BH in all her infinite time. For C, the Programmer shines on the sky for eternity. The only effect of C's time passing is that this image gets dimmer and dimmer, but it will never disappear. Under this sky, C computes away her task consisting of potentially infinitely many steps, i.e. checking the theorems of ZFC one by one, in an infinite amount of time.

How does the Programmer experience the task of this computing? He is traveling towards the black hole, and he only has to check whether he received a special signal from the Computer or not. For this task, which consists of finitely many steps, he has a finite amount of time.

What would he see would he watch his team-member, the Computer? He would see the Computer computing faster and faster, speeding up so that when his (P's) wristwatch time reaches b, C would just flare up and disappear. Well, this flare-up would burn P, because it carries the energy of photons emitted during the whole infinite life of C, thus the total amount of this energy is infinite. In fact, we have to design a shield (or mirror) so that only intended signals from C can reach P. This means that we have to ensure that P does not see C! P's task is to watch whether there is one special kind of signal coming through this shield. All in all, P's task is to do finitely many steps in a finite amount of time.

A task in the literature is called supertask if it involves one to carry out infinitely many steps in a finite amount of time [12]. Therefore, by the above, we think that the relativistic computer need not implement a supertask.

The above led us to the so-called blue-shift problem [11]. This is the following. The frequency of light-signals (photons) sent by C to P gets increased (i.e. blue-shifted) by the time they reach P because of the infinite speed-up we worked so hard to achieve! Thus, if we do nothing about this, the one signal that C sends, can kill P. Further, P may not be able to recognize the blue-shifted signal. There are many solutions for this problem, two such solutions can be found in sections 5.3.1 and 5.4.1 of [30]. For example, C can arrange sending the signal to P such that C asks her sister C' to embark on a spaceship S which speeds up in the direction opposite to the direction of the Kerr hole, and send the signal from this spaceship. If S moves fast enough, then any signal sent from S to P will be red-shifted because of the speed of S. Then C chooses the speed of S to be such that the red-shift caused by this speed exactly cancels out the blue-shift caused by the gravitational effects at the event when P receives the signal.

Some new ideas on the blue-shift problem using mirrors. Programmer P sends out a second spaceship P' inhabited by robot P' running ahead of P in the same direction, i.e. towards the inner horizon of the black hole. So, P' travels faster than P and P' is between P and the inner horizon (in the relevant time period, of course). When computer C sends out the message (that e.g. an inconsistency was found in ZFC), the message is "beamed" to P' and not to P. Careful engineering is needed to ensure that the photons coming from C (or from anywhere the outside area) avoid P. See the Penrose diagram in Figure 3.

Bold solution: P' uses a huge mirror, by which P' reflects the message "back" to P. Since P' is moving extremely fast, we can say that P' is "running away" from the incoming photons. Therefore by the Doppler Effect, the frequency of the reflected photons (message) will be arbitrarily smaller (depending on the relative velocity[2] of P' and P) and so the energy of the message received by P from P' will be suitably small to ensure (i) recognizability and (ii) not burning P to death. For this, the pilots of P and P' should adjust their relative velocities appropriately. This seems to be possible as indicated in Figure 3. For the case something would go wrong with the above "bold plan", we include below a cautious plan.

Cautious plan: Instead of carrying a huge mirror, P' carries a large banner that can be spared. Now, the light signal (message) from C is directed to the banner carried by P'. So by the blue-shift effect, the message burns (destroys) the banner of P'. Even simpler solution is obtained by burning the whole of P'. Now, P is watching P' for the message. If P' disappears, then P concludes that P' was burned by the message, hence there was a message, hence P concludes that ZFC is inconsistent. If ZFC *is* consistent then C does not send a message to P', hence P' does not get burned, hence P "sees" that P' is still there, hence after both P and P' crossed the inner event horizon, P concludes that ZFC is consistent. Although this will happen only some time after P crosses the inner horizon, but that is consistent with our plan. The above plan of burning P' by the message might contain possibilities for error, e.g. P' might disappear because of some completely irrelevant accident (without C's sending a message). While this is so, the likelihood of such an accident can be minimized by the usual techniques of careful engineering, e.g. using redundancy (several copies of

[2] measurable e.g. by radar

P' moving in different directions all of which are targeted by C if C finds the inconsistency etc).

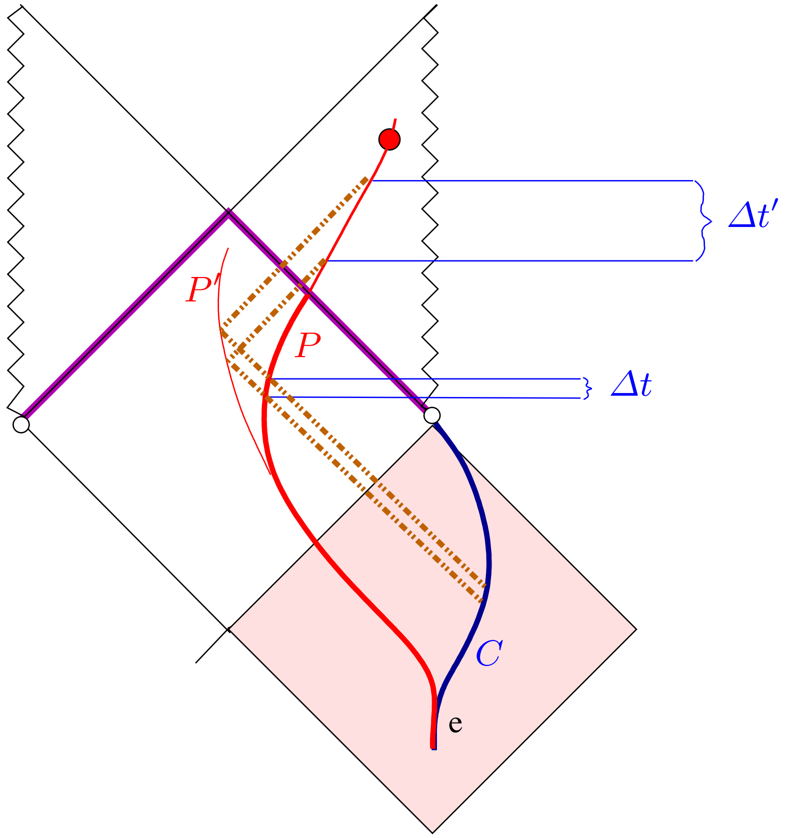


Fig. 3. Attacking the blue-shift problem by "mirrors" and by an "escaping" second spaceship P'. The "decoded" message is received by P only after he crossed the inner event horizon but not later than P''s crossing the inner event horizon is observed by P.

5 Some questions naturally arising

The following questions come up in connection with realizability of the plan described in section 3.

– can the programmer check whether the distant object he chose for a slowly rotating black hole is indeed one (whether it has the spacetime structure needed for his purposes)?

– can he check when he passed the event horizon?
– can he survive passing the event horizon?
– can he receive and recognize the signal sent by the computer?
– how long can he live inside the black hole?
– is there a way for the programmer to know that absence of signal from the computer is not caused by some catastrophe in the life of the computer?
– is it possible for a civilization to exist for an infinite amount of time?
– can the programmer repeat the computation or is this a once-for-a-lifetime computation for him?

Here we just assert that the answers to all these questions are in the affirmative, or at least do not contradict present scientific knowledge. These questions are discussed in detail in [30]. Below we address three of these questions.

On the question of traverseability of the event horizon: We chose the black hole to be large. If the black hole is huge[3], the programmer will feel nothing when he passes either event horizon of the black hole—one can check that in case of a huge black hole the so-called tidal forces on the event horizons of the black hole are negligibly small [32], [15].

On the question of how long the programmer can live after crossing the event horizon: The question is whether the programmer can use this new information, namely that set theory is consistent, or whatever he wanted to compute, for his purposes. A pessimist could say that OK they are inside a black hole, so—now we are using common sense, we are not using relativity theory—common sense says that the black hole is a small unfriendly area and the programmer will sooner or later fall into the middle of the black hole where there is a singularity and the singularity will kill the programmer and his friends. The reason why we chose our black hole to be a huge slowly rotating one, say of mass $10^{10}m_{\odot}$, is the following. If the programmer falls into a black hole which is as big as this and it rotates slowly, then the programmer will have quite a lot of time inside the black hole because the center of the black hole is relatively far from the event horizon. But this is not the key point. If it rotates, the "matter content", the so-called singularity, which is the source of the gravitational field of the black hole so-to-speak, is not a point but a ring (see Fig.1). So if the programmer chooses his route in falling into the black hole in a clever way, say, relatively close to the north pole instead of the equatorial plane, then the programmer can comfortably pass through the middle of the ring, never get close to the singularity and happily live on forever (see Fig.s 1,2). We mean, the rules of relativity will not prevent him from happily living forever. He may have descendants, he can found society, he can use and pass on the so obtained mathematical knowledge.

On the question of whether the computation can be repeated: Let us look at the extension of slow Kerr spacetime in [31, §3.3, pp.116-140]. Especially, consider the maximal slow Kerr spacetime (MSK) on Fig. 3.16, p.139. By considering this MSK, we can convince ourselves that our GR-computation is repeatable and with appropriate care it can be made deterministic. Some further

[3] this is a technical expression in observational astronomy

meditation on this repeatability of GR-computing can lead to new perspectives on the Platonism - formalism debates and views in the philosophical schools of the foundations of mathematics. E.g. one of the ages old arguments fades away, namely the argument saying that we cannot have access to any instances of actual infinity.

6 Can we learn something about infinity? Impact on the foundation of mathematics.

The relativistic computer as we implemented it in section 3 assumes that an infinite amount of time is available for C for computing. This seems to be essential for breaking the Turing barrier (by our construction). We are in a good position here, because of the following. As a result of very recent revolution in cosmology, there is a so-called standard model of cosmology. This standard model is based on matching members of a family of GR spacetimes against a huge number of observational data obtained by three different astronomical projects. This huge number of measurements (made and processed by using computers) all point amazingly to one specific GR spacetime. This spacetime is called the standard cosmological model, and in accordance with the so far highly successful scientific practice of the last 2500 years, we regard this standard model of the latest form of high-precision cosmology as the model best suited to explain observations and experience collected so far. According to this standard model, our universe is infinite both in regard of time and space, moreover there is an infinite amount of matter-energy available in it. We will see soon that the latter infinity is not needed for our construction. For more on this see Dávid [10], [30] and the references therein. Our point here is not about believing that our universe indeed has infinite time or not. The point is that assuming availability of an infinite amount of time for computing is not in contradiction with our present-day scientific knowledge.

We would like to say some words on the question of how much matter/energy is needed for storing, say, 10 bits of information. Although this question is not essential for the realizability of the relativistic computer (because of availability of infinite energy in the standard model of cosmology), we still find this question interesting for purely intellectual/philosophical reasons.

Is information content strongly tied to matter/energy content? Is there a lower bound to mass which is needed to store 10 bits of information? This is a question which has nagged one of the authors ever since he wrote his MsC thesis [27] where a separate section was devoted to this issue. The question is: “If I want to write more, do I need more paper to write on?” Right now it seems to us that the answer is in the negative. Matter and information might be two independent (orthogonal) “dimensions” of reality. The reason for this is the following. One might decide to code data by photons. Then the amount of matter/energy used is the energy total of these photons. But the energy of a photon is inversely proportional with its wavelength. So, one might double the wavelength of all photons and then one halved the energy needed to carry the same information one coded originally. If this is still too much energy expense, then one can double the wavelength again. Since there is no upper bound to the wavelengths of photons, there is no lower bound for the energy needed for storing 10 bits of data.

So, it seems to us that energy and information are not as strongly linked entities as energy and mass are (via $E = mc^2$). In the above argument when we said that there was no upper bound to the wavelength of possible photons, we used that according to the standard cosmological model the Universe is infinite in space. We note that Einstein when inventing photons did not say that there is a smallest nonzero value for energy. He said this only for light of a fixed color, i.e. fixed wavelength.

We would like to emphasize that we did not use that space is continuous. We seem to have used that time is continuous, but we can avoid that assumption by refining the implementation of relativistic computer. Constructions for this are in [30]. Thus, no contradictions with the principles of quantum mechanics seems to be involved in the idea of relativistic computer.

In the above, we argued that in principle, one can even build a relativistic computer sometime in the future. However, a fascinating aspect of relativistic computers for us is that they bring up mind-boggling questions about the nature of infinity. These questions would be worth thinking over even if our present-day science would predict a finite universe. We seem to understand and be familiar with the use of potential infinity in science. However, the above thought-experiment seems to use the notion of actual infinity. Is infinity a mental construction only or does it exist in a more tangible way, too? Can we learn something about actual infinity by making physical experiments? This leads to questions inherent in foundational issues in mathematics and physics. For more about this and about connection with Hilbert's Programme for mathematics we refer to [4].

7 Relativistic Computers and Causality Hypotheses in Physics

Let us consider the hierarchy of causality hypotheses $C0, \ldots, C6$ summarized in the monograph Earman [11, §6.3, pp.164-166]. None of these follow from GR (cf. e.g. [42, p.303]), they function as extra possible hypotheses for narrowing the scope of the theory. The strongest of these is the strong cosmic censor hypothesis $C6$ saying that spacetime is globally hyperbolic. A spacetime is called *globally hyperbolic* if it contains a spacelike hypersurface that intersects every causal curve without endpoint exactly once. This implies that the "temporal" structure of spacetime is basically the same as that of the Newtonian world in that it admits a "global time" associating a real number $t(p)$ to every point p of spacetime. In other words, $C6$ implies that spacetime admits a "global foliation", i.e. it is a disjoint union of global time-slices or "global now"-s. This is a quite extreme assumption and its role is more of a logical status, i.e. one investigates questions of what follows if $C6$ is assumed rather than assuming that it holds for the actual universe. Recall that Wald [42, p.305 lines 7-8] wrote about the cosmic censor hypothesis that "there is virtually no evidence for or against the validity of this", as we quoted around the end of section 1. Cf. also [11, pp.97,99] for further doubts on $C6$.

In section 1 we said that the assumption of absolute time has its marks on the Turing Machine as a model for computer, and now we see that $C6$ provides a kind of global time. Indeed, the construction of a GR-computer in this paper relies heavily on failure of $C6$, because Hogarth proved that no MH-spacetime is globally hyperbolic [11, Lemma 4.1, p.107]. This motivates the following question.

Question 1. Does (a carefully formulated variant of) PhCT follow from GR+$C6$?

On this question: PhCT has not been formalized precisely yet, this is part of why this question is asking for a formulation of PhCT which would follow from GR+$C6$. In this question we are asking if there are some natural and convincing extra conditions on physically realistic computability which would yield PhCT from GR+$C6$. The need for such extra realisticity assumptions is demonstrated by e.g. Tipler [41, pp.446-447]. Actually, we started collecting such conditions for physical realisticity in the middle of sec. 1 (in the paragraph beginning with "We are careful to avoid ..."). One might conjecture that, under suitable assumptions on physical realisticity and with a suitable formulation of PhCT, a kind of positive answer to Question 1 might be plausible[4].

Question 1 is about the connection between PhCT and the cosmic censor hypothesis $C6$. Next we concern ourselves with the connection between PhCT and the Malament-Hogarth property of a spacetime. We note that $C6$ implies that the spacetime is not Malament-Hogarth (NoMH for short), [11, Lemma 4.1, p.107] but NoMH does not imply $C6$ [11, p.110, first 2 sentences of sec.4.5]. Hence NoMH is a strictly weaker causality hypothesis than $C6$. We note that [14] explores the connection between $C6$ and MH. Again, we have no reason for believing in NoMH.

Question 2. Under what natural (extra) conditions is NoMH equivalent with what version of PhCT?

On this question: In theory, the MH property implies failure of PhCT (i.e. PhCT $\Rightarrow$ NoMH), because in any MH-spacetime one can, at least in theory, construct a GR-computer like in this paper, cf. [11, §4], [19]. However, there is a reason why in the works [15], [30], [29] we chose to implement our relativistic computer via a huge rotating black hole. Namely, huge-ness of the rotating BH was used to ensure that the tidal forces at the event horizons do not kill

[4] Very tentatively: Recently, emerging new kinds of computing devices like the Internet seem to pose a challenge against the conjecture in Question 1, cf. [44]. However, even the Internet (or even the Human Mind) will probably not prove ZFC consistent (assuming C6). So, perhaps we should separate PhCT into two theses, one about "hard" problems like proving the consistency of ZFC, and the other about problems in general which are not Turing computable.

the programmer. It is possible to construct a toy-example of a MH spacetime in which our kind of relativistic computer is not realistic physically. By physical realisticity we mean requirements that we do not use infinitely small computers (objects), infinitely precise measurements, or the like in designing our non-Turing computer, cf. [30] for more detail. We note that if we do not insist on physical realisticity, then already in Newtonian Mechanics PhCT would fail as demonstrated e.g. in Tipler [41, pp.446-447]. This motivates the question of what natural assumptions would ensure PhCT ⇒ NoMH or equivalently MH ⇒ NotPhCT in a physically realistic way. This is one direction of Question 2 above.

The other direction of Question 2 seems to be the harder one: Under what natural conditions (if any) does NotPhCT imply MH. I.e. under what conditions is

$(\star)$ NotPhCT ⇒ MH

true? One way of rephrasing $(\star)$ is to conjecture that if there is a physically realistic non-Turing computer then there must be one which is built up in the style of the present paper utilizing MH property of spacetime. (By non-Turing computer we mean a physical computer that can compute beyond the Turing barrier.) This seems to be a daring conjecture. But let us remember that the question was: under what conditions is statement $(\star)$ true. In particular, if the physical non-Turing computer "designed" in the book Pour-El and Richards [34] turns out to be physically realizable, then our conjecture (that under some reasonable conditions $(\star)$ might become true) might get refuted.

We note that the tentative conjecture implicit in Question 2 was arrived at jointly with Gábor Etesi.

8 Logic based conceptual analysis of GR and "reverse mathematics" for GR

So far we have been applying general relativity (GR) to logic and to the foundation of mathematics (FOM). In the other direction, logic and FOM are being applied to a conceptual analysis and logical/mathematical foundation of relativity (including GR). In more detail: FOM has an important branch called reverse mathematics. In the latter we ask the question(s) of which axioms of set theory are responsible (needed) for which important theorems of mathematics. In a series of papers, e-books, and book chapters, a team containing the present authors has been working on a programme which could be called "exporting the success story of FOM to a foundation of relativity" (set theorist Harvey Friedman coined this slogan), cf. e.g. [1]–[3], [23]–[25]. Roughly speaking, this group builds up relativity in first-order logic (like FOM is built up as a theory in the sense of mathematical logic), then analyzes the so obtained logical theory from various perspectives and (like in reverse mathematics) attempts to answer the so-called why-type questions asking why a certain prediction of general relativity is being predicted, i.e. which axioms of the theory are responsible for that particular prediction.

Section 7 above can be regarded as a small sample from the above quoted logical/conceptual analysis of GR and its extensions. A particular example of this feedback from logic and FOM to GR is the paper [5].

By the above sketched feedback from logic and FOM to GR, the "circle" GR $\mapsto$ Logic $\mapsto$ FOM $\mapsto$ GR promised in the introduction is completed.

9 History of relativistic computation

The idea of general relativistic computing as described in section 2 was found at different parts of the globe, independently. It was discovered by Németi in 1987 [28], Pitowsky in 1990 [33], Malament in 1989 [26], and Hogarth in 1992 [18] independently. Németi's idea used large slowly rotating black holes (slow Kerr spacetimes) but the careful study of feasibility and transversability of these was done later in Etesi-Németi [15]. All this led to a fruitful cooperation between the parties mentioned above, e.g. between Cambridge (Hogarth et al), Budapest (Németi et al), Pittsburgh (Earman et al). The first thorough and systematic study of relativistic computation was probably Hogarth [19]. Related work on relativistic computing include [43], [36], [11, §4], [12], [13], [17], [29].

Acknowledgements

Special thanks are due to Gyula Dávid for intensive cooperation and ideas on this subject. We are grateful for enjoyable discussions to Attila Andai, John Earman, Gábor Etesi, Stefan Gruner, Balázs Gyenis, Mark Hogarth, Judit Madarász, Victor Pambuccian, Gábor Sági, Endre Szabó, László E. Szabó, Renáta Tordai, Sándor Vályi, Jiří Wiedermann, Chris Wüthrich. We gratefully acknowledge support by the Hungarian National Foundation for scientific research grant No. T73601.

References

1. Andréka, H., Madarász, J. X. and Németi, I., Logic of Spacetime and Relativity Theory. In: Handbook of Spatial Logics, eds: M. Aiello, J. van Benthem, and I. Pratt-Hartman, Springer-Verlag, 2007. pp.607-711.
2. Andréka, H., Madarász, J. X. and Németi, I. with contributions from Andai, A., Sain, I., Sági, G., Tőke, Cs. and Vályi, S., On the logical structure of relativity theories. Internet book, Budapest, 2000. http://www.math-inst.hu/pub/algebraic-logic/olsort.html
3. Andréka, H., Madarász, J. X., Németi, I. and Székely, G., Axiomatizing relativistic dynamics without conservation postulates. Studia Logica 89,2 (2008), 163-186.
4. Andréka, H. and Németi, I., Relativistic computers and Hilbert's Program for mathematics. Manuscript.
5. Andréka, H., Németi, I. and Wüthrich, A twist in the geometry of rotating black holes: seeking the cause of acausality. General Relativity and Gravitation, to appear.
6. Barrow, J. D., How to do an infinite number of things before breakfast. New Scientist, 29 January 2005.

7. Barrow, J. D. and Tipler, F. J., The anthropic cosmological principle. Oxford University Press, 1986.
8. Chown, M., Smash and grab. New Scientist, 2002. April 6, pp.24-28.
9. Cooper, S. B., How can Nature Help Us Compute? In SOFSEM 2006: Theory and Practice of Computer Science - 32nd Conference on Current Trends in Theory and Practice of Computer Science, Merin, Czech Republic, January 2006 (Jiri Wiedermann, Julius Stuller, Gerard Tel, Jaroslav Pokorny, Maria Bielikova, editors), Springer Lecture Notes in Computer Science No. 3831, 2006, pp. 1-13.
10. Dávid, Gy., Modern cosmology - astronomical, phsical and logical approaches. Abstracts for the conference "Logic in Hungary 2005", http://atlas-conferences.com/cgi-bin/abstract/caqb-64
11. Earman, J., Bangs, crunches, whimpers, and shrieks. Singularities and acausalities in relativistic spacetimes. Oxford university Press, Oxford, 1995.
12. Earman, J. and Norton, J., Forever is a day: supertasks in Pitowsky and Malament-Hogarth spacetimes. Philosophy of Science 60 (1993), 22-42.
13. Earman, J. and Norton, J., Infinite pains: the trouble with supertasks. In: S. Stich (ed.), Paul Benacerraf: the Philosopher and his critics. Blackwell, New York, 1994.
14. Etesi, G., Note on a reformulation of the strong cosmic censor conjecture based on computability. Physics Letters B 550 (2002), 1-7. Revised version (2008).
15. Etesi, G. and Németi, I., Turing computability and Malament-Hogarth spacetimes. International Journal of Theoretical Physics 41,2 (2002), 342-370.
16. Gödel, K., Lecture on rotating universes. In: Kurt Gödel Collected Works, Vol. III. Eds.: Feferman, S., Dawson, J. S., Goldfarb, W., Parson, C. and Solovay, R. N., Oxford University Press, New York Oxford 1995. pp. 261-289.
17. Gyenis, B. and Roberts, B., Supertasks: Gödel strikes back. Preprint, Department of History and Philosophy of Science, University of Pittsburgh, April 2007.
18. Hogarth, M. L., Non-Turing computers and non-Turing computability. In: D. Hull, M. Forbes, and R. M. Burian (eds), PSA 1994, Vol. 1. East Lansing: Philosophy of Science Association. pp. 126-138.
19. Hogarth, M. L., Predictability, computability, and spacetime. PhD Dissertation, University of Cambridge, UK, 2000. http://ftp.math-inst.hu/pub/algebraic-logic/Hogarththesis.ps.gz
20. Hogarth. M. L., Deciding arithmetic using SAD computers. Brit. J. Phil. Sci. 55 (2004), 681-691.
21. Kalmár, L., An Argument Against the Plausibility of Church's Thesis. In Heyting, A. (ed.) Constructivity in Mathematics. North-Holland, Amsterdam, 1959.
22. van Leeuwen, J. and Wiedermann, J., The Turing Machine paradigm in comtemporary computing. In B. Enquist & W. Schmid (Eds.), Mathematics Unlimited - 2001 and Beyond. Springer-Verlag, Berlin, 2001. pp. 1139-1155.
23. Madarász, J. X., Németi, I. and Székely, G., Twin Paradox and the logical foundation of space-time. Foundation of Physics 36,5 (2006), 681-714. arXiv:gr-qc/0504118.
24. Madarász, J. X., Németi, I. and Székely, G., First-order logic foundation of relativity theories. In: Mathematical problems from applied logic II, International Mathematical Series Vol 5, Springer-Verlag, 2007. pp.217-252.
25. Madarász, J. X. and Székely, G., The effects of gravitation on clocks, proved in axiomatic relativity. Abstracts for the conference "Logic in Hungary 2005", http://atlas-conferences.com/cgi.bin/abstract.caqb-41.
26. Malament, D., Private communication to Earman, J., 1988. Cf. Earman's book in the present list of references, p.238.

27. Németi, I., Computing the disturbance of phone lines caused by high-voltage electrical power lines. Master Thesis, Dept. Electrodynamics, Technical University Budapest, 1965.
28. Németi, I., On logic, relativity, and the limitations of human knowledge. Iowa State University, Department of Mathematics, Ph. D. course during the academic year 1987/88.
29. Németi, I. and Andréka, H., Can general relativistic computers break the Turing barrier? In: Arnold Beckmann, Ulrich Berger, Benedikt Löwe, and John V. Tucker (eds.): Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, CiE 2006, Swansea, UK, July 2006, Proceedings, Lecture Notes in Computer Science 3988, Springer-Verlag Berlin Heidelberg 2006. pp.398-412.
30. Németi, I. and Dávid, Gy., Relativistic computers and the Turing barrier. Journal of Applied Mathematics and Computation 178 (2006), 118-142.
31. O'Neill, B., The geometry of Kerr black holes. A K Peters, Wellesley, 1995.
32. Ori, A., On the traversability of the Cauchy horizon: Herman and Hiscock's argument revisited, in: Internal Structures of Black Holes and Spacetime Singularitites (ed. A. Ori, L.M. Ori), Ann. Isra. Phys. Soc. 13, IOP (1997).
33. Pitowsky, I., The physical Church thesis and physical computational complexity. Iyyun A Jerusalem Philosophical Quarterly 39 (1990), 81-99.
34. Pour-El, M. B. and Richards, J. I., Computability in analysis and physics, Springer-Verlag, Berlin (1989).
35. Reynolds, C. C., Brenneman, L. W. and Garofalo, D., Black hole spin in AGN and GBHCs. Oct. 5, 2004. arXiv:astro-ph/0410116. (Evidence for rotating black holes.)
36. Sági, G. and Etesi, G., Transfinite computations and diameters of spacetimes. Preprint, Hungarian Academy of Sciences, Budapest.
37. Shagrir, O. and Pitowsky, I., Physical hypercomputation and the Church-Turing Thesis. Minds and machines 13 (2003), 87-101.
38. Strohmayer, T. E., Discovery of a 450 HZ quasi-periodic oscillation from the microquasar GRO J1655-40 with the Rossi X-Ray timing explorer. The Astrophysical Journal, 553,1 (2001), pp.L49-L53. (Evidence for rotating black holes.) arXiv:astro-ph/0104487.
39. Taylor, E. F. and Wheeler, J. A., Black Holes. Addison, Wesley, Longman, San Francisco, 2000.
40. Tegmark, M., Parallel Universes. Scientific American May 2003, pp.41-51.
41. Tipler, F. J., The physics of immortality. Anchor Books, New York, 1994.
42. Wald, R. M., General relativity. University of Chicago Press, Chicago 1984.
43. Welch, P. D., Turing unbound: on the extent of computation in Malament-Hogarth spacetimes. The Britisch Journal for the Philosophy of Science, to appear. School of Mathematics, University of Bristol, September 2006. arXiv:gr-qc/0609035v1
44. Wiedermann, J. and van Leeuwen, J., Relativistic computers and non-uniform complexity theory. In: Calude et al (eds.) UMC 2002. Lecture Notes in Computer Science Vol. 2509, Springer-Verlag, Berlin, 2002. pp.287-299.
45. Wischik, L., A formalization of non-finite computation. Dissertation for the degree of Master of Philosophy, University of Cambridge, 1997. www.wischik.com/lu/philosophy/non-finite-computation.html

The Computational Status of Physics: A Computable Formulation of Quantum Theory ☆

Mike Stannett

Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, Sheffield S1 4DP, United Kingdom

Abstract

According to the Church-Turing Thesis, effective formal behaviours can be simulated by Turing machines; this has naturally led to speculation that *physical* systems can also be simulated computationally. But is this wider claim true, or do behaviours exist which are strictly *hypercomputational*? Several idealised computational models are known which suggest the feasibility of physical hypercomputation – some based on cosmology; some on quantum theory; some on Newtonian physics. While the physicality of these models is debatable, they nonetheless throw into question the validity of simply extending the Church-Turing Thesis to include all physical, as well as effective formal, systems.

We propose that the physicality of hypercomputational behaviours be determined instead from first principles, and show that quantum theory can be reformulated in a way that partially resolves the question, by explaining why all physical behaviours can be regarded as 'computing something' in the standard computational state-machine sense. While our approach does not rule out the possibility of hypercomputation completely, it strongly limits the form such hypercomputation must take.

Key words: Hypercomputation, quantum theory, theory of computation, philosophy of mathematics, temporal structure, natural computation
PACS: 89.20.-a, 03.67.Lx
2000 MSC: 68Q05, 81P10, 81P68

1. Introduction

According to the Church-Turing Thesis (CTT), all effective computational behaviours can be simulated by Turing machines [1]. Although CTT was proposed in the context of formal mathematical systems, it is widely accepted that it can be applied more generally; in particular, given that physical devices are routinely used for computational purposes, it is now widely assumed that all (finitely-resourced, finitely-specified) physical machine behaviours can be simulated by Turing machines. However, this extended claim (known in the literature as *Thesis M* [2, 3]) is not a logical consequence of CTT, since it is not

☆This paper is still under the reviewing process.

Email addresses: M.Stannett@dcs.shef.ac.uk (Mike Stannett)

clear that every physical machine can meaningfully be said to 'compute something' in the same sense as Turing machines. Proponents of *digital physics* [4, 5, 6] stretch CTT still further, interpreting it to mean that *all* physical behaviours (whether machine-generated or not) are Turing-simulable.

The main aim of this paper is to investigate Thesis M and its extensions in more detail. Is it actually true that all physical behaviours are necessarily computable, or are there behaviours which go beyond the Turing limit? We will show that quantum theory can be reformulated in a way that partially resolves this question, by explaining why physical behaviours can indeed *always* be regarded as 'computing something' in the strict state-machine sense. While our approach does not rule out the possibility of hypercomputation completely, it limits the form such hypercomputation must take.

As we recall in section 2, this question has been debated indirectly over many decades [7]; but it has become prominent recently with the rise of quantum computation and digital physics. As is well known, Shor's algorithm [8] can factorise integers faster than any Turing program, and this already suggests that quantum theory has super-Turing potential. However, we need to distinguish carefully what we mean by 'hypercomputation' in this context. Where a computational model – for example, Deutsch's Universal Quantum Computer (UQC) [9] – computes the same class of functions as the Turing machine, albeit potentially faster, we call it a *super-Turing* model. If it is capable of computing functions which *no* Turing machine can compute, we call it *hypercomputational.* In particular, then, while the UQC is an apparently super-Turing model, it is well known that it is not hypercomputational, whence its implementation would not resolve the question whether hypercomputation is physically feasible.

1.1. Layout of the paper

We begin in section 2 by considering briefly what is already known concerning the relationship between physics and (hyper)computation. After summarising the information-theoretic approach familiar from *It from Bit*, we review three known hypercomputational systems: non-collision singularities in the Newtonian n-body problem; the Swansea *Scatter Machine Experiment* (also Newtonian); and Hogarth's cosmologically inspired family of *SAD* computers. We then focus on quantum theory, where it is unclear whether any hypercomputational model has yet been established. The question then arises whether a new approach might be able to resolve the issue. We will show that this is indeed the case, though only to a limited extent, by deriving a first-principles reformulation of Feynman's path-integral model; we review the standard formulation briefly in section 3, and present our new formulation in section 4.

In our version of Feynman's model, there is no such thing as a continuous trajectory. Instead, whenever a particle moves from one spacetime event to another, it does so by performing a finite sequence of 'hops', where each hop takes the particle directly from one location to another, with no intervening motion. Although this seems somewhat iconoclastic, we argue that 'finitary' motion of this kind is the only form of motion actually supported by observational evidence.

In section 5 we consider the computational significance of the model, insofar as it addresses the question whether hypercomputation is physically feasible. From a mathematical point of view it makes little difference whether we allow 'hops' to move a particle backwards as well as forwards in time, and we consider both models. In each case, the motion of a particle from one location to another generates a finite state machine (technically, an extended form of FSM called an X-machine [10]), where the machine's states are spacetime locations, and its transition labels reflect the (classical) action associated with each hop. In unidirectional time, the regular language generated by such a machine comprises just a single word, but if we allow time to be bidirectional, the availability of loops ensures that infinite regular languages can be generated. In both cases, when the motion is interpreted as an X-machine, the function computed by the motion can be interpreted as an amplitude, and if we sum the amplitudes of all machines with a given initial and final state, we obtain the standard quantum mechanical amplitude for the particle to move from the initial to the final location.

Section 6 concludes our argument, and includes suggestions for further research.

2. Motivation

In this section we review various arguments both for and against the physical feasibility of hypercomputation, and its converse, digital physics; for a more complete discussion of hypercomputational models, readers are invited to consult our earlier surveys of the field [7, 11]. The question, whether hypercomputational behaviours are physically feasible, obviously depends on ones conception of physics itself. Hypercomputational systems have been identified with respect to both relativistic and Newtonian physics. Where quantum theory is concerned, however, the situation is less clear cut.

2.1. Digital physics

Proponents of digital physics argue that the Universe *as a whole* is essentially computational, in the sense that its entire history can be viewed as the output of a digital computation [12]. The underlying idea appears to have been proposed first by Zuse, who suggested as early as 1967 that the Universe might be computed by a deterministic cellular automaton inhabited by 'digital particles' [13, 14].

Wheeler's subsequent *It from Bit* conception [15] reflected his conviction that information is just as physical as mass and energy, and indeed the relationship between information and gravitation has remained central to theories of quantum gravity ever since Bekenstein realised that black holes must possess intrinsic entropy [16, 17]. Hawking's observation that black holes can evaporate [18] naturally raises the question, what happens to quantum correlations that previously existed between particles on either side of the event horizon? Quantum theory appears to be inconsistent with causality in such a situation [19].[1]

The *It from Bit* doctrine focusses on the relationship between observation and information. Just as observations provide information, so information can affect observations, as was graphically illustrated (at first theoretically and eventually experimentally) by Wheeler's famous 'delayed-choice experiment', a modified version of the dual-slit experiment. As is well known, if one slit in a barrier is covered over, photons passing through

[1] There is as yet no empirical evidence that Hawking radiation, the mechanism by which evaporation takes place, exists in Nature. However, the final stages of a primordial micro black hole's evaporation should theoretically result in a burst of gamma-rays; one of the goals of the GLAST satellite, launched by NASA on 11th June 2008, is to search for such flashes.

the apparatus behave like particles, but when both slits are opened the 'particles' demonstrate interference effects. Wheeler asked what would happen if the decision to cover or uncover a slit were made *after* the photon had passed through the barrier, but before the outcome were detected. In practice, the photon's behaviour reflects the decision the experimenter will eventually make, even though this decision occurs after the encounter with the barrier has taken place. This suggests that the outcome of an experiment involves an interaction between the apparatus and the observer; the results you get are in some sense changed by the questions you decide to ask; or as Wheeler put it, "Every 'it' – every particle, every field of force, even the spacetime continuum itself – derives its function, its meaning, its very existence entirely – even if in some contexts indirectly – from the apparatus-elicited answers to yes-or-no questions, binary choices, bits" [20].

Schmidhuber has investigated a model of physics in which all possible realities are the outcomes of computations [12, 21]. By considering algorithmic complexity, we can examine the probability that a randomly selected universe would conform to any given set of behaviours; specific physical situations can be examined and predictions made, some of which might, in principle, be subject to experimental verification. It is important to note, however, that the type of physics this model generates is *not* generally consistent with conventional wisdom. For example, because digital physics assumes that universes are inherent deterministic, Schmidhuber's model rejects the notion that beta decay is truly random. Similarly, his model suggests that experiments carried out on widely-separated, but initially entangled, particles, should display non-local algorithmic regularities, a prediction which, he notes, 'runs against current mainstream trends in physics'.

A related concept is Tegmark's *Mathematical Universe Hypothesis* [6]. Tegmark notes that, if a complete Theory of Everything (TOE) exists, then the Universe must necessarily be a mathematical structure. In essence, this is because a *complete* TOE should make sense to any observer, human or otherwise, whence it ought to be a formal theory devoid of 'human baggage'; consequently the TOE (and hence the Universe it specifies) is a purely mathematical structure. While this argument can obviously be challenged – it is entirely possible that pure mathematics is itself a form of human baggage and that the concept 'mathematical structure' has no meaning to creatures whose brains have evolved differently to our own – Tegmark shows that it entails a surprisingly wide range of consequences, but interestingly, these do *not* include computability. Rather, Tegmark introduces an additional *Computable Universe Hypothesis*, according to which the relations describing the Universal structure can be implemented as halting computations. This is similar to Schmidhuber's model, except that it is the relationships between objects that are deemed computable, rather than their evolution through time.

2.2. Examples of physical hypercomputation

A key feature of the digital physics models described above (as well as others, e.g. Zizzi's loop quantum gravity model [22]) is that the models take the assumption of an information- or computation-based universe as their *starting point*, and then ask what consequences follow. This is inevitable, since the authors are ultimately interested in identifying experiments which might provide evidence in support of (or which falsify) their models. Clearly, however, if experiments are to distinguish between digital physics and 'conventional wisdom', it must first be necessary that digital physics and the standard model are not equivalent. It follows, therefore, that digital physics cannot tell us about the feasibility or otherwise of hypercomputation in 'standard' quantum theory.

Unfortunately, this is precisely the question we wish to answer. Rather than invent a *new* model of physics that is computational by fiat, we wish to determine whether the *standard* model is computational. Our approach, which we outline in some detail in sections 3 and 4, is to reformulate the existing model in such a way that its computational nature becomes intuitively obvious. Before doing so, however, we should explain why this task is worth undertaking – as Zuse put it, "Is Nature digital, analog or hybrid? And is there essentially any justification for asking such a question?" [14]

2.2.1. Newtonian models

It is not often appreciated that standard Newtonian physics is already complex enough to support both super-Turing and hypercomputational behaviours, but as Xia has shown, the Newtonian n-body problem exhibits 'non-collision singularities', solutions in which massive objects can be propelled to infinity in finite time [23]. This is particularly problematic for those models of digital physics which claim the Universe is generated by essentially local interactions, like those connecting processes in a cellular automaton, because the laws of physics are typically considered to be time-reversible. Consequently, if a particle can be propelled *to* infinity in finite time, it should also be possible for a particle to arrive *from* infinity in finite time. Clearly, however, there is no earliest time at which such an emerging particle first arrives in the Universe (the set of times at which the emerging particle exists does not contain its greatest lower bound). Consequently, if all objects in the Universe have finite extent and finite history, the particle's 'emergence at infinity' must involve some non-local form of interaction between infinitely many of these objects. On the other hand, Xia's model depends implicitly on an idealised version of Newtonian physics, in which gravitationally bound objects can approach arbitrarily closely (some such idealisation is unavoidable, as the system needs to supply unbounded kinetic energy to the escaping object as it accelerates away to infinity). While this means that Xia's result doesn't actually undermine the case for digital physics, it reminds us that the situation is considerably more complicated than might at first appear.

A recent series of investigations, reported in Beggs *et al.* [24], concerns a collision-based computational system called the *Scatter Machine Experiment* (SME), in which a projectile is fired from a cannon at an inelastic wedge in such a way that it bounces into a detector either to one side (*up*) of the apparatus or the other (*down*); if the projectile hits the vertex, various scenarios can be posited. The wedge is fixed in position with its vertex at some height x whose binary expansion we wish to compute. The cannon can also be moved up and down, but whereas x can take any real value, we only allow the cannon to be placed at heights u which can be expressed in the form $u = m/2^n$ for suitable m and n. By repeatedly firing and then re-aligning the cannon, they attempt to compute the binary expansion of x, one digit at a time. The class of sets which are decidable in polynomial time, when a certain protocol is used to run the SME, is exactly $P/poly$ (the complexity class of languages recognized by a polynomial-time Turing machine with a polynomial-bounded advice function). Since $P/poly$ is known to contain recursively undecidable languages [25], it follows that the scatter machine experiment – despite its evident simplicity – is behaving in a hypercomputational way.

2.2.2. Relativistic models

The SAD_n hierarchy is a family of computational models which exploit the properties of certain singularities in *Malament-Hogarth* spacetimes [26]. These are singularities with computationally useful properties; in particular, if a test particle falls into the singularity, it experiences infinite proper time during its journey; but an outside observer sees the entire descent occurring in finite time. By exploiting such a singularity, we can easily solve the Halting Problem. For suppose we want to know whether some program P halts. We set it running on a computer, and then send that computer into the singularity. From our vantage point, the entire process lasts just a finite length of time, say T seconds. From the computer's point of view the descent takes forever, so if P is going to halt, it will have enough time to do so. We therefore program the computer's operating system so that, if P halts, a rocket is launched back out of the singularity (this is possible for this kind of singularity) so as to arrive at some previously determined place and time (sufficiently later than T seconds from now). We then travel to the rendezvous point. If a rocket arrives at the scheduled time, we know that P must have halted. If no rocket arrives, we know that the operating system never had cause to launch it, and we conclude that P ran forever.

Hogarth refers to this hypercomputational system as an SAD_1 computer; it uses a standard Turing machine to run the underlying program P, but gains hypercomputational power from the geometrical properties of the spacetime in which that Turing machine finds itself. If we now repeat the construction, but this time using an SAD_1 computer in an attempt to decide some question, the resulting (SAD_1 + singularity) system is called an SAD_2 machine, and so on. Finally, by dovetailing a sequence of machines, one from from each level of the hierarchy, and sending the whole lot into a singularity, we obtain an AD machine. The SAD_n machines decide precisely those first order statements which occupy the n^{th} of the Arithmetic Hierarchy, while the AD machine can decide the whole of arithmetic [27].

2.2.3. Quantum theoretical models

Quantum mechanics is, perhaps, mankind's most impressive scientific achievement to date; it enables us to predict various physical outcomes with remarkable accuracy across a wide range of both everyday and exotic situations. In addition, as *It from Bit* demonstrates, there are clear parallels between quantum theory and information theory; since computation is largely seen as the study of information processing, it is not surprising that the field has proven fertile ground for researchers in both digital physics and hypercomputation theory.

One possible hypercomputational model in quantum theory is Kieu's adiabatic quantum algorithm for deciding Hilbert's Tenth problem, concerning the solution of Diophantine equations. Since this problem is known to be recursively undecidable [28], Kieu's algorithm – essentially a method for searching infinite sets in finite time – must be hypercomputational. Although Kieu's claims are controversial and his algorithm have been disputed by various authors, he has sought to address these criticisms in a forthcoming paper [29]. For the time being, therefore, the jury is out.

3. The Standard Path-Integral Formulation

As we explained in section 2.2, we aim to reformulate the standard version of quantum theory from first principles in such a way that its computational aspects become essentially self-evident. We begin by recapitulating Feynman's (non-relativistic) path-integral formulation presented in [30, §§3–4]. Given initial and final locations $q_I = (x_I, t_I)$ and $q_F = (x_F, t_F)$ (where $t_F > t_I$), the goal of the standard formulation is to determine the amplitude $\phi(q_F, q_I)$ that a particle P follows a trajectory $q_I \to q_F$ lying entirely within some prescribed non-empty open space-time region R. As Feynman shows, this amplitude can then be used to generate a Schrödinger wave-equation description of the system, whence this formulation is equivalent to other standard (non-relativistic) models of quantum theory. In Section 4, we will develop a generalised finitary formulation of the same amplitude, and show that it is equivalent to the standard path-integral formulation presented below.

For the sake of illustration, we shall assume that space is 1-dimensional, so that spatial locations can be specified by a single coordinate x — the extension to higher dimensions is straightforward. Furthermore, we shall assume in this paper that the region R is a simple rectangle of the form $R = X \times T$, where X and $T = (t_{\min}, t_{\max})$ are non-empty open intervals in $\mathbb{R}$; this does not limit our results, because open rectangles form a base for the standard topology on $\mathbb{R}^2$, and all of our formulae are derived via integration.[2]

Suppose, then, that a particle P is located initially at $q_I = (x_I, t_I)$, and subsequently at $q_F = (x_F, t_F)$, and that its trajectory from q_I to q_F is some continuous path lying entirely within the region $R = X \times T$. Choose some positive integer ν, and split the duration $\delta t = t_F - t_I$ into $\nu + 1$ equal segments: for $n = 0, \ldots, \nu + 1$, we define $t_n = t_I + {}^{n\delta t}/_{(\nu+1)}$, so that $t_0 = t_I$ and $t_{\nu+1} = t_F$. We write $x_0, \ldots, x_{\nu+1}$ for the corresponding spatial locations, and define $q_n = (x_n, t_n)$. While each of the values x_n can vary from path to path, the values t_n are fixed. To distinguish this situation from the situation below (where t_n is allowed to vary), we shall typically write $q^\dagger = (x, t^\dagger)$ for those locations q_n whose associated t_n-value is fixed (the points q_I and q_F are assumed to be fixed throughout). We will also sometimes write $[\,q^\dagger\,]$ or $[\,q_1^\dagger, \ldots, q_\nu^\dagger\,]$ for the arbitrary path $q_I = q_0^\dagger \to q_1^\dagger \to \cdots \to q_\nu^\dagger \to q_{\nu+1}^\dagger = q_F$. Apart from the fixed values $x_0 \equiv x_I$ and $x_{\nu+1} \equiv x_F$, each of the x_n is constrained only by the requirement that $x_n \in X$, whence the path $[\,q^\dagger\,]$ has ν degrees of freedom.

In classical physics, the *action* associated with a path p is given by $S = \int_p L\, dt$, where the function $L = L(x(t), \dot{x}(t))$, the *Lagrangian*, is a function of position x and velocity $\dot{x}$, only. However, to form this integral we need to specify the motion of the particle in each subinterval $(t_n^\dagger, t_{n+1}^\dagger)$, so we assume that P follows some path $q_n^\dagger \to q_{n+1}^\dagger$ that is classically permissible. Each segment $q_n^\dagger \to q_{n+1}^\dagger$ of the path has associated classical action $S(q_{n+1}^\dagger, q_n^\dagger)$, and probability amplitude $\left\langle q_{n+1}^\dagger \,\middle|\, q_n^\dagger \right\rangle$ defined for all q and (subsequent) q' by $\langle q' \,|\, q\rangle = \exp\{iS(q', q)/\hbar\}$. The action S is determined by the classical *Principle of Least Action.* This says that the classical path is one which minimises this action, so that $S(q', q) = \min \int_t^{t'} L\, dt$. The total action associated with the

[2] Integrating over a union of disjoint rectangles is the same as summing the component integrals: given any integrable function $f(x,t)$ defined on a disjoint union $R = \bigcup_\alpha R_\alpha$, we have $\int_R f = \sum_\alpha \int_{R_\alpha} f$.

path is $S[\,q_1^\dagger, \ldots, q_\nu^\dagger\,] = \sum_n S(q_{n+1}^\dagger, q_n^\dagger)$ and the associated amplitude is the product $\left\langle q_F \,\middle|\, q_\nu^\dagger \right\rangle \left\langle q_\nu^\dagger \,\middle|\, q_{\nu-1}^\dagger \right\rangle \ldots \left\langle q_2^\dagger \,\middle|\, q_1^\dagger \right\rangle \left\langle q_1^\dagger \,\middle|\, q_I \right\rangle$. Summing over all such paths now yields the composite amplitude

$$\phi_\nu(q_F, q_I) = \frac{1}{A_\nu} \int \left\langle q_F \,\middle|\, q_\nu^\dagger \right\rangle dx_\nu \left\langle q_\nu^\dagger \,\middle|\, q_{\nu-1}^\dagger \right\rangle dx_{\nu-1} \ldots \left\langle q_2^\dagger \,\middle|\, q_1^\dagger \right\rangle dx_1 \left\langle q_1^\dagger \,\middle|\, q_I \right\rangle \qquad (1)$$

where A_ν is a normalisation factor. All that remains is to take the limit as $\nu \to \infty$, subject to the assumption that the resulting path $x = x(t)$ is continuous. This gives us the required amplitude $\phi(x_F, x_I)$ that the particle travels from q_I to q_F by a trajectory that lies entirely[3] within R:

$$\phi(q_F, q_I) = \lim_{\nu\to\infty} \frac{1}{A_\nu} \int \left\langle q_F \,\middle|\, q_\nu^\dagger \right\rangle dx_\nu \left\langle q_\nu^\dagger \,\middle|\, q_{\nu-1}^\dagger \right\rangle dx_{\nu-1} \ldots \left\langle q_2^\dagger \,\middle|\, q_1^\dagger \right\rangle dx_1 \left\langle q_1^\dagger \,\middle|\, q_I \right\rangle \ .$$

4. A Finitary Formulation

In section 3 we showed how the amplitude $\phi(q_F, q_I)$, that the particle P travels from q_I to q_F along some path lying entirely within the non-empty open spacetime region $R = X \times T$, is given by $\phi = \lim_{\nu\to\infty} \phi_\nu$. If we now write

$$\Delta_n = \phi_n - \phi_{n-1} \ , \qquad (2)$$

it follows from the identity $\phi_\nu = (\phi_\nu - \phi_{\nu-1}) + \cdots + (\phi_1 - \phi_0) + \phi_0$ that

$$\lim_{\nu\to\infty} \phi_\nu = \lim_{\nu\to\infty} \left(\phi_0 + \sum_{n=1}^{\nu} \Delta_n \right) = \phi_0 + \sum_{n=1}^{\infty} \Delta_n \ .$$

This replacing of a limit with a sum is a key feature of our model, since it allows us to describe a system in terms of a set of mutually distinct finite sets of observations. We can think of this sum in terms of *correction factors*. For, suppose you were asked to estimate the amplitude $\phi(q_F, q_I)$ that some object or particle P will be observed at q_F, given that it had already been observed at q_I and was constrained to move within the region R. With no other information to hand, your best bet would be to assume that P follows some action-minimising classical path, and so the estimate you give is the associated amplitude $\langle q_F \,|\, q_I \rangle$. Some time later, you realise that one or more observations may have been made on the particle while it was moving from q_I to q_F, and that this would have perturbed the amplitude. To take account of these possibilities, you add a series of correction factors to your original estimate; first you add Δ_1 in case 1 observation had taken place, instead of the 0 observations you had originally assumed. Then you add Δ_2 in case there were actually 2 observations, and so on. Each Δ_n takes into account the extra information acquired by performing n observations instead of $n-1$, and since the overall estimate needs to take all of the corrections into account, we have $\phi = \phi_0 + \sum \Delta_n$.

The simple truth, however, is that *continuous motion cannot be observed*, because making an observation takes time. The best we can ever do is to make a series of distinct

[3] Strictly, only the internal points of the trajectory are required to lie in R. Either (or both) of the endpoints q_I and q_F can lie outside R, provided they are on its boundary.

measurements showing us where an object was at finitely many closely-spaced instants $t_1, t_2, \dots, t_\nu$ during the relocation from q_I to q_F. The classical spirit within us then tells us to extrapolate these discrete points into a continuous curve (namely, that path which 'best' joins the points). It is as if we draw the individual locations on celluloid, and then play a mental film projector to give ourselves the comfortable impression of continuous movement. But this mental film projector — represented in the standard formulation by the construction of $\lim \phi_\nu$ — is no part of physical observation; it represents instead an *assumption* about the way the world 'ought to be'. All we can truthfully say is that the object was at such and such a location x_n when we observed it at time t_n, and was subsequently at location x_{n+1} at time t_{n+1}. Regardless of underlying reality (about which we can say virtually nothing), the *observed* universe is inherently discrete. We can ask ourselves how the motion appears if no observations are made; the composite answer, taking into account all potential observers, is given by some amplitude ψ_0. If we ask how it appears if precisely ν observations are made during the relocation from q_I to q_F, we get another amplitude ψ_ν. Since these possibilities are all mutually exclusive, and account for every possible finitely observed relocation from q_I to q_F, the overall amplitude that the relocation happens is the sum of these amplitudes, namely some function $\psi = \sum \psi_\nu$.

Although they both involve infinite sums, these two descriptions are very different, because ψ_n tells us the amplitude for a path with a specific number of hops, while Δ_n describes what happens when we *change* the number of hops. Nonetheless, prompted by the formal structural similarity of the equations $\phi = \phi_0 + \sum \phi_n$ and $\psi = \psi_0 + \sum \psi_n$, we shall equate the two sets of terms, and attempt to find solutions. By requiring $\psi_0 = \phi_0$ and $\psi_n = \Delta_n$, this will ensure that the description we generate – no matter how unnatural it might appear at first sight — satisfies $\phi = \psi$, whence it describes exactly the same version of physics as the standard formulation.

The surprising feature in what follows is that the description we generate is *not* unnatural. Quite the opposite. To see why, we need to remember that amplitudes are normally given in the form $\phi_n = \exp\{i(S_1 + \cdots + S_n))/\hbar\}$. In very rough terms, we can think of the various S values as being essentially equal, so that $\phi_n \approx \exp\{inS/\hbar\}$. When we compute Δ_n, we are asking how ϕ_n changes when n changes; in other words, we can think of Δ_n in fairly loose terms as a measure of ${}^{d\phi_n}/_{dn}$. Again arguing loosely, we can calculate ${}^{d\phi_n}/_{dn} \approx {}^{iS\phi_n}/_{\hbar}$, and now it becomes clear why equating the two sets of terms works, for in essence, Δ_n is approximately proportional to ϕ_n. Since ψ_n is structurally similar to ϕ_n, in the sense that both measure the amplitude associate with a sequence of jumps, it is not surprising to find a similar relationship holding between Δ_n and ψ_n. Since the equations we form will eventually include integrals with normalisation factors, these factors will effectively absorb any remaining constants of proportionality.

4.1. Paths, Actions and Amplitudes

The standard formulation assumes that each trajectory $x(t)$ is a consistently future-pointing spacetime path; this is implicit in the continuity of the representation $x \equiv x(t)$, which assigns one location to each t in the interval $[t_I, t_F]$. Since our formulation rejects this assumption, we need to provide a different definition for *paths*.

We shall assume the abstract existence of a clock, represented by the integer variable τ, used to indicate the order in which observations occur. Each time the clock ticks, i.e. for each $\tau = 0, 1, 2, \dots$, the particle is observed to exist at some space-time location $q_\tau = (x_\tau, t_\tau)$. We call each transition $q_\tau \to q_{\tau+1}$ a *hop*. A finite sequence of consecutive hops

$q_0 \to \cdots \to q_{\nu+1}$ constitutes a *path.* As before, we take $q_0 = (x_I, t_I)$ and $q_{\nu+1} = (x_F, t_F)$, and consider the properties of an arbitrary path from q_I to q_F via ν intermediate points, all of which are required to lie in the prescribed space-time region $R = X \times T$.

We again write $[\,q_1, \ldots, q_\nu\,]$ for the path $q_I \to q_1 \to \cdots \to q_\nu \to q_F$. However, whereas the intervals $t_{n+1} - t_n$ were formally fixed to have identical duration ${}^{\delta t}/_{(\nu+1)}$, there is no constraint on the temporal separation $t_{\tau+1} - t_\tau$ in the finitary formulation; the path $q_0 \to \cdots \to q_{\nu+1}$ therefore has 2ν degrees of freedom, or *twice* the number in the standard formulation. Notice that we now write q_n rather than $q_n^\dagger$, to show that the value t_n is no longer fixed.

What is not clear at this stage is whether hops need necessarily always be future-pointing. The standard formulation forces this on us through its assumption that some continuous motion $t \mapsto x(t)$ is being observed, but this assumption is no longer relevant. We shall therefore describe two finitary formulations, one in which hops are unidirectional in time, and one in which space and time are treated symmetrically, in that hops can move both forwards and backwards in time as well as space. Both models are related to computation theory, but the second is by far the more interesting, both from a computational, and a physical, point of view. The mathematical distinction between the two models is minor. If time is unidirectional into the future, then $t_{\tau+1}$ must lie in the range $t_\tau < t_{\tau+1} \leq t_{\max}$. Otherwise, it can take any value in T.

In the standard formulation, any unobserved motion from one observation to the next is assumed to be classical, and its amplitude is determined by minimising the classical action S. Since we no longer assume that any such motion exists, we shall simply assume that each hop $q \to q'$ has a *hop amplitude*, denoted $\langle q' \,|\, q \rangle_h$, and that this amplitude (when it is non-zero) is associated with an abstract *hop action*, denoted $s_h(q', q)$, by the formula $\langle q' \,|\, q \rangle_h = e^{i s_h(q',q)/\hbar}$. One of our tasks will be to identify the function s_h.

The amplitude associated with the path $[\,q_1, \ldots q_\nu\,]$ is defined, as usual, to be the product $\langle q_F \,|\, q_\nu \rangle_h \times \cdots \times \langle q_1 \,|\, q_I \rangle_h$. The amplitude computed by summing over all paths of this length will be denoted ψ_n, so that the overall *finitary amplitude* that the particle moves from q_I to q_F along a sequence of hops lying entirely within R is just $\psi(q_F, q_I) = \sum_{n=0}^{\infty} \psi_n$.

4.2. The Finitary Equations

Consider again the formulae giving the amplitude that a particle P follows a path from q_I to q_F that lies entirely within the region R, *subject to the assumption* that q_F occurs later than q_I — the standard formulation isn't defined when this isn't the case. We can write these in the form

$$\phi = \phi_0 + \sum_{n=1}^{\infty} \Delta_n \tag{3}$$

$$\psi = \psi_0 + \sum_{n=1}^{\infty} \psi_n \tag{4}$$

whence it is clear that one particular solution can be obtained by solving the infinite family of equations

$$\psi_0 = \phi_0 \tag{5}$$

$$\psi_n = \phi_n - \phi_{n-1} \quad \text{(i.e. } \psi_n = \Delta_n\text{)} \quad \text{for } n > 0 \tag{6}$$

to find the hop-action s_h. Since the terms ϕ_n and A_n are those of the standard formulation, we shall henceforth assume that S, ϕ_n, Δ_n and A_n are all *known functions* in what follows.

4.3. Solving the Equations

As usual, we shall assume that q_F occurs later than q_I (so that $\phi_n = \phi_n(q_F, q_I)$ is defined for each n). We shall be careful to distinguish locations $q^\dagger = (x, t^\dagger)$ for which the time of observation is fixed in the standard formulation, from those of the form $q = (x, t)$ used in the finitary version, for which the value of t is variable. Note first that (1) can be rewritten to give us a recursive definition of ϕ_n, viz.

$$\begin{aligned} \phi_\nu(q_F, q_I) &= \frac{1}{A_\nu} \int \langle q_F \,|\, q^\dagger_\nu \rangle \, dx_\nu \left\langle q^\dagger_\nu \,\middle|\, q^\dagger_{\nu-1} \right\rangle dx_{\nu-1} \ldots \left\langle q^\dagger_2 \,\middle|\, q^\dagger_1 \right\rangle dx_1 \left\langle q^\dagger_1 \,\middle|\, q_I \right\rangle \\ &= \frac{A_{\nu-1}}{A_\nu} \int \langle q_F \,|\, q^\dagger_\nu \rangle \, dx_\nu \frac{1}{A_{\nu-1}} \int \left\langle q^\dagger_\nu \,\middle|\, q^\dagger_{\nu-1} \right\rangle dx_{\nu-1} \ldots \left\langle q^\dagger_2 \,\middle|\, q^\dagger_1 \right\rangle dx_1 \left\langle q^\dagger_1 \,\middle|\, q_I \right\rangle \\ &= \frac{A_{\nu-1}}{A_\nu} \int \langle q_F \,|\, q^\dagger_\nu \rangle \, \phi_{\nu-1}(q^\dagger_\nu, q_I) \, dx_\nu \end{aligned} \tag{7}$$

and an identical derivation gives ψ_ν in the form

$$\psi_\nu(q_F, q_I) = \frac{B_{\nu-1}}{B_\nu} \int_X \int_{T'} \langle q_F \,|\, q_\nu \rangle_h \, \psi_{\nu-1}(q_\nu, q_I) \, dt_\nu \, dx_\nu \tag{8}$$

where the B_n are normalisation factors, and the integration range T' depends on whether we allow hops to jump backwards in time, or insist instead that they move only forwards (we consider the two cases separately, below).

Using (7) to substitute for ϕ_ν in the definition (2) of Δ_n gives

$$\begin{aligned} \Delta_\nu(q_F, q_I) &= \phi_\nu(q_F, q_I) - \phi_{\nu-1}(q_F, q_I) \\ &= \left[\frac{A_{\nu-1}}{A_\nu} \int \langle q_F \,|\, q^\dagger_\nu \rangle \, \phi_{\nu-1}(q^\dagger_\nu, q_I) \, dx_\nu \right] - \phi_{\nu-1}(q_F, q_I) \ . \end{aligned}$$

The case $\nu = 0$ is worth noting in detail. The amplitudes $\phi_0(q_F, q_I)$ and $\psi_0(q_F, q_I)$ describe the situation in which P moves from q_F to q_I without being observed. In the standard formulation, it is assumed in such circumstances that P follows some classical path for which the action S is minimal, while in the finitary formulation we assume that the particle *hops* directly from q_I to q_F. The amplitudes for these behaviours are $\langle q_F \,|\, q_I \rangle$ and $\langle q_F \,|\, q_I \rangle_h$, respectively. However, we need to remember that ϕ_0 and ψ_0 are defined in terms of their contribution to the *overall* amplitudes ϕ and ψ; it is important, therefore, to include the relevant normalisation factors. We therefore define, in accordance with (1), (3), (4) and (8),

$$\phi_0(q_F, q_I) = \frac{1}{A_0} \langle q_F \,|\, q_I \rangle \qquad \text{and} \qquad \psi_0(q_F, q_I) = \frac{1}{B_0} \langle q_F \,|\, q_I \rangle_h \ ,$$

so that, whenever q_F occurs later than q_I,

$$\langle q_F \,|\, q_I \rangle_h = \sigma \, \langle q_F \,|\, q_I \rangle \tag{9}$$

where

$$\sigma = B_0/A_0 \ .$$

Taking principal logarithms on both sides of (9) now gives

$$s_h(q_F, q_I) = S(q_F, q_I) - i\hbar \log \sigma$$

and if we assume that s_h should be real-valued (the classical action S is always real-valued), then $\log \sigma$ must be a real multiple of i, say $\sigma = e^{i\rho}$ where $\rho \in \mathbb{R}$, whence $|\sigma|^2 = 1$. Consequently, $|\langle q_F \,|\, q_I \rangle_h|^2 = |\langle q_F \,|\, q_I \rangle_h|^2$, and the two formulations assign the same standard and finitary probabilities to the relocation $q_I \to q_F$, whenever this is unobserved and future-directed. Moreover, since

$$s_h(q_F, q_I) = S(q_F, q_I) + \rho\hbar$$

we see that our earlier intuition is essentially confirmed: the hop-action s_h (the best estimate of the path-amplitude, given that no observations will be made) is just the classical action S, though possibly re-scaled by the addition of a constant action of size $\rho\hbar$ (which we can think of as a kind of 'zero-point' action). For the purposes of this paper, the values of ρ and $\sigma = e^{i\rho}$ are essentially arbitrary; we shall leave ρ (and hence σ) an undetermined parameter of the model, in terms of which

$$B_0 = \sigma A_0 \tag{10}$$

and

$$s_h(q_F, q_I) = S(q_F, q_I) + \rho\hbar \quad \text{if } q_F \text{ occurs after } q_I \ . \tag{11}$$

The physical significance of ρ is discussed briefly in Section 4.5, in relation to *null-hops*.

4.4. The Unidirectional Model

If we wish to allow only future-pointing hops — we shall call this the *unidirectional* model — there is little left to do. We know from (5) and (6) that each function ψ_n is defined in terms of the known functions ϕ_0 and Δ_n. It only remains to identify the hop amplitude s_h and the normalisation factors B_n. As explained above, our solutions will be given in terms of the undetermined phase parameter σ.

Since the side-condition on (11) is satisfied, the hop amplitude is given in terms of the classical action by the formula $\langle q' \,|\, q \rangle_h = \sigma \langle q' \,|\, q \rangle = \sigma \exp\{iS(q', q)/\hbar\}$, whenever q' follows q.

To find the normalisation factors, we note first that (10) gives us the value $B_0 = \sigma A_0$ directly. Next, when $\nu > 0$, we observe that, since t_ν must come after $t_{\nu-1}$, the range T' in (8) is the interval $(t_{\nu-1}, t_F)$. Consequently,

$$\begin{aligned} \psi_\nu(q_F, q_I) &= \frac{B_{\nu-1}}{B_\nu} \int_X \int_{t_{\nu-1}}^{t_F} \langle q_F \,|\, q_\nu \rangle_h \, \psi_{\nu-1}(q_\nu, q_I) \, dt_\nu \, dx_\nu \\ &= \frac{\sigma B_{\nu-1}}{B_\nu} \int_X \int_{t_{\nu-1}}^{t_F} \langle q_F \,|\, q_\nu \rangle \, \psi_{\nu-1}(q_\nu, q_I) \, dt_\nu \, dx_\nu \ . \end{aligned} \tag{12}$$

When $\nu = 1$, (12) can be rewritten

$$\begin{aligned}
\psi_1(q_F, q_I) &= \frac{\sigma B_0}{B_1} \int_X \int_{t_I}^{t_F} \langle q_F \,|\, q_1 \rangle \, \psi_0(q_1, q_I) \, dt_1 \, dx_1 \\
&= \frac{\sigma B_0}{B_1} \int_X \int_{t_I}^{t_F} \langle q_F \,|\, q_1 \rangle \, \frac{1}{B_0} \, \langle q_1 \,|\, q_I \rangle_h \, dt_1 \, dx_1 \\
&= \frac{\sigma^2}{B_1} \int_X \int_{t_I}^{t_F} \langle q_F \,|\, q_1 \rangle \, \langle q_1 \,|\, q_I \rangle \, dt_1 \, dx_1
\end{aligned}$$

and, since $\psi_1 = \Delta_1$, this gives us

$$B_1 = \left(\frac{\int_X \int_{t_I}^{t_F} \langle q_F \,|\, q_1 \rangle \, \langle q_1 \,|\, q_I \rangle \, dt_1 \, dx_1}{\Delta_1(q_F, q_I)} \right) \sigma^2 \quad .$$

Finally, for $\nu > 1$, (12) becomes

$$\begin{aligned}
\Delta_\nu(q_F, q_I) &= \psi_\nu(q_F, q_I) \\
&= \frac{\sigma B_{\nu-1}}{B_\nu} \int_X \int_{t_{\nu-1}}^{t_F} \langle q_F \,|\, q_\nu \rangle \, \psi_{\nu-1}(q_\nu, q_I) \, dt_\nu \, dx_\nu \\
&= \frac{\sigma B_{\nu-1}}{B_\nu} \int_X \int_{t_{\nu-1}}^{t_F} \langle q_F \,|\, q_\nu \rangle \, \Delta_{\nu-1}(q_\nu, q_I) \, dt_\nu \, dx_\nu
\end{aligned}$$

and hence B_ν can be defined recursively, as

$$B_\nu = \frac{\sigma B_{\nu-1}}{\Delta_\nu(q_F, q_I)} \int_X \int_{t_{\nu-1}}^{t_F} \langle q_F \,|\, q_\nu \rangle \, \Delta_{\nu-1}(q_\nu, q_I) \, dt_\nu \, dx_\nu \quad .$$

4.5. The Bidirectional Model

Far more interesting is the case where hops are allowed to jump backwards as well as forwards in time. It is important to note that the derivation of B_ν given above for the unidirectional model no longer works, because it relies on using (9) to replace $\langle q_F \,|\, q_\nu \rangle_h$ with $\sigma \langle q_F \,|\, q_\nu \rangle$, and on (6) to replace $\psi_{n+1}(q_\nu, q_I)$ with $\Delta_{n+1}(q_\nu, q_I)$. But our use of (9) assumes that q_F occurs after q_ν, and that of (6) that q_ν comes after q_I, and neither assumption is generally valid in the bidirectional model. Consequently, before we can make progress, we need to decide how $\langle q' \,|\, q \rangle_h$ should be defined when the hop $q \to q'$ moves *backwards* in time.

To address this problem, we recall the standard interpretation of *anti-matter* as 'matter moving backwards in time'. For example, the Feynman diagram in Figure 1 shows how the annihilation of e.g. an electron and a positron (its antiparticle) to form two photons can be interpreted instead as showing an electron that moves forward in time, interacts with the photons, and then returns into the past.

Accordingly, whenever we are presented with a backwards hop by the particle P, we re-interpret it as a *forwards* hop by the appropriate anti-particle, $\overline{P}$. Writing $\overline{S}$ for the classical action associated with the antiparticle $\overline{P}$, we therefore define

$$s_h(q_F, q_I) = \begin{cases} \rho\hbar + S(q_F, q_I) & \text{if } q_I \text{ is earlier than } q_F \text{, and} \\ \rho\hbar + \overline{S}(q_I, q_F) & \text{if } q_I \text{ is later than } q_F. \end{cases} \tag{13}$$

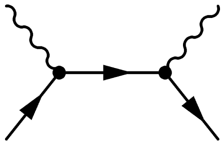


Figure 1: Anti-matter can be thought of as matter moving backwards in time. A particle arrives at bottom left, and the corresponding antiparticle (shown as usual with the arrow reversed) at bottom right; they annihilate to produce two gamma rays, emitted top left and top right. Time advances up the page.

It is tempting to assume that $\overline{S}$ is just the negative of S, but this need not be the case. For example, since photons are their own anti-particles, they would require $\overline{S} = S$. Or consider an electron moving in both an electric and a gravitational field. If we replaced it with a positron, the electric forces would reverse, but the gravitational forces would remain unchanged, and the overall change in action would reflect both effects.

Spatial hops - the physical meaning of σ. What about purely spatial hops that move the particle P sideways in space, without changing its temporal coordinate? There are two cases to consider. If $q_F = q_I$, the particle has not actually moved, and the classical solution $S(q, q) = 0$ holds valid. Consequently, we can simply extend our existing solution by defining $s_h(q, q) = \rho\hbar$, or $\langle q \,|\, q \rangle_h = \sigma$. This, then, explains the physical significance of σ — it is the amplitude associated with the *null hop*, i.e. that hop which leaves the particle in its original location from one observation to the next.

If q_F and q_I differ in their x (but not their t) values, we shall simply take $\langle q_F \,|\, q_I \rangle_h = 0$; i.e. we ban all such hops (this definition is, of course, purely arbitrary, and other definitions may be more appropriate in regard to other investigations[4]; but for our current purposes the specific choice of purely spatial hop action makes little difference, because the paths in question contribute nothing to the integrals we shall be constructing). This doesn't mean, of course, that a path cannot be found from q_I to a simultaneous location q_F — it can, via any past or future location! — but that more than one hop is required to complete the journey. Indeed, the possibility of purely spatial relocations is highly significant, since one could interpret them as explaining quantum uncertainty: one cannot say definitely where a particle is at any given time t, precisely *because* it is able to relocate from one location to another, with no overall change in t.

Solving the Equations. As before, we know from (5) and (6) that each function ψ_n is defined in terms of the known functions ϕ_0 and Δ_n, and it remains to identify the hop amplitude s_h and the normalisation factors B_n. Once again, our solutions will be given in terms of the undetermined phase parameter σ. As always, we assume that $t_I < t_F$, although we allow individual hops to move backwards through time.

To define the hop amplitude, we appeal to (13), and the relationship $\langle q' \,|\, q \rangle_h = e^{is_h(q',q)/\hbar}$. Taken together with our discussion of spatial hops, these allow us to define

[4] For example, suppose we know (from wave-equation methods, say) that P has amplitude $\eta(x)$ to be at location $x^\dagger = (x, t^\dagger)$, for each $x \in X$. A more intuitive solution might then be to take $\langle x^\dagger \,|\, y^\dagger \rangle_h = \eta(x^\dagger)/\eta(y^\dagger)$. This gives $\langle x^\dagger \,|\, x^\dagger \rangle_h = 1$ in agreement with the 'classical amplitude', but also provides information about the relative amplitudes of all other spatial locations at time $t^\dagger$.

s_h fully:

$$\langle q_F \,|\, q_I \rangle_h = \begin{cases} \sigma \, \overline{\langle q_I \,|\, q_F \rangle} & \text{if } q_F \text{ is earlier than } q_I, \\ \sigma \, \langle q_F \,|\, q_I \rangle & \text{if } q_F \text{ is later than } q_I \\ \sigma & \text{if } q_F = q_I, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

where $\overline{\langle q_I \,|\, q_F \rangle} = \exp\{i\overline{S}(q_I, q_F)/\hbar\}$ is the 'classical amplitude' associated with the antiparticle. This idea extends throughout the functions defined in this section; for example, when q' is earlier than q, we write $\overline{\psi_n}(q, q')$ for the amplitude that the antiparticle follows some path $q' \to q$ lying entirely within R. We will see below that the amplitude functions $\psi_n(q', q)$ and $\overline{\psi_n}(q', q)$ are, as one would expect, related to one another in a mutually recursive way.

Now we consider the normalisation constants B_n. We already know that $B_0 = \sigma A_0$, so we consider the case when $n > 0$. Because hops are allowed to move in both directions through time, the integration range T' in (8) is the whole of T. Consequently, (8) becomes

$$\psi_\nu(q_F, q_I) = \frac{B_{\nu-1}}{B_\nu} \int_X \int_T \langle q_F \,|\, q_\nu \rangle_h \, \psi_{\nu-1}(q_\nu, q_I) \; dt_\nu \; dx_\nu \ \ .$$

The integral over T splits into three parts, depending on the value of t_ν relative to t_I and t_F. We have

$$\begin{aligned} \psi_\nu(q_F, q_I) &= \frac{B_{\nu-1}}{B_\nu} \int_X \int_T \langle q_F \,|\, q_\nu \rangle_h \, \psi_{\nu-1}(q_\nu, q_I) \; dt_\nu \; dx_\nu \\ &= \frac{B_{\nu-1}}{B_\nu} \int_X [I_L(x_\nu) + I_M(x_\nu) + I_R(x_\nu)] \, dx_\nu \end{aligned} \tag{14}$$

where $I_L(x_\nu)$ is the integral over $[t_{\min}, t_I]$, $I_M(x_\nu)$ over $[t_I, t_F]$ and $I_R(x_\nu)$ over $[t_F, t_{\max}]$.

When $\nu = 1$, (14) becomes

$$\psi_1(q_F, q_I) = \frac{B_0}{B_1} \int_X [I_L(x_1) + I_M(x_1) + I_R(x_1)] \, dx_1$$

and the integrals I_L, I_M and I_R are defined by

$$\begin{aligned} I_L(x_1) &= \sigma \int_{t_{\min}}^{t_I} \langle q_F \,|\, q_\nu \rangle_h \, \psi_0(q_\nu, q_I) \; dt_\nu = & \frac{\sigma^2}{B_0} \int_{t_{\min}}^{t_I} \langle q_F \,|\, q_1 \rangle \, \overline{\langle q_I \,|\, q_1 \rangle} \, dt_\nu \\ I_M(x_1) &= \sigma \int_{t_I}^{t_F} \langle q_F \,|\, q_\nu \rangle_h \, \psi_0(q_\nu, q_I) \; dt_\nu = & \frac{\sigma^2}{B_0} \int_{t_I}^{t_F} \langle q_F \,|\, q_1 \rangle \, \langle q_1 \,|\, q_I \rangle \;\, dt_\nu \\ I_R(x_1) &= \sigma \int_{t_F}^{t_{\max}} \langle q_F \,|\, q_\nu \rangle_h \, \psi_0(q_\nu, q_I) \; dt_\nu = & \frac{\sigma^2}{B_0} \int_{t_F}^{t_{\max}} \overline{\langle q_1 \,|\, q_F \rangle} \, \langle q_1 \,|\, q_I \rangle \;\, dt_\nu \ \ . \end{aligned}$$

Thus $I_L(x_\nu) + I_M(x_\nu) + I_R(x_\nu) =$

$$\frac{\sigma^2}{B_0} \left[\int_{t_{\min}}^{t_I} \langle q_F \,|\, q_1 \rangle \, \overline{\langle q_I \,|\, q_1 \rangle} \, dt_\nu + \int_{t_I}^{t_F} \langle q_F \,|\, q_1 \rangle \, \langle q_1 \,|\, q_I \rangle \;\, dt_\nu + \int_{t_F}^{t_{\max}} \overline{\langle q_1 \,|\, q_F \rangle} \, \langle q_1 \,|\, q_I \rangle \;\, dt_\nu \right]$$

and

$$\psi_1(q_F, q_I) = \frac{\sigma^2}{B_1}\left[\int_{t_{\min}}^{t_I} \langle q_F \,|\, q_1\rangle \overline{\langle q_I \,|\, q_1\rangle} + \int_{t_I}^{t_F} \langle q_F \,|\, q_1\rangle \langle q_1 \,|\, q_I\rangle + \int_{t_F}^{t_{\max}} \overline{\langle q_1 \,|\, q_F\rangle} \langle q_1 \,|\, q_I\rangle\right] \ .$$

On the other hand, (2) tells us that $\psi_1 = \Delta_1$, and so $B_1 =$

$$\left[\frac{\left(\int_{t_{\min}}^{t_I} \langle q_F \,|\, q_1\rangle \overline{\langle q_I \,|\, q_1\rangle}\ dt_\nu + \int_{t_I}^{t_F} \langle q_F \,|\, q_1\rangle \langle q_1 \,|\, q_I\rangle\ dt_\nu + \int_{t_F}^{t_{\max}} \overline{\langle q_1 \,|\, q_F\rangle} \langle q_1 \,|\, q_I\rangle\ dt_\nu\right)}{\Delta_1(q_F, q_I)}\right] \sigma^2 \ .$$

Finally, when $\nu > 1$, the integrals I_L, I_M and I_R are given by

- $I_L(x_\nu) = \sigma \int_{t_{\min}}^{t_I} \langle q_F \,|\, q_\nu\rangle \overline{\Delta_{\nu-1}(q_I, q_\nu)}\ dt_\nu$;
- $I_M(x_\nu) = \sigma \int_{t_I}^{t_F} \langle q_F \,|\, q_\nu\rangle \Delta_{\nu-1}(q_\nu, q_I)\ dt_\nu$;
- $I_R(x_\nu) = \sigma \int_{t_F}^{t_{\max}} \overline{\langle q_\nu \,|\, q_F\rangle} \Delta_{\nu-1}(q_\nu, q_I)\ dt_\nu$,

and (14) gives us B_ν recursively,

$$\begin{aligned} B_\nu = \frac{\sigma B_{\nu-1}}{\Delta_\nu(q_F, q_I)} \int_X \Bigg\{ & \int_{t_{\min}}^{t_I} \langle q_F \,|\, q_\nu\rangle \overline{\Delta_{\nu-1}(q_I, q_\nu)}\ dt_\nu \\ & + \int_{t_I}^{t_F} \langle q_F \,|\, q_\nu\rangle \Delta_{\nu-1}(q_\nu, q_I)\ dt_\nu + \int_{t_F}^{t_{\max}} \overline{\langle q_\nu \,|\, q_F\rangle} \Delta_{\nu-1}(q_\nu, q_I)\ dt_\nu \Bigg\} \ . \end{aligned}$$

5. Computational Interpretation of the Model

To illustrate the full computational significance of our reformulation (especially the bidirectional version), we first need to digress slightly, and explain Eilenberg's 1974 *X-machine* model of computation [10]. This is an extremely powerful computational model, which easily captures (and extends) the power of the Turing machine. We will then show that a particle's trajectory can be regarded as an X-machine drawn in spacetime, and that (a minor variant of) this machine computes its own amplitude (as a trajectory).

5.1. X-machines

An X-machine $M = F^\Lambda$ (where X is a data type) is a finite state machine F over some alphabet A, together with a *labelling* function $\Lambda\colon a \mapsto a^\Lambda\colon A \to R(X)$, where $R(X)$ is the ring of relations of type $X \leftrightarrow X$.

Each word $w = a_1 \ldots a_n$ in the language $|F|$ recognised by the machine F can be transformed by Λ into a relation w^Λ on X, using the scheme

$$w^\Lambda = {a_1}^\Lambda \circ \cdots \circ {a_n}^\Lambda$$

and taking the union of these relations gives the relation $\left|F^\Lambda\right|$ computed by the machine,

$$\left|F^\Lambda\right| = \bigcup \{\ w^\Lambda \ |\ w \in |F|\ \} \ .$$

If we want to model a relation of type $Y \leftrightarrow Z$, for data types $Y \neq Z$, we equip the machine with encoding and decoding relations, $E : Y \to X$ and $D : X \to Z$. Then the behaviour computed by the extended machine is the relation $E \circ \left|F^\Lambda\right| \circ D$.

Although the language $|F|$ is necessarily regular, the computational power of the X-machine model is unlimited. For, given any set-theoretic relation $\zeta\colon Y \to Z$, we can compute it using the trivial (2-state, 1-transition)-machine with $X = Y \times Z$, by picking any $z^\dagger \in Z$, and using the labelling $y^E = (y, z^\dagger)$, the decoder $(y, z)^D = z$, and encoder $a^\Lambda = \overline{\zeta}$, where $(y, z^\dagger)^{\overline{\zeta}} = (y, \zeta(y))$. For now, given any $y \in Y$, we have $\left|F^\Lambda\right| = \bigcup \{ a^\Lambda \} = \overline{\zeta}$, and

$$y^{(E \circ \left|F^\Lambda\right| \circ D)} = y^{(E \circ \zeta \circ D)} = (y, z^\dagger)^{(\overline{\zeta} \circ D)} = \bigcup (y, \zeta(y))^D = \zeta(y) \ .$$

5.2. Computation by admissible machines

In our case, all of the path relations we consider will be constant multipliers of the form $f_c\colon z \mapsto zc$, where $c, z \in \mathbb{C}$. The resulting machine behaviour will therefore be a set of such multipliers, and we can meaningfully form their sum (which is again a multiplier). For reasons that will shortly become clear, however, we will restrict attention to those paths which visit each state of the machine at least once. We therefore define the *additive behaviour* of such a machine $M = F^\Lambda$ to be the function $|M|^+$ on $\mathbb{C}$ given by

$$|M|^+(z) = \sum \left\{ w^\Lambda(z) \ \middle| \ w \in |F|\,,\ w \text{ visits each state of } F \text{ at least once } \right\}$$

If M is a machine of this form, we will declare the behaviour of M to be the function $|M|^+$, and speak of M as an *additive X-machine* . Any finitary path $[\,q\,] = q_I \to q_1 \to \cdots \to q_\nu \to q_F$ generates an additive X-machine M_q with state set $\{ q_I, q_1, \ldots, q_\nu, q_F \}$, alphabet $A = \{ h_0, \ldots, h_\nu \}$, and transitions $\{q_n \xrightarrow{h_n} q_{n+1} \mid n = 0, \ldots, \nu\}$. Each transition in the machine is a hop along the path, and is naturally associated with the function ${h_n}^\Lambda = \lambda z.(z.\, \langle q_{n+1} \,|\, q_n \rangle_h) : \mathbb{C} \to \mathbb{C}$ that multiplies any input amplitude z by the hop amplitude $\langle q_{n+1} \,|\, q_n \rangle_h$. If M_q is an additive X-machine generated by some path $[\,q\,]$ with initial state q_I, final state q_F, and intermediate states in R, we shall say that M is *admissible*, and that $[\,q\,]$ *generates* M. We claim that each path computes its own amplitude, when considered as the machine it generates.

Computation by the unidirectional model.. For unidirectional machines, each hop h_n involves a jump forward in time, so the states $\{ q_n \}$ must all be distinct, and the path $[\,q\,]$ forms a future-pointing chain through spacetime. Consequently, the machine M_q recognises precisely one string, and the additive and standard behaviours of the X-machine are identical. The function computed by this path maps each $z \in \mathbb{C}$ to

$$z^{\left[({h_0}^\Lambda) \circ \cdots \circ ({h_\nu}^\Lambda)\right]} = z \times \langle q_{n+1} \,|\, q_n \rangle_h \, \langle q_n \,|\, q_{n-1} \rangle_h \cdots \langle q_1 \,|\, q_0 \rangle_h = z \times \psi[\,q\,] \ . \qquad (15)$$

As claimed, therefore, each (unidirectional) trajectory directly computes its own contribution to the amplitude of any path containing it.

Computation by the bidirectional model.. Equation (15) holds also for unidirectional paths in bidirectional machines, but the general physical interpretation is more complicated, because of the possibility of loops. Essentially, we need to distinguish carefully between two related questions, viz.

- what is the amplitude that the path $[\,q\,]$ is traversed?
- what is the amplitude that the path $[\,q\,]$ is *observed* to have been traversed?

To see why, let us suppose that the path $[\,q\,]$ contains only one loop, and that m is minimal such that $q_{m+1} = q_{n+1}$ for some n satisfying $m < n$; write the associated sequence of hops as a concatenation of three segments, viz. $h_0 \dots h_\nu = u.v.w$, where $u = h_0 \dots h_m$, $v = h_{m+1} \dots h_n$ and $w = h_{n+1} \dots h_\nu$. Since v represents a spacetime loop from q_{m+1} back to $q_{n+1} = q_{m+1}$, there is no observable difference between any of the paths $u.v^j.w$, for $j \geq 1$. Consequently, while the amplitude for the path $[\,q\,]$ is just $\psi[\,q\,]$, the amplitude that this path is *observed* is instead the amplitude $\psi^*[\,q\,] = \sum_{j=1}^\infty \psi[\,u\,] \times (\psi[\,v\,])^j \times \psi[\,w\,]$.

More generally, given the machine F generated by any bidirectional trajectory $[\,q\,]$, and any two strings α, β which are recognised by F, *and which visit each state at least once*, there will be no observable difference between α and β. Consequently, if we define

$$F^+ = \left\{\, w^\Lambda \;\middle|\; w \in |F|\,,\ w \text{ visits each state at least once } \right\}$$

then the amplitude ψ^+ that $[\,q\,]$ is *observed* to have been the path traversed will satisfy, for $z \in \mathbb{C}$,

$$z.\psi^+ = \sum \left\{\, w^\Lambda(z) \mid w \in F^+ \,\right\} = \left|F^\Lambda\right|^+(z)$$

and once again, if we think of $[\,q\,]$ as an additive X-machine, it computes its own contribution to the amplitude of any path containing it.

6. Concluding Arguments

Recall that an additive X-machine M is *admissible* provided there is some finitary bidirectional path $[\,q\,]$ that generates it. Say that two paths $[\,q\,]_1$ and $[\,q\,]_2$ are *equivalent*, provided they generate precisely the same admissible machine M. Clearly, this *is* an equivalence relation, and given any path $[\,q\,]$, there will some equivalence class $\widetilde{q}$ containing it. Moreover, the amplitude $|M|^+$ is given by summing the amplitudes of the various paths in $\widetilde{q}$. Consequently, summing over all paths is the same as summing over all admissible machines, so that (regarding $\psi(q_F, q_I)$ as a multiplier),

$$\psi(q_F, q_I) = \sum \left\{\, |M|^+ \;\middle|\; M \text{ is admissible } \right\} \quad ,$$

and $\psi(q_F, q_I)$ can be regarded as integrating all of the admissible machine amplitudes. In the bidirectional formulation, then, the nature of motion in quantum theory reveals itself to be inherently computational. It is not that trajectories can be computed; rather, they *are* computations. As a particle hops through spacetime, it simultaneously *constructs* and *executes* a computational state machine, and the amplitude computed by this machine is precisely the amplitude of the trajectory that constructed it.

In section 2.1, we noted how digital physics assumes the existence of a computation that computes each universe's history, which suggests that the 'computer' which executes the computation is somehow external to the universes being constructed. In contrast, the bidirectional model is telling us that each universe is a *process*, in which each trajectory is a sub-process which computes its own amplitude. Moreover, all of these sub-processes interact with one another non-locally, because hop amplitudes are based on the classical action, and this in turn depends on the ever-changing spacetime distribution of the other particles. In other words, as we have argued elsewhere, quantum theory is best thought of, not in terms of computation, but in terms of *interactive formal processes* [31].

Clearly, this idea has echoes of *It from Bit*, and indeed the bidirectional model helps explain Wheeler's delayed-choice experiment. The apparent paradox relies on two assumptions concerning the experimental set-up. First, the photon must pass through the barrier in order to be observed on the other side; and second, we can reliably identify a time by which the photon has travelled beyond the barrier (we need to make our delayed choice after this time). Both of our reformulations refute the first assumption (the discontinuous nature of hop-based motion means that the Intermediate Value Theorem cannot be invoked to prove that the trajectory necessarily passes through the barrier), while the bidirectional model also refutes the second assumption, since there is no reliable sense in which the decision can be said to have been made 'after' the trajectory intersects the barrier. Thus the delayed-choice experiment contains no paradox, and there is nothing to explain.

We should also be clear as to what our reformulation does *not* say. Throughout this discussion we have focussed on the computational nature of trajectories, but it should be stressed that there is an important distinction to be be drawn between what a process *does*, and how that process is *structured*. This is the same distinction as that highlighted in section 2.1 between Schmidhuber's and Tegmark's versions of the computational universe hypothesis: whereas Schmidhuber considers process evolutions to be computable, Tegmark requires instead that their descriptions be computable. In our case, while we know that each trajectory computes its amplitude, we cannot say that the amplitude itself is necessarily 'computable' in the Turing sense, because we cannot as yet identify the extent to which the two forms of computation are related. As a process, each trajectory is computational, but the values it manipulates need not be.

6.1. Open questions

1. Clearly, we need to determine the relationship between trajectory computations and Turing computations. There must certainly be some such relationship, because the underlying X-machine model is closely related to the Finite State Machine, which also underpins the basic structure of the Turing machine.

2. Although we have exchanged continuous motion for motion based on discrete hops, we have not as yet done away with continuous spaces in their entirety, because many of the expressions given in this paper make use of integration. As we argued above, continuity is not directly observable, so we would prefer a purely discrete model. We should therefore investigate the extent to which the formulation presented here can be re-expressed in purely formal terms, for example using the π-calculus (a standard theoretical vehicle for modelling mobile distributed process-based systems).

3. Suppose we impose the condition that whenever a particle hops inside some arbitrary region (which we can think of as the interior of an event horizon), it cannot hop back out again. This will have a global influence upon trajectory amplitudes, because every journey would otherwise have the option to include hops that pass through the excluded region. In particular, the observed positions of geodesics (which can be modelled in terms of trajectories) can be expected to change position, whence the presence of the excluded region will generate a perceived 'warping' of spacetime geometry. Does this warping agree with the warping predicted by General Relativity? Can the bidirectional model be extended to give a model of quantum gravity?

References

[1] S. C. Kleene. *Introduction to Metamathematics.* North Holland, 1952.

[2] R. Gandy. Church's Thesis and Principles for Mechanisms. In J. Barwise, H.J. Keisler, and K. Kunen, editors, *The Kleene Symposium*, pages 123–148. North-Holland, Amsterdam, 1980.

[3] B. Jack Copeland. The Church-Turing Thesis. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Stanford University, Fall 2002. Online publication, retrieved 4 July 2008 from `http://plato.stanford.edu/archives/fall2002/entries/church-turing/`.

[4] S. Wolfram. *A New Kind of Science.* Wolfram Media, 2002. ISBN 1-57955-008-8.

[5] S. Lloyd. A theory of quantum gravity based on quantum computation. Preprint, retrieved 4 July 2008 from `arXiv:quant-ph/0501135`, 26 Apr 2006.

[6] M. Tegmark. The Mathematical Universe. *Foundations of Physics*, 38(2):101–150, 2008. Preprint available online, retrieved 4 July 2008 from `arXiv:0704.0646[gr-qc]`.

[7] M. Stannett. The case for hypercomputation. *Applied Mathematics and Computation*, 178:8–24, 2006.

[8] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In S. Goldwasser, editor, *Proc. 35th Annual Symp. on Foundations of Computer Science*, page 124. IEEE Computer Society Press, Los Alamitos, CA, 1994. Revised version (1995) available online, retrieved 4 July 2008 from `arXiv:quant-ph/9508027v2`. Subsequently published as *SIAM J. Sci. Statist. Comput.*, 26:1484, 1997.

[9] D. Deutsch. Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proc. Royal Society, Series A*, 400:97–117, 1985.

[10] S. Eilenberg. *Automata, Languages, and Machines, Vol. A.* Academic Press, London, 1974.

[11] M. Stannett. Computation and Hypercomputation. *Minds and Machines*, 13(1):115–153, 2003.

[12] J. Schmidhuber. A Computer Scientist's View of Life, the Universe, and Everything. In C. Freksa, editor, *Foundations of Computer Science: Potential – Theory – Cognition*, volume 1337 of *Lecture Notes in Computer Science*, pages 201–208. Springer, 1997.

[13] K. Zuse. Rechnender Raum. *Elektronische Datenverarbeitung*, 8:336–344, 1967. Available online, retrieved 4 July 2008 from `ftp://ftp.idsia.ch/pub/juergen/zuse67scan.pdf`. This is a preliminary summary of [14].

[14] K. Zuse. *Rechnender Raum.* Schriften zur Datenverarbeitung, Band 1. Friedrich Vieweg & Sohn, Braunschweig, 1969. English translation: Calculating Space, MIT Technical Translation AZT-70-164-GEMIT, MIT (Proj. MAC), Cambridge, Mass. 02139, Feb. 1970. Available online, retrieved 4 July 2008 from `ftp://ftp.idsia.ch/pub/juergen/zuserechnenderraum.pdf`. A German-language summary appeared in 1967 as [13].

[15] J. Wheeler. Recent Thinking about the Nature of the Physical World. It from Bit. In Rosalind B. Mendell and Allen I. Mincer, editors, *Frontiers in Cosmic Physics; Symposium in memory of Serge Alexander Korff, held on September 13, 1990, in New York, NY*, volume 655 of *Annals of the New York Academy of Sciences*, page 349. Academy of Sciences, New York, NY, 1990.

[16] J. D. Bekenstein. Black Holes and the Second Law. *Lettere al Nuovo Cimento*, 4:737, 1972.

[17] J. D. Bekenstein. Black Holes and Entropy. *Physical Reviews*, D7:2333–2346, 1973.

[18] S. W. Hawking. Black hole explosions? *Nature*, 248:30–31, 1 March, 1974.

[19] L. Susskind and J. Lindesay. *An Introduction to Black Holes, Information and the String Theory Revolution – The Holographic Universe.* World Scientific, 2005.

[20] J. Horgan. Profile: Physicist John A. Wheeler, Questioning the "It from Bit". *Scientific American*, pages 36–37, June 1991.

[21] J. Schmidhuber. Algorithmic Theories of Everything. Technical Report IDSIA–20–00, IDSIA, Galleria 2, 6928 Manno (Lugano), Switzerland, December 2000. Available online, retrieved 4 July 2008 from `arXiv:quant-ph/0011122`.

[22] P. Zizzi. Spacetime at the Planck Scale: The Quantum Computer View. In C. Garola, A. Rossi, and S. Sozzo, editors, *The Foundations of Quantum Mechanics: Historical Analysis and Open Questions – Cesena 2004*, pages 345–358. World Scientific, 2004. Preprint available online, retrieved 4 July 2008 from `arXiv:gr-qc/0304032`.

[23] Z. Xia. The existence of non-collision singularities in the n-body problem. *Annals of Mathematics*, 135(3):411–468, 1992.

[24] E. Beggs, J. F. Costa, B. Loff, and J. V. Tucker. Computational complexity with experiments as oracles. *Proceedings of the Royal Society A*, To appear, 2008. Published online, 24 June 2008.

[25] O. Goldreich. *Computational Complexity: A Conceptual Perspective.* Cambridge University Press, 2008.

[26] M. Hogarth. Does General Relativity Allow an Observer to View an Eternity in a Finite Time? *Foundations of Physics Letters*, 5:173–181, 1992.

[27] M. Hogarth. Deciding Arithmetic Using *SAD* Computers. *The British Journal for the Philosophy of Science*, 55(4):681–691, 2004.

[28] Y. Matiyasevich. *Hilbert's Tenth Problem.* MIT Press, Cambridge, MA, 1993.

[29] T. Kieu. Hypercomputability of Quantum Adiabatic Processes. *International Journal of Unconventional Computing (Special Issue on Hypercomputation)*, 4, In press, 2008.

[30] R. P. Feynman. Space-Time Approach to Non-Relativistic Quantum Mechanics. *Reviews of Modern Physics*, 20(2):367–387, April 1948.

[31] M. Stannett. Physical Hypercomputation. In A. Adamatzky, L. Bull, B. De Lacy Costello, S. Stepney, and C. Teuscher, editors, *Unconventional Computing 2007*, pages 265–288. Luniver Press, 2007.

On the solution of trivalent decision problems by quantum state identification

Karl Svozil*

Institute for Theoretical Physics, Vienna University of Technology, Wiedner Hauptstraße 8-10/136, A-1040 Vienna, Austria

Josef Tkadlec†

Department of Mathematics, Faculty of Electrical Engineering, Czech Technical University, 166 27 Praha, Czech Republic

The trivalent functions of a trit can be grouped into equipartitions of three elements. We discuss the separation of the corresponding functional classes by quantum state identifications.

PACS numbers: 03.67.Lx,03.65.Ud
Keywords: Trits, trivalent decision problems, quantum state identification

I. QUANTUM COMPUTATION BY STATE IDENTIFICATION

One of the advantages of quantum computation (Beals et al., 2001; Bennett et al., 1997; Cleve, 2000; Fortnow, 2003; Gruska, 1999; Mermin, 2007; Nielsen and Chuang, 2000; Ozhigov, 1998) over classical algorithms (Odifreddi, 1989; Rogers, Jr., 1967) is due to the fact that during a quantum computation some classically useful information can be coded in or "spread among" multi-partite coherent states in such a way that certain decision problems can be solved by identifying a quantum state which "globally" contains that solution (Mermin, 2003; Svozil, 2006). Thereby, information about single cases contributing to the solution of the decision problem are not useful for (and even makes impossible) the quantum computation.

In comparison, the *retreival* of information, such as the result of a decision problem, is analogous to the *preparation* and *characterization* of states with the properties corresponding to such a decision problem. In particular, unlike classical physical states, quantum states can also be characterized with respect to propositions and properties not encoded into a *single* quantum, but "spread among" quanta in an entangled multi-partite state (Brukner and Zeilinger, 1999a,b, 2003; Donath and Svozil, 2002; Svozil, 2002, 2004; Zeilinger, 1999). Stated differently (Brukner et al., 2002), the essence of entanglement can be identified by two observations: the finiteness of the amount of information per participating quantum, and the possibility that *"the information in a composite system resides more in the correlations than in properties of individuals."* This is also evident from the fact that entangled states cannot be represented as the product of individual states of the participating quanta (cf. Ref. (Mermin, 2007), Sect. 1.5).

The potentiality to quantum mechanically solve decision problems by considering an appropriate multipartite state is not only present in binary decision problems of the usual type, such as Deutsch's algorithm (Cleve et al., 1998; Deutsch, 1985; Deutsch and Jozsa, 1992; Mermin, 2007; Nielsen and Chuang, 2000). It can be extended to d-ary decision problems on dits. (For the related state determination problem, see Ref. (Zeilinger, 1999), footnote 6, and Ref. (Svozil, 2002).)

In what follows we shall consider as the simplest of such problems the trivalent functions of one trit. We shall group them in three functional classes corresponding to an equipartition of the set of functions into three elements. We then investigate the possibility to separate each of these classes by quantum state identifications (Donath and Svozil, 2002; Svozil, 2002).

A strategy to identify an observable associated with the solution of a decision problem can be implemented via the method of general state identification (Donath and Svozil, 2002; Svozil, 2002, 2004) as follows (Svozil, 2006):

1. Re-encode the behaviour of the algorithm or function involved in the decision problem into an orthogonal set of states, such that every distinct function results in a *single* distinct state orthogonal to all the other ones. Suppose that this is impossible because the number of functions exceeds the number of orthogonal states, then

*Electronic address: svozil@tuwien.ac.at; URL: http://tph.tuwien.ac.at/~svozil
†Electronic address: tkadlec@fel.cvut.cz; URL: http://math.feld.cvut.cz/tkadlec/

(a) one could attempt to find a suitable representation of the functions in terms of the base states; e.g., the generalized Deutsch algorithm in Ref. (Svozil, 2006).

(b) Alternatively, the dimension of the Hilbert space could be increased by the addition of auxiliary Qdbits. The latter method is hardly feasible for general q-ary functions of n dits, since the number of possible functions increases with q^{d^n}, as compared to the dimension d^n of the Hilbert space of the input states. In our case of trivalent ($q = 3$) functions of a single ($n = 1$) trit ($d = 3$), there are 27 such functions on three-dimensional Hilbert space. [For the original Deutsch algorithm computing the parity (constancy or nonconstancy) of the four binary functions of one bit, there are $2^{2^1} = 4$ such functions.]

2. For a one-to-one correspondence between functions and orthogonal states, trivalent decision problems among the 27 trivalent functions of a single trit require three three-state quanta associated with the set of $3^3 = 27$ states corresponding to some orthogonal base in $\mathbb{C}^3 \otimes \mathbb{C}^3 \otimes \mathbb{C}^3$. Then, create three *equipartitions* containing three elements per partition — thus, every such partition element contains nine orthogonal states — such that

   (a) *one of the partitions* corresponds to the solution of the decision problem.

   (b) The other two partitions "complete" the system of partitions such that the set theoretic intersection of any three arbitrarily chosen elements of the three partition with *one element per partition* always yields a *single* base state.

3. Formally, the three partitions correspond to a system of three co-measurable *filter operators* F_i, $i = 1, 2, 3$ with the following properties:

   (F1) Every filter F_i corresponds to an operator (or a set of operators) which generates one of the three equipartitions of the 27-dimensional state space into three slices (i.e., partition elements) containing $27/3 = 9$ states per slice. A filter is said to separate two eigenstates if the eigenvalues are different.

   (F2) From each one of the three partitions of (F1), take an arbitrary element. The intersection of these elements of all different partitions (one element per partition) results in a *single* one of the 27 different states.

   (F3) The union of all those single states generated by the intersections of (F2) is the entire set of states.

4. As the first partition corresponds to the solution of the decision problem, the corresponding first filter operator corresponds to the "quantum oracle" operator solving the decision problem on the set of states corresponding to the different cases or branches involved — one state per case or branch.

Ideally, in order for the above strategy to work in three-dimensional Hilbert space of a single Qtrit, one should find a function g on the set of trivalent functions of a trit "folding" the decision problem into a *single* triple of orthogonal vectors and the one-dimensional subspaces spanned by the vectors; with nine orthogonal vectors per component of this triple. However, as has been already pointed out, because the number of functions may exceed the dimension of the Hilbert space, this task might be impossible. For some decision problem, it might still be possible to find a suitable vector representation for the functional values. Another possibility might be the enlargement of Hilbert space by the inclusion of more auxiliary Qdits.

II. OPTIONS FOR "FOLDING" THE DECISION PROBLEM INTO A SINGLE QTRIT

For the sake of demonstration, let us again consider our example of trivalent functions of a single trit. Formally, we shall consider the functions

$$f\colon \{-, 0, +\} \to \{-, 0, +\}$$

which will be denoted as triples

$$\big(f(-), f(0), f(+)\big) \, .$$

There are $3^{3^1} = 27$ such functions. They can be enumerated in lexicographic order "$- < 0 < +$" as in Table I.

The trits will be coded by elements of some orthogonal base in $\mathbb{C}^3$. Without loss of generality we may take $(1, 0, 0) = |-\rangle$, $(0, 1, 0) = |0\rangle$, $(0, 0, 1) = |+\rangle$.

For a given "quantum oracle" function

$$g\colon \{-, 0, +\} \to \mathbb{C}$$

TABLE I Enumeration of all trivalent functions of a single trit in lexicographic order "$- < 0 < +$."

$f_0:$	$(- - -)$	$f_9:$	$(0 - -)$	$f_{18}:$	$(+ - -)$
$f_1:$	$(- - 0)$	$f_{10}:$	$(0 - 0)$	$f_{19}:$	$(+ - 0)$
$f_2:$	$(- - +)$	$f_{11}:$	$(0 - +)$	$f_{20}:$	$(+ - +)$
$f_3:$	$(-0-)$	$f_{12}:$	$(00-)$	$f_{21}:$	$(+0-)$
$f_4:$	(-00)	$f_{13}:$	(000)	$f_{22}:$	$(+00)$
$f_5:$	$(-0+)$	$f_{14}:$	$(00+)$	$f_{23}:$	$(+0+)$
$f_6:$	$(- + -)$	$f_{15}:$	$(0 + -)$	$f_{24}:$	$(+ + -)$
$f_7:$	$(- + 0)$	$f_{16}:$	$(0 + 0)$	$f_{25}:$	$(+ + 0)$
$f_8:$	$(- + +)$	$f_{17}:$	$(0 + +)$	$f_{26}:$	$(+ + +)$

we represent a function $f\colon \{-,0,+\} \to \{-,0,+\}$ by a linear subspace of $\mathbb{C}^3$ generated by the vector

$$g\big(f(-)\big)\,|-\rangle + g\big(f(0)\big)\,|0\rangle + g\big(f(+)\big)\,|+\rangle\,,$$

i.e., by the vector

$$\big(g(f(-)), g(f(0)), g(f(+))\big).$$

In order to be able to implement the first, re-encoding, step of the above strategy, we will be searching for a function g such that the subspaces representing functions $\{-,0,+\} \to \{-,0,+\}$ are nonzero and form the smallest possible number — ideally only one — of orthogonal triples.

First, let us show that we may find a function g such that we obtain three orthogonal triples of orthogonal vectors, each one of the three triples containing nine triples of the form $\big(f(-), f(0), f(+)\big)$ and associated with cases of the functions f, which can grouped into three partitions of three triples of the form $\big(f(-), f(0), f(+)\big)$. Let the values of g be the $\sqrt[3]{1}$ (in the set of complex numbers). Let us, for the sake of simplicity and briefness of notation, denote $\alpha = \mathrm{e}^{2\pi\mathrm{i}/3} = -\frac{1}{2}(1 - i\sqrt{3})$. Then the values of g are α, $\alpha^2 = \alpha^* = \mathrm{e}^{-2\pi\mathrm{i}/3} = -\frac{1}{2}(1 + i\sqrt{3})$ and $\alpha^3 = 1$. Moreover, $\alpha\alpha^* = 1$ and $\alpha + \alpha^* = -1$. Then, the "quantum oracle" function g might be given by the following table:

x	$-$	0	$+$
$g(x)$	α^*	1	α

and (if we identify '$-$' with '-1' and '$+$' with '$+1$') might be expressed by

$$g(x) = \alpha^x = \mathrm{e}^{2\pi\mathrm{i}x/3}\,.$$

g maps the 27 triples of functions $\big(f(-), f(0), f(+)\big)$ into nine groups of three triples of functions, such that triples within the nine groups are assigned the same vector (except a nonzero multiple) by the scheme enumerated in Table II. In every column we obtain an orthogonal triple of vectors

$$\begin{aligned} t_1 &= \{(1,1,1), (1,1,\alpha), (1,1,\alpha^*)\}\,, \\ t_2 &= \{(1,\alpha,\alpha^*), (1,\alpha,1), (1,\alpha^*,1)\}\,, \\ t_3 &= \{(1,\alpha^*,\alpha), (\alpha,1,1), (\alpha^*,1,1)\}\,. \end{aligned}$$

Moreover, vectors from different orthogonal triples are apart by the same angle ϕ, for which $\cos\phi = \sqrt{3}/3$.

Now, let us prove by contradiction that in general the function g cannot be defined in such a way that we obtain at most two orthogonal triples of subspaces. This implies that g cannot "generate" a *single* triple of orthogonal vectors or subspaces, — with nine different functions $\big(f(-), f(0), f(+)\big)$ per element of that triple — required for the method of computation by state identification in three-dimensional Hilbert space.

For the sake of contradiction, let us suppose that this proposition is false, e.g., that there is a function g such that we obtain at most two orthogonal triples of subspaces.

First, all values $g(-), g(0), g(+)$ are nonzero [if, e.g., $g(-) = 0$ then the vector $\big(g(-), g(-), g(-)\big)$ assigned to the function $(-,-,-)$ is a zero vector]. Hence, we obtain a vector $\big(g(-), g(-), g(-)\big)$ that is a nonzero multiple of the vector $(1,1,1)$.

Second, $g(-), g(0), g(+)$ cannot have the same value (in this case we obtain only one subspace generated by the vector $(1,1,1)$).

TABLE II Enumeration of the map g of all trivalent functions $\big(f(-),f(0),f(+)\big)$ into nine groups of three triples of functions, such that triples within the nine groups are assigned the same vector (except a nonzero multiple).

$(-,-,-)$	$\mapsto (1,1,1)$	$(-,-,0)$	$\mapsto (1,1,\alpha)$	$(-,-,+)$	$\mapsto (1,1,\alpha^*)$
$(0,0,0)$		$(0,0,+)$		$(0,0,-)$	
$(+,+,+)$		$(+,+,-)$		$(+,+,0)$	
$(-,0,+)$	$\mapsto (1,\alpha,\alpha^*)$	$(-,0,-)$	$\mapsto (1,\alpha,1)$	$(-,+,-)$	$\mapsto (1,\alpha^*,1)$
$(0,+,-)$		$(0,+,0)$		$(0,-,0)$	
$(+,-,0)$		$(+,-,+)$		$(+,0,+)$	
$(-,+,0)$	$\mapsto (1,\alpha^*,\alpha)$	$(0,-,-)$	$\mapsto (\alpha,1,1)$	$(+,-,-)$	$\mapsto (\alpha^*,1,1)$
$(+,0,-)$		$(+,0,0)$		$(-,0,0)$	
$(0,-,+)$		$(-,+,+)$		$(0,+,+)$	

Let us show that the vectors assigned to the functions $(-,-,0)$ and $(-,0,0)$ are not orthogonal. Indeed, if $\big(g(-),g(-),g(0)\big)$ and $\big(g(-),g(0),g(0)\big)$ are orthogonal, then they have a zero scalar product $0 = g(-)\,g(-)^* + g(-)\,g(0)^* + g(0)\,g(0)^* = |g(-)|^2 + g(-)\,g(0)^* + |g(0)|^2$ and therefore $g(-)\,g(0)^*$ is a negative real number. Hence $0 = |g(-)|^2 - |g(-)|\cdot|g(0)| + |g(0)|^2 = \big(|g(-)| - \frac{1}{2}\,|g(0)|\big)^2 + \frac{3}{4}\,|g(0)|^2$ and therefore $g(0)=0$ that is impossible.

Let us show that all values $g(-),g(0),g(+)$ are different. Indeed, let, e.g., $g(-)=g(0)$. Since $g(-),g(0),g(+)$ cannot have the same value, we obtain $g(+)\neq g(-)$ and therefore the vectors $\big(g(-),g(-),g(+)\big)$ and $\big(g(-),g(+),g(+)\big)$ are not multiples of the vector $(1,1,1)$ and do not generate the same subspace. Analogously as in the previous paragraph we can show that the vectors $\big(g(-),g(-),g(+)\big)$ and $\big(g(-),g(+),g(+)\big)$ are not orthogonal, hence they do not belong to one orthogonal triple and therefore at least one of these vectors is orthogonal to $(1,1,1)$. Let, e.g., $\big(g(-),g(-),g(+)\big)$ is orthogonal to $(1,1,1)$. Then we obtain a zero scalar product $0 = 2\,g(-)+g(+)$ and therefore the vector $\big(g(-),g(-),g(+)\big)$ is a multiple of $(1,1,-2)$. The subspace making an orthogonal triple with subspaces generated by vectors $(1,1,1)$ and $(1,1,-2)$ is generated by $(1,-1,0)$. But, since all values $g(-),g(0),g(+)$ are nonzero, this subspace is not obtained.

We have shown that the subspaces assigned to functions $(-,-,0)$ and $(-,0,0)$ are not orthogonal and do not coincide (otherwise $g(-)=g(0)$). Hence they do not belong to one orthogonal triple and at least one of them should belong to an orthogonal triple with the space generated by the vector $(1,1,1)$. Let, e.g., $\big(g(-),g(-),g(0)\big)$ is orthogonal to the vector $(1,1,1)$. Then we obtain a zero scalar product $0 = 2\,g(-)+g(0)$. Analogously (using the transformations $(-,0)\to(-,+)$ and $(-,0)\to(0,+)$) we can show that one of the vectors $\big(g(-),g(-),g(+)\big)$ and $\big(g(-),g(+),g(+)\big)$ $\big(\big(g(0),g(0),g(+)\big)$ and $\big(g(0),g(+),g(+)\big)$, resp.) is orthogonal to the vector $(1,1,1)$ and therefore $0=2\,g(-)+g(+)$ or $0=g(-)+2\,g(+)$ ($0=2\,g(0)+g(+)$ or $0=g(0)+2\,g(+)$, resp.). Since all values $g(-),g(0),g(+)$ are different and $0=2\,g(-)+g(0)$, we obtain that $0\neq 2\,g(-)+g(+)$ and $0\neq g(0)+2\,g(+)$. Hence $0=g(-)+2\,g(+)$ and $0=2\,g(0)+g(+)$. The system of equations $0=2\,g(-)+g(0)$, $0=g(-)+2\,g(+)$ and $0=2\,g(0)+g(+)$ has the only solution $g(-)=g(0)=g(+)=0$, which results in a complete contradiction.

III. INCREASING THE DIMENSION OF STATE SPACE BY ADDITIONAL QUANTA

The geometric constraints obtained in the last section can be interpreted as the impossibility to “fold” a decision problem into an appropriate quantum state identification in low-dimensional Hilbert space. As has been mentioned already, this can be circumvented by the introduction of additional quanta, thereby increasing the dimension of Hilbert space. In that way, the functions of a small number of dits can be mapped one-to-one onto orthogonal quantum states. However, this strategy fails for a large number of arguments, since the ratio of the number of q-ary functions of n dits to the dimension of the Hilbert space of n dits $d^{-n}q^{d^n}$ increases fast with growing n.

One possibility of mapping the 27 trivalent functions of one trit into the 27 orthogonal base states of the Hilbert space spanned by three Qtrits is

$$|h\big(f(-)\big)\rangle \otimes |h\big(f(0)\big)\rangle \otimes |h\big(f(+)\big)\rangle\,,$$

with $h = id$ being the identity function. A reversible implementation of this function can be given by

$$\begin{aligned} h:\ & \prod_{x\in\{-,0,+\}} |x\rangle|0\rangle \rightarrow \\ & \rightarrow \prod_{x\in\{-,0,+\}} |x\rangle|0 \oplus h(f(x))\rangle = \\ & \prod_{x\in\{-,0,+\}} |x\rangle|h(f(x))\rangle\,, \end{aligned}$$

where "$\oplus$" stands for modulo-two addition.

For the sake of demonstration, consider the following trivalent decision problem associated with the three triples of vectors t_1, t_2, and t_3 as follows:

> *Given some trivalent function of a single trit $f_i(x)$, $i \in \{0,\ldots,26\}$, $x \in \{-,0,+\}$. Find the triple of vectors t among the three triples t_1, t_2 and t_3, such that $g(f_i) \in t$.*

IV. SUMMARY

In summary we find that, in three-dimensional Hilbert space, we cannot solve the type of trivalent decision problems discussed above by a single query. Such a behaviour has already been observed for the problem to find the parity of an unknown binary function $f : \{0,1\}^k \rightarrow \{0,1\}$ of k bits, which turned out to be quantum computationally hard (Beals et al., 2001; Farhi et al., 1998; Miao, 2001; Orus et al., 2004; Stadelhofer et al., 2005). We conjecture that this hardness increases with the number d of possible states of a single dit.

We have also explicitly discussed a trivalent decision problem which can be interpreted as the solution of a quantum state identification problem.

Acknowledgements

The work was supported by the research plan of the Ministry of Education of the Czech Republic no. 6840770010 and by the grant of the Grant Agency of the Czech republic no. 201/07/1051 and by the exchange agreement of both of our universities.

References

Beals R, Buhrman H, Cleve R, Mosca M and de Wolf R (2001) Quantum lower bounds by polynomials. Journal of the ACM, 48:778–797.

Bennett C H, Bernstein E, Brassard G and Vazirani U (1997) Strengths and weaknesses of quantum computing. SIAM Journal on Computing, 26:1510–1523.

Brukner Č and Zeilinger A (1999a) Malus' law and quantum information. Acta Physica Slovaca, 49(4):647–652.

Brukner Č and Zeilinger A (1999b) Operationally invariant information in quantum mechanics. Physical Review Letters, 83(17):3354–3357.

Brukner Č and Zeilinger A (2003) Information and fundamental elements of the structure of quantum theory. In Castell, L. and Ischebek, O., editors, Time, Quantum and Information, pages 323–355. Springer, Berlin.

Brukner Č, Zukowski M and Zeilinger A (2002) The essence of entanglement. Translated to Chinese by Qiang Zhang and Yond-de Zhang, New Advances in Physics (Journal of the Chinese Physical Society)

Cleve R (2000) An introduction to quantum complexity theory. In Macchiavello, C., Palma, G. and Zeilinger, A., editors, Collected Papers on Quantum Computation and Quantum Information Theory, pages 103–127, Singapore. World Scientific.

Cleve R, Ekert A, Macchiavello C and Mosca M (1998) Rapid solution of problems by quantum computation. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 454(1969):339–354.

Deutsch D (1985) Quantum theory, the Church-Turing principle and the universal quantum computer. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences (1934-1990), 400(1818):97–117.

Deutsch D and Jozsa R (1992) Rapid solution of problems by quantum computation. Proceedings of the Royal Society: Mathematical and Physical Sciences (1990-1995), 439(1907):553–558.

Donath N and Svozil K (2002) Finding a state among a complete set of orthogonal ones. Physical Review A (Atomic, Molecular and Optical Physics), 65:044302.

Farhi E, Goldstone J, Gutmann S and Sipser M (1998) Limit on the speed of quantum computation in determining parity. Physical Review Letters, 81:5442–5444.

Fortnow L (2003) One complexity theorist's view of quantum computing. Theoretical Computer Science, 292:597–610.

Gruska J (1999) Quantum Computing. McGraw-Hill, London.

Mermin N D (2003) From Cbits to Qbits: Teaching computer scientists quantum mechanics. American Journal of Physics, 71:23–30.

Mermin N D (2007) Quantum Computer Science. Cambridge University Press, Cambridge.

Miao X (2001) A polynomial-time solution to the parity problem on an NMR quantum computer.

Nielsen M A and Chuang I L (2000) Quantum Computation and Quantum Information. Cambridge University Press, Cambridge.

Odifreddi P (1989) Classical Recursion Theory, Vol. 1. North-Holland, Amsterdam.

Orus R, Latorre J I and Martin-Delgado M A (2004) Systematic analysis of majorization in quantum algorithms. European Physical Journal D, 29:119–132.

Ozhigov Y (1998) Quantum computer can not speed up iterated applications of a black box. Lecture Notes In Computer Science, 1509:152–159.

Rogers Jr H (1967) Theory of Recursive Functions and Effective Computability. MacGraw-Hill, New York.

Stadelhofer R, Suterand D and Banzhaf W (2005) Quantum and classical parallelism in parity algorithms for ensemble quantum computers. Physical Review A (Atomic, Molecular and Optical Physics), 71:032345.

Svozil K (2002) Quantum information in base n defined by state partitions. Physical Review A (Atomic, Molecular and Optical Physics), 66:044306.

Svozil K (2004) Quantum information via state partitions and the context translation principle. Journal of Modern Optics, 51:811–819.

Svozil K (2006) Characterization of quantum computable decision problems by state discrimination. In Adenier, G., Khrennikov, A. and Nieuwenhuizen, T. M., editors, Quantum Theory: Reconsideration of Foundations - 3, volume 810, pages 271–279. American Institute of Physics.

Zeilinger A (1999) A foundational principle for quantum mechanics. Foundations of Physics, 29(4):631–643.

Unifying computers and dynamical systems using the theory of synchronous concurrent algorithms[1]

B.C. Thompson
Department of Computer Science,
Swansea University, Swansea SA2 8PP, Wales

J.V. Tucker
Department of Computer Science,
Swansea University, Swansea SA2 8PP, Wales
J.V.Tucker@swansea.ac.uk

J.I. Zucker
Department of Computing and Software,
McMaster University, Hamilton, Ont. L8S 4T1, Canada
zucker@mcmaster.ca

Abstract

A *synchronous concurrent algorithm* (SCA) is a parallel deterministic algorithm based on a network of modules and channels, computing and communicating data in parallel, and synchronised by a set of clocks. Many types of algorithms, computer architectures, and mathematical models of spatially extensive physical and biological systems are examples of SCAs. For example, conventional digital hardware is made from components that are SCAs and many computational models possess the essential features of SCAs, including *systolic arrays*, *neural networks*, *cellular automata* and *coupled map lattices.*

In this paper we formalise the general concept of an SCA equipped with a global clock in order to analyse precisely (*i*) specifications of their spatio-temporal behaviour; and (*ii*) the senses in which the algorithms are correct. We start the mathematical study of SCA computation, specification and correctness using methods based on computation on many-sorted topological algebras and equational logic. We show that specifications can be given equationally and, hence, that the correctness of SCAs can be reduced to the validity of equations in certain computable algebras. Since the idea of an SCA is general, our methods and results apply to each of the classes of algorithms and dynamical systems above.

Key words and phrases: synchronous concurrent algorithm; dynamical systems; many sorted algebras; equational specifications; streams; computability on topological algebras; computable physical systems.

[1] This paper is still under review due to late submission

1 Introduction

1.1 The Concept

A *synchronous concurrent algorithm* (SCA) is an algorithm based on a network of modules and channels, computing and communicating data in parallel, and synchronised by a set of clocks. The etymology of 'synchronous' is Greek: "at the same time". SCAs can process infinite streams of input data and return infinite streams of output data. Most importantly, a SCA is a *parallel deterministic algorithm.* The deterministic nature of these algorithms is established by a single, global clock.

Many types of algorithms, computer architectures, and mathematical models of spatially extensive physical and biological systems are examples of SCAs. First and foremost, conventional digital hardware, including all forms of serial and parallel computers and digital controllers, is made from components that are SCAs. In many cases, complete specifications of computers at different levels of abstraction are SCAs. Interestingly, the structure of Charles Babbage's Analytical Engine (developed from 1833 onwards) is that of an SCA.

Further, many specialised models of computation possess the essential features of SCAs, including *systolic arrays*, *neural networks*, *cellular automata* and *coupled map lattices.*

The parallel algorithms, architectures and dynamical systems that comprise the class of SCAs have many applications, ranging from their use in special purpose devices (for communication and signal processing, graphics, and process control instrumentation) to computational models of biological and physical phenomena.

From the point of view of computing, an SCA can be considered to be a type of *deterministic* data flow network, in which time is explicit and enjoys a primary role. SCAs require a new specialised mathematical theory with applications of its own.

From the point of view of mathematical physics and biology, an SCA can be considered to be a type of spatially extensive discrete space, discrete time, deterministic dynamical system that is studied independently or as an approximation to continuous space, continuous time dynamical systems.

In most cases, SCAs are complicated and require extensive simulation and mathematical analysis to understand their operation, behaviour and verification. In fact, in the independent literatures on the above types of SCAs it is often difficult to formulate precisely

(*i*) specific SCAs and their operation in time;

(*ii*) specifications of their spatio-temporal behaviour; and

(*iii*) the senses in which the algorithms are correct.

In the case of neural networks, correctness is further complicated by the difficulty of writing problem specifications, the existence of a learning phase, and notions of approximate correctness. In the case of non-linear dynamical systems, correctness is concerned with properties such as chaotic, stable, and coherent behaviour over time. Thus, SCAs constitute a wide ranging class of useful algorithms for which many basic questions concerning their structure and design remain unanswered.

In this paper we formalise the general concept of an SCA equipped with a global clock and analyse precisely ideas about the specification and correctness of SCAs. Our mathematical study of SCA computation, specification and correctness provides a unified theory of deterministic parallel computing systems and deterministic, spatially extensive, nonlinear dynamical systems.

The methods are based on abstract computability theory on many-sorted topological algebra and equational logic. We show how to define SCAs by equations over stream algebras in a simple way. We also show that specifications can be given equationally and, hence, that the correctness of SCAs can always be reduced to the validity of equations in certain algebras. Thus, a natural method for verification of SCAs is equational reasoning, although this is incomplete.

Our methods and results apply to each of the classes of algorithms and architectures listed above. In particular, they can be used in case studies and software tools for design and verification of specific classes of SCAs, and as a starting point for a general theoretical analysis of hardware verification.

1.2 The theory

Data is modelled by a "T-standard" algebra

$$A \;=\; (A, \mathbb{B}, \mathbb{T};\; F_1 \ldots F_k)$$

with three carrier sets: the set A of data, $\mathbb{B}$ of booleans and $\mathbb{T}$ of naturals $\{0,1,2,\ldots\}$ (written $\mathbb{T}$ instead of $\mathbb{N}$ because it represents the discrete time on the global clock), and functions $F_1, \ldots, F_k$ which include the standard boolean operations (with possibly equality on A) and the arithmetic operations of 0 and successor.

The behaviour of SCAs in time is modelled using *streams* of elements of A, which are infinite sequences indexed by (discrete) time. Let $[\mathbb{T} \to A]$ be the set of all streams. operations on data, time and streams are combined to form a *stream algebra*:

$$\bar{A} \;=\; (A, \mathbb{B}, \mathbb{T}, [\mathbb{T} \to A];\; F_1 \ldots F_k, \mathsf{eval})$$

Typically, in models of hardware systems, SCAs compute with streams of bits, integers or terms. In dynamical systems, SCAs compute with streams of real and complex numbers. To prepare for this mathematical view, we provide some preliminaries on topological algebras in Section 2 and stream algebras and computable algebras in Section 3. We note that all stream algebras are topological algebras and often have certain dense subalgebras that are computable.

In Section 5 we define synchronous concurrent algorithms and architectures and formalise their semantics by means of functions defined by *simultaneous primitive recursion equations* over $\bar{A}$.

More specifically, an SCA based on a network N m modules and p input streams is specified by a network value function

$$\boldsymbol{V}^N : \mathbb{T} \times A^m \times [\mathbb{T} \to A]^p \;\to\; A^m$$

defined by in which $\boldsymbol{V}^N(t, \boldsymbol{a}, \boldsymbol{x})$ denotes the state of the SCA at time $t \in \mathbb{T}$ on processing p input streams $\boldsymbol{x} \in [\mathbb{T} \to A]^p$ from initial state $\boldsymbol{a} \in A^m$.

In Section 5 we consider specifications and correctness criteria for a simple form of the space-time behaviour of SCAs: correctness based on specifications with respect to a single system clock of the SCA. Other forms of correctness are possible, such as correctness based on specifications with respect to a clock external to the SCA [TT91].

In Section 6 we consider the SCA equational models from the point of view of computability theory. We define a class of predicates on $\bar{A}$: *equational PR*, broader than the class of PR predicates.

We consider specifications and correctness relations and impose conditions that they be algorithmically testable e.g., by primitive recursive computations. We prove some results concerning the logical and computational structure of SCA correctness, including results having the following form:

Theorem 1. *The network value function* $\boldsymbol{V}^N$ *is primitive recursive on* A.

Theorem 2 (Computability of correctness specification). *Suppose*

(a) *P, Q and R are equationally* λPR *on* $\bar{A}$,

(b) *A has a dense computable subalgebra D.*

Then we can effectively construct a computable algebra $C_{V,P,Q,R}$ *with signature* $\Sigma_{V,P,Q,R}$ *that expands* $\overline{D}_{\mathsf{reg}}$ *by functions, and equations* e_P, $e_Q, e_{V,R}$ *over* $\Sigma_{V,P,Q,R}$ *such that the following are equivalent:*

(i) $\boldsymbol{V}$ *is correct w.r.t. P, Q and R, i.e.,* (6.3) *holds;*

(ii) $C_{V,P,Q,R} \models e_P \wedge e_Q \to e_{V,R}$.

Thus *the correctness of the SCA* (as in (i)) which through our definitions involves a wide variety of complex space-time behaviours for a wide variety of computing devices and dynamical systems, *can be reduced to the validity of a conditional equation in a computable algebra* (as in (ii)).

This has several consequences, including the fact that SCA correctness is co-recursively enumerable. This suggests *there are no nice complete proof systems for SCA verification.*

However, we do have the following result in this direction.

Theorem 3 (Validity of correctness specification). *Given the hypotheses of Theorem 2, we can effectively construct a finite conditional equational specification* ($\Sigma_{V,P,Q,R}$), $E_{V,P,Q,R}$) *and equations* e_P, $e_Q, e_{V,R}$ *over* $\Sigma_{V,P,Q,R}$ *such that the following are equivalent:*

(i) $\boldsymbol{V}$ *is correct w.r.t. P, Q and R, i.e.,* (6.3) *holds;*

(ii) $T(\Sigma_{V,P,Q,R}, E_{V,P,Q,R}) \models e_P \wedge e_Q \to e_{V,R}$.

Since the emphasis in this paper is on the a general mathematical model of SCAs, it will be helpful if the reader has some familiarity with theory for algorithmic computability on discrete and continuous data [PER89, Wei00, TZ00, TZ04, BT87].

1.3 Origins

The idea of a making a mathematical theory of SCAs that would uncover and analyse common structures and properties between hardware, parallel algorithms, and dynamical systems modelling natural phenomena arises in the work of the second author (JVT) at Leeds University, starting in 1981. Over many years, the SCA notion was developed primarily through studying applications in:

- work with N.A. Harman on hardware design and verification [Har89, HT88b, HT88a, HT90, HT91, HT93, HT96]
- work with A.V. Holden and M.J. Poole on non-linear dynamical systems [HTT90, HTT91, HPTZ92, PTH98, PTH02]

The first author (BCT) and JVT started work on these mathematical foundations for SCA theory in November 1987, leading to the report [TT91]. Although unpublished, it was widely circulated (forming, *e.g.*, part of JVT's lecture notes for the NATO Summer School on *Logic and algebra of specification*, Marktoberdorf, Germany, 1991).

There is a full conceptual analysis and extensive reflection on correctness and examples this report. However, the subtlety of the connections between the SCA models and abstract and concrete computability theories for continuous data types, such as streams of real numbers, was a problem. Thus, a gap of 17 years is partly excused by the need to master computability theories for topological algebras, to which JVT and the third author (JIZ) have devoted many pages in the period [TZ94, TZ99, TZ00, TZ02a, TZ02b, TZ04, TZ05] Our current understanding enabled us to look at continuous time, discrete space systems in our paper [TZ07], where we were motivated by the idea of models capable of unifying disparate analogue technologies. Clearly, this application to analogue computation was inspired by the earlier unification of models work on SCAs.

2 Topological algebras

We briefly survey the basic concepts of topological and metric many-sorted algebras. More details can be found in [TZ00, TZ99, TZ04]

2.1 Basic algebraic definitions

A *signature* Σ (for a many-sorted algebra) is a pair consisting of (i) a finite set $\boldsymbol{Sort}(\Sigma)$ of *sorts*, and (ii) a finite set $\boldsymbol{Func}(\Sigma)$ of *(basic) function symbols*, each symbol F having a *type* $s_1 \times \cdots \times s_m \to s$, where $s_1, \ldots, s_m, s \in \boldsymbol{Sort}(\Sigma)$; in that case we write $F : s_1 \times \cdots \times s_m \to s$. (The case $m = 0$ corresponds to *constant symbols.*)

A *Σ-product type* has the form $u = s_1 \times \cdots \times s_m$ $(m \geq 0)$, where $s_1, \ldots, s_m$ are Σ-sorts.

A *Σ-algebra* A has, for each sort s of Σ, a non-empty *carrier set* A_s of sort s, and for each Σ-function symbol $F : u \to s$, a function $F^A : A^u \rightharpoonup A_s$, where, for the Σ-product type $u = s_1 \times \cdots \times s_m$, we write $A^u =_{df} A_{s_1} \times \cdots \times A_{s_m}$. For $m = 0$, F^A is an element of A_s. (The notation $f : X \rightharpoonup Y$ refers in general to a function from X to Y.)

The algebra A is *total* if F^A is total for each Σ-function symbol F.

Remark 2.1.1 (Assumption of total algebras). for the purpose of this paper, we work only with total algebras, for the sake of simplicity. The interesting generalisation to the framework of partial algebras (with partial operations and partial streams) is left to a future paper.

Given an algebra A, we write $\Sigma(A)$ for its signature.

Examples 2.1.2. (a) The algebra $\mathcal{B}$ of *booleans* has the carrier $\mathbb{B} = \{\mathbb{t}, \mathbb{f}\}$ of sort bool:

$$\mathcal{B} \;=\; (\mathbb{B};\; \mathbb{t}, \mathbb{f}, \mathsf{and}, \mathsf{or}, \mathsf{not})$$

(b) The algebra $\mathcal{T}_0$ of naturals has a carrier $\mathbb{T}$ of sort nat, together with the zero constant and successor function:

$$\mathcal{T}_0 \;=\; (\mathbb{T};\; 0, \mathsf{S})$$

Note that here and elsewhere we use the notation

$$\mathbb{T} \;=_{df}\; \mathbb{N} \;=\; \{\, 0, 1, 2, \dots \}$$

for the set of ***natural numbers*** (denoted $t, t', \dots$), since the interpretation of $\mathbb{N}$ throughout this paper will be almost exclusively as a ***discrete global clock***. Similary we write '$\mathcal{T}_0$', etc.

(c) The ring $\mathcal{R}_0$ of reals has a carrier $\mathbb{R}$ of sort real:

$$\mathcal{R} \;=\; (\mathbb{R};\; 0, 1, +, \times, -).$$

We make the following

Instantiation Assumption. *For every Σ-sort s, there is a closed term of that sort, called the* ***default term $\boldsymbol{\delta}^s$*** *of that sort. In any Σ-algebra A, it names an element of A_s, called the* ***default element of*** *A_s.*

2.2 Adding booleans: Standard signatures and algebras.

Definition 2.2.1 (Standard signature). A signature Σ is *standard* if it includes the signature of booleans, *i.e.*, $\Sigma(\mathcal{B}) \subseteq \Sigma$.

Given a standard signature Σ, a sort of Σ is called an *equality sort* if Σ includes an *equality operator* $\mathsf{eq}_s : s^2 \to \mathsf{bool}$.

Definition 2.2.2 (Standard algebra). Given a standard signature Σ, a Σ-algebra A is *standard* if (i) it is an expansion of $\mathcal{B}$; (ii) the equality operator eq_s is interpreted as *identity* on the carrier of each equality sort s.

An example of an equality sort is the sort nat of naturals, with carrier $\mathbb{T}$. Intuitively, equality is "computable" or "decidable" on $\mathbb{T}$. as an interpretation of eq_s are:

A non-equality sort is the sort real of reals. Intuitively, equality is ("co-semi-decidable", but) *not* (totally) decidable on $\mathbb{R}$.

Any many-sorted signature Σ can be *standardised* to a signature $\Sigma^{\mathcal{B}}$ by adjoining the sort bool together with the standard boolean operations; and, correspondingly, any algebra A can be standardised to an algebra $A^{\mathcal{B}}$ by adjoining the algebra $\mathcal{B}$, together with equality at the equality sorts.

Examples 2.2.4.

(a) A *standard algebra of naturals* $\mathcal{T}$ is formed by standardising the algebra $\mathcal{T}_0$ (Example 2.1.2(b)), with (total) equality and order operations on $\mathbb{T}$:

$$\mathcal{T} \;=\; (\mathcal{T}_0, \mathcal{B};\; \mathsf{eq}_{\mathsf{nat}}, \mathsf{less}_{\mathsf{nat}})$$

(b) The *standardised ring of reals* (*cf.* Example 2.1.2(c)):

$$\mathcal{R} \;=\; (\mathcal{R}_0, \mathcal{B})$$

Note that there is no (total) equality on $\mathbb{R}$, as discused above.

2.3 Adding the naturals: T-standard signatures and algebras.

Definition 2.3.1 (T-standard signature). A signature Σ is *T-standard* if (i) it is standard, and (ii) it contains the standard signature of naturals, *i.e.*, $\Sigma(\mathcal{T}) \subseteq \Sigma$.

Definition 2.3.2 (T-standard algebra). Given an T-standard signature Σ, a corresponding Σ-algebra A is *T-standard* if it is an expansion of $\mathcal{T}$.

Any standard signature Σ can be T-*standardised* to a signature Σ^{T} by adjoining the sort nat and the operations 0, S, $\mathsf{eq}_{\mathsf{nat}}$ and $\mathsf{less}_{\mathsf{nat}}$. Correspondingly, any standard Σ-algebra A can be T-*standardised* to an algebra $\mathcal{A}^{T}$ by adjoining the carrier $\mathbb{T}$ together with the corresponding standard functions.

Throughout this paper, we will assume:

T-standardness Assumption. *The signature Σ, and the Σ-algebra A, are T-standard.*

2.4 Topological algebras.

Definition 2.4.1. (a) A *topological Σ-algebra* is a Σ-algebra with topologies on the carriers such that each of the basic Σ-functions is continuous.

(b) A (T-)*standard topological algebra* is a topological algebra which is also an (T-)standard algebra, such that the carriers $\mathbb{B}$ (and $\mathbb{T}$) have the discrete topology.

Examples 2.4.2. (a) ***Discrete algebras:*** The standard algebras $\mathcal{B}$ and $\mathcal{T}$ of booleans and naturals respectively (§§2.1, 2.2) are topological (total) algebras under the discrete topology. All functions on them are trivially continuous, since the carriers are discrete.

(b) The T-standard *topological total real algebra* $\mathcal{R}^T$ is defined by

$$\mathcal{R}^T \;=\; (\mathcal{R}, \mathcal{T};\, \mathsf{div}_{\mathsf{nat}})$$

where $\mathcal{R}$ is the standardised ring of reals (2.2.4(b)), $\mathcal{T}$ is the standard algebra of naturals (2.2.4(a)), and $\mathsf{div}_{\mathsf{nat}}:\ \mathbb{R} \times \mathbb{T} \to \mathbb{R})$ is the total (continuous!) function defined by

$$\mathsf{div}_{\mathsf{nat}}(x,t) = \begin{cases} x/t & \text{if } \ t \neq 0 \\ 0 & \text{if } \ t = 0 \end{cases}$$

Note that $\mathcal{R}^T$ does *not* contain (total) boolean-valued functions '<' or '=' on the reals, since they are not continuous; nor does it contain division of reals by reals, since that cannot be total and continuous. See [TZ99, TZ04, TZ05] for discussions of these issues.

2.5 Metric algebra

A particular type of topological algebra is a *metric algebra.* This is a many-sorted standard algebra A with an associated metric:

$$A \;=\; (A_1, \ldots, A_r, \mathcal{R};\; F_1^A, \ldots, F_k^A, \mathsf{d}_1^A, \ldots, \mathsf{d}_r^A)$$

where $\mathcal{R}$ is the standardised ring of reals (Example 2.2.4(b)), the carriers A_i are metric spaces with metrics $\mathsf{d}_i^A\colon A_i^2 \to \mathbb{R}$, $(i = 1, \ldots, r)$, $F_1, \ldots, F_k$ are the Σ-function symbols other than $\mathsf{d}_1, \ldots, \mathsf{d}_k$, and the functions F_i^A are all continuous with respect to these metrics. The carriers $\mathbb{B}$ and $\mathbb{T}$ (included among the A_i) are given the *discrete metric*, which induces the discrete topology.

Clearly, metric algebras can be viewed as special cases of topological algebras.

Example 2.5.1. The real algebra $\mathcal{R}^T$ (Example 2.4.3(b)) can be recast as a metric algebra in an obvious way.

3 Stream algebras; Computable algebras

3.1 Adding streams to algebras: Algebras $\bar{A}$ of signature $\overline{\Sigma}$

LetΣ be a T-standard signature, and A a T-standard Σ-algebra. We define an extension of Σ and a corresponding expansion of A.

We choose a set $\boldsymbol{S} \subseteq \boldsymbol{Sort}(\Sigma)$ of *pre-stream sorts*, and then extend Σ^N to a ***stream signature*** $\overline{\Sigma}^{\boldsymbol{S}}$ ***relative to*** $\boldsymbol{S}$, as follows. With each $s \in \boldsymbol{S}$, associate a new ***stream sort*** $\bar{s}$, also written $\mathsf{nat} \to s$. Then

(a) $\boldsymbol{Sort}(\overline{\Sigma}^{\boldsymbol{S}}) \;=\; \boldsymbol{Sort}(\Sigma) \cup \{\bar{s} \mid s \in \boldsymbol{S}\}$;

(b) $\boldsymbol{Func}(\overline{\Sigma}^{\boldsymbol{S}})$ consists of $\boldsymbol{Func}\,(\Sigma)$, together with the *evaluation function*

$$\mathsf{eval}_s : (\mathsf{nat} \to s) \times \mathsf{nat} \to s,$$

for each $s \in \boldsymbol{S}$.

Now we can expand A^T to a $(\overline{\Sigma}^{\boldsymbol{S}})$-***stream algebra*** $\bar{A}^{\boldsymbol{S}}$ by adding for each $s \in \boldsymbol{S}$:

(i) the carrier for $\bar{s}$, which is the set

$$A_{\bar{s}} \;=\; \bar{A}_s \;=\; [\mathbb{T} \to A_s]$$

of all *streams on* A_s *i.e.*, functions $u\colon \mathbb{T} \to A_s$;

(ii) the interpretation of eval_s on A as the function $\mathsf{eval}_s^A : [\mathbb{T} \to A_s] \times \mathbb{T} \to A_s$ which evaluates a stream at a time instant: $\mathsf{eval}_s^A(u,t) = u(t)$.

The algebra $\bar{A}^{\boldsymbol{S}}$ is the ***(full) stream algebra over*** A ***with respect to*** $\boldsymbol{S}$. (We will usually omit explicit reference to the set $\boldsymbol{S}$.)

Note that the Instantiation Assumption does not hold (in general) for the signature of a stream algebra.

3.2 Expanding topological algebras to stream algebras

The algebraic expansion of an algebra A to a stream algebra $\bar{A}$ induces a corresponding topological expansion.

(a) The topological T-standardisation A^T, of signature Σ^T, is constructed from A by giving the new carrier $\mathbb{T}$ the discrete topology.

(b) Next, a topology on A^T can be extended to one on $\bar{A}$ by giving the stream carriers $[\mathbb{T} \to A_s]$ the ***product topology*** based on A_s, where the *basic open sets* have the form

$$U \;=\; \{\, u \in \bar{A}_s \mid u(t_i) \in U_i \;\text{ for } i = 1, \ldots, n \,\} \tag{2.1}$$

for some $n > 0$, $t_1, \ldots, t_n \in \mathbb{T}$ and $U_1, \ldots, U_n$ open subsets of A_s.

With this topology, the operator eval_s^A is continuous.

Remarks 3.2.1.

(1) This topology is the same as the ***inverse limit topology*** on $[\mathbb{T} \to A_s]$ [TZ08, §2.1].

(2) If A_s is ***metrisable*** by the metric d_s, then so is $[\mathbb{T} \to A_s]$ [TZ08, §3.1], by the metric

$$\mathsf{d}_{\bar{s}}(u,v) \;=_{df}\; \sum_{t=0}^{\infty} \min\big(\mathsf{d}_s(u(t), v(t)),\; 2^{-t}\big)$$

3.3 Regular streams

Let B be a Σ-subalgebra of A. Then the stream algebra $\overline{B}$ over B is a $\overline{\Sigma}$-subalgebra of the stream algebra $\bar{A}$. Further, for any stream sort s, if we replace $[\mathbb{T} \to B_s]$ by *any* nonempty subset of it in the definition of $\overline{B}$, then we again obtain a "stream subalgebra" of $\bar{A}$. All subalgebras of $\bar{A}$ are obtained in this way.

Of special interest is the following subset of the set $\bar{A}_s$ of all streams in $\bar{A}$ of sort s. Define the set of ***regular streams of*** A ***of sort*** s by

$$(\bar{A}_s)_{\mathsf{reg}} \;=\; [\mathbb{T} \to A_s]_{\mathsf{reg}} \;=\; \{\, u \in [\mathbb{T} \to A_s] \mid \exists t_0\, \forall t \geq t_0\, (u(t) = \boldsymbol{\delta}^s)\,\}$$

where $\boldsymbol{\delta}^s$ is the *default element* of A_s (2.1).

Further, for each T-standard Σ-algebra A we define $\bar{A}_{\mathsf{reg}}$, the ***regular stream algebra over*** A***, to be the*** $\overline{\Sigma}$***-subalgeba*** of the stream algebra $\bar{A}$ obtained by *restricting*, at each stream sort s, $\bar{A}_s$ to the set $(\bar{A}_s)_{\mathsf{reg}}$ of *regular streams* of sort $\bar{s}$.

Lemma 3.3.1. *If B is a Σ-subalgebra of A then the regular stream algebra $(\overline{B})_{\mathsf{reg}}$ over B is a $\overline{\Sigma}$-subalgebra of the stream algebras $\overline{B}$, $\bar{A}_{\mathsf{reg}}$, and $\bar{A}$.*

3.4 Dense regular subalgebras

We need the following general topological result.

Lemma 3.4.1. *If X is a topological space and Y a Hausdorff space, and $f\colon X \to Y$ and $g\colon X \to Y$ are both continuous, with $f \restriction D = g \restriction D$ for some dense subset D of X, then $f = g$.*

Let A be a Σ-algebra.

Definition 3.4.2 (Dense subset). A ***Sort***(Σ)-indexed subset D is *dense* in A if for all Σ-sorts s, D_s is dense in A_s.

Lemma 3.4.3. *Let A be a T-standard topological Σ-algebra. Then*

(i) if A is Hausdorff then so is $\bar{A}$;

(ii) if D is a dense Σ-subalgebra of A then $\overline{D}$ and $\overline{D}_{\mathsf{reg}}$ are dense $\overline{\Sigma}$-subalgebras of $\bar{A}$.

Proof: We prove the second part of (ii). Note first that $D_{\mathsf{bool}} = A_{\mathsf{bool}} = \mathbb{B}$ and $D_{\mathsf{nat}} = A_{\mathsf{nat}} = \mathbb{N}$. Now, for any stream sort s, by assumption D_s is dense in A_s. It remains to show that $(\overline{D}_s)_{\mathsf{reg}}$ is dense in $\bar{A}_s = [\mathbb{T} \to A_s]$. Choose any basic open set U in $[\mathbb{T} \to A_s]$, as in (2.1). Since D_s is dense in A_s, we can find $d_i \in U_i \cap D_s$ for $i = 1, \ldots, n$. Now define a stream u by

$$u_i(t) \;=\; \begin{cases} d_i & \text{if } \; t = t_i \text{ for } i = 1, \ldots, n \\ \\ \boldsymbol{\delta}^s & \text{otherwise.} \end{cases}$$

Then $u \in U \cap (\overline{D}_s)_{\mathsf{reg}}$. □

From now on, we will assume that all our topological algebras satisfy

Hausdorff Assumption. *A is a Hausdorff topological algebra.*

3.5 Computable algebras; Computable stream algebras

In order to investigate effective aspects of correctness specification of SCAs (Section 8), we need the concept of a *computable algebra* [BT87].

Definition 3.5.1 (Recursive number algebra). A *recursive number Σ-algebra* Ω is a Σ-algebra in which for each Σ-sort s, Ω_s is a recursive subset of $\mathbb{N}$ and for each Σ-function symbol $F\colon u \to s$,

$$F^{\Omega}\colon \Omega^u \;\to\; \Omega_s$$

is a total recursive function.

Let A be a T-standard Σ-algebra.

Definition 3.5.2 (Effectively presented algebra). An *effective presentation* (α, Ω) for A consists of a recursive number Σ-algebra Ω and a Σ-epimorphism $\alpha\colon \Omega \;\to\; A$.
We assume that $\Omega_{\mathsf{nat}} = \mathbb{N}$ and $\alpha_{\mathsf{nat}} = \boldsymbol{id}_{\mathbb{N}}$.
A is said to be *effectively presented* by (α, Ω).

Next we define the ***Sort***(Σ)***-sorted congruence relation***

$$\equiv_\alpha \;=\; \langle \equiv_{\alpha,s} |\; s \in \boldsymbol{Sort}(\Sigma)\rangle$$

induced by α on Ω:

$$x \equiv_{\alpha,s} y \iff \alpha_s(x) = \alpha_s(y)$$

for all $x, y \in \Omega_s$. Note also that $A \;\cong\; \Omega/{\equiv_\alpha}$.

Definition 3.5.3 (Computable algebra). A is *computable* if it has an effective presentation (α, Ω) in which $\equiv_\alpha$ is decidable on Ω; that is, for each $s \in S$, $\equiv_{\alpha,s}$ is decidable.

Note, next, that the ***stream algebra*** $\bar{A}$ has uncountable carrier sets $\bar{A}_s$ and so it cannot be effectively presented. We therefore work with a regular subalgebra of $\bar{A}$.

Lemma 3.5.4. *Let D be a computable dense Σ-subalgebra of A. Then $\overline{D}_{\mathsf{reg}}$ is a computable dense Σ-subalgebra of $\bar{A}$.*

Proof: It is easy to extend an effective presentation for A with decidable equality to one for $\bar{A}$. The denseness of $\overline{D}_{\mathsf{reg}}$ in $\bar{A}$ follows from Lemma 3.4.3. □

Remark 3.5.5. An example of a computable dense subalgebra of an algebra, satisfying the assumptions of Lemma 3.5.4, is in the real algebra $\mathcal{R}^T$ (Example 2.4.3(b)), in which the rationals $\mathbb{Q}$ form a dense subset of $\mathbb{R}$.

4 Synchronous Concurrent Algorithms

4.1 Introduction to SCAs

An SCA is an algorithm given by a ***network*** N of ***modules***, ***channels***, ***sources*** and ***sinks***. The modules compute and communicate in parallel; computation and data flow between modules is synchronised by a single ***global clock*** measuring discrete time, with values in $\mathbb{T}$.

For simplicity, assume that our T-standard Σ-algebra A contains only one carrier (apart from $\mathbb{B}$ and $\mathbb{T}$), also called A, of sort data. The data flowing between modules are taken from this set.

The SCA processes ***streams*** or infinite sequences $u(0), u(1), u(2), \dots$ of data from A, clocked by $\mathbb{T}$. Such a stream is represented as a function $u\colon \mathbb{T} \to A$. Let $[\mathbb{T} \to A]$ be the set of all streams over A.

The network N is made from a set $M_1, \dots, M_m$ of modules, a set $\mathbf{I}_{\mathsf{in}}$ of p sources and a set $\mathbf{I}_{\mathsf{out}}$ of q sinks. For simplicity we represent the modules, sources and sinks as natural numbers: $I = \{1, \dots, m\}$, $\mathbf{I}_{\mathsf{in}} = \{1, \dots, p\}$ and $\mathbf{I}_{\mathsf{out}} = \{1, \dots, q\}$.

Communication between modules occurs by means of the ***channels***. These have unit bandwidth and are unidirectional; that is, they can transmit only a single datum $a \in A$ at any one time in one direction. Channels may branch with the intention that the datum transmitted along the channel is “copied” and transmitted along each branch. However, channels may not merge.

A ***module*** is an atomic computing device capable of some specific internal processing. If module M_i has $k_i(> 0)$ input channels and one output channel then we assume the processing of M_i to be specified by a total function $\mathsf{F}_i : A^{k_i} \to A$ with the intention that if $a_1, \dots, a_{k_i} \in A$ arrive on the module’s k_i input channels (one datum per channel) at time t then M_i computes $\mathsf{F}_i(a_1, \dots, a_{k_i})$, and transmits it at time $t + 1$.

A ***source*** has no input and one output channel (which may branch). A network with p sources will process p ***input streams*** $u_1, \dots, u_p \in [\mathbb{T} \to A]$, or, equivalently, a vector-valued input stream $\boldsymbol{u} \in [\mathbb{T} \to A]^p$ with $\boldsymbol{u}(t) = (u_1(t), \dots, u_p(t))$.

The ***sinks*** each have one input and no output channel. These will transmit the q output streams.

An SCA’s architecture is given by three ***wiring functions***

$$\begin{aligned} \alpha\colon \mathbf{I} \times \mathbb{N} &\to \mathbf{I}_{\mathsf{in}} \cup \mathbf{I} \\ \beta\colon \mathbf{I} \times \mathbb{N} &\to \{\mathsf{S}, \mathsf{M}\} \\ \mathsf{out}\colon \mathbf{I}_{\mathsf{out}} &\to \mathbf{I}. \end{aligned}$$

The map out is such that for each sink i, $\mathsf{out}(i)$ is the module that supplies i.

The maps α and β are partial functions that enumerate the inputs to a given module in the following way. Given a module $i \in I$ with k_i input channels, for $j = 1, \dots, k_i$ the jth input channel is the output channel of module $\alpha(i, j)$ if $\beta(i, j) = \mathsf{M}$, or the output channel of source $\alpha(i, j)$ if $\beta(i, j) = \mathsf{S}$. If $j \notin \{1, \dots, k_i\}$ then $\alpha(i, j)$ and $\beta(i, j)$ are undefined.

Note that ***feedback*** is characterised by a module i with input j, where $\alpha(i,j) = i$ and $\beta(i,j) = \mathsf{M}$.

The ***initial state*** of the network is specified by a vector $\boldsymbol{a} = (a_1, \ldots, a_m) \in A^m$, where a_i denotes the value output by module i at time $t = 0$.

4.2 Informal Explanation of Operation

Initially, at time $t = 0$, each module i has some initial value $a_i \in A$ on its output channel. Each source j of N is yet to supply its first input datum $u_j(0)$ to the network. Thus, at $t = 0$ there is a single datum on every channel in the network.

Each module i now computes by first reading ite input data and then evaluating F_i on these data. The result of this evaluation is stored on the module's output channel.

Unit Delay Assumption. *For each module in N, the element on its output channel at time $t+1$ is uniquely determined by the data on its input channels at time t.*

So we assume that it takes at most one time cycle for every module to read, evaluate and store in some order, and that any module taking less than one time unit is forced to wait until any slower modules have finished.

Hence, as the clock beats $t = 0, 1, 2, \ldots$, the modules concurrently pass data and compute with each module performing its t-th read/evaluate/store sequence starting at time t and ending by time $t+1$.

4.3 Algebraic Formalisation

We start with a T-standard signature Σ^T and Σ-algebra A^T (§2.3). As stated above, we assume for convenience that there are only three carriers: A of data, $\mathbb{B}$ of booleans and $\mathbb{T}$ of naturals (*i.e.*, discrete time instants). Apart from the standard boolean and arithmetic operations, there may be other functions, including (perhaps) equality on A.

Now we form the ***module algebra*** A^N by adding the module functions to A^T:

$$A^N \;=\; (A^T;\; \mathsf{F}_1, \ldots, \mathsf{F}_m)$$

Next, we extend this to the algebra $\bar{A}^N$ of streams over A^N (§3.1), which we call the ***module stream algebra***:

$$\bar{A}^N \;=\; (A^N,\; [\mathbb{T} \to A];\, \mathsf{eval}).$$

Recall that the ***input*** to N is a stream tuple $\boldsymbol{u} = (u_1, \ldots, u_p) \in [\mathbb{T} \to A]^p$ and the initial values are $\boldsymbol{a} = (a_1, \ldots, a_m) \in A^m$.

Termination Assumption. *At each time $t \in \mathbb{T}$ there is a value output from each module, which can be determined uniquely from t, $\boldsymbol{u}$ and $\boldsymbol{a}$.*

We return to the Unit Delay and Termination Assumptions in Section 4.6.

For each module $i \in I$ we define its ***module value function***

$$\boldsymbol{V}_i \colon \mathbb{T} \times A^m \times [\mathbb{T} \to A]^p \;\to\; A$$

where $\boldsymbol{V}_i(t, \boldsymbol{u}, \boldsymbol{a})$ is the value output by the module i at time t when the network is executed on input $\boldsymbol{u}$ and initial data $\boldsymbol{a}$. Note that these functions are total by Assumption T.

Thus the state of the network N is given by combining the module value functions $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_m$ into the single ***network value function***

$$\boldsymbol{V}^N \colon \mathbb{T} \times A^m \times [\mathbb{T} \to A]^p \;\to\; A^m \tag{4.1a}$$

defined by

$$\boldsymbol{V}^N(t, \boldsymbol{a}, \boldsymbol{x}) \;=\; (\,\boldsymbol{V}_1(t, \boldsymbol{a}, \boldsymbol{x}) \,\ldots,\, \boldsymbol{V}_m(t, \boldsymbol{a}, \boldsymbol{x})). \tag{4.1b}$$

This defines the state of N at each time cycle. (We will sometimes drop the “network superscript” ‘N’.)

The concurrent execution of the modules of N is modelled by the parallel evaluation of $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_m$. We now develop general formulae for the computation of $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_m$ and hence of $\boldsymbol{V}^N$.

4.4 Recursive network equations

We define $\boldsymbol{V}_1(t, \boldsymbol{a}, \boldsymbol{x}), \ldots, \boldsymbol{V}_m(t, \boldsymbol{a}, \boldsymbol{x})$ for $\boldsymbol{a} = (a_1, \ldots, a_m) \in A^m$, $\boldsymbol{x} = (x_1, \ldots, x_p) \in [\mathbb{T} \to A]^p$, and $t = 0, 1, 2, \ldots$, by simultaneous recursion on t.

Base case: Initialisation. For $i = 1, \ldots, m$:

$$\boldsymbol{V}_i(0, \boldsymbol{a}, \boldsymbol{x}) \;=\; a_i \tag{4.2}$$

Recursion step: State transition. Each module i has a functional specification $\mathsf{F}_i \colon A^{k_i} \to A$, where, if $b_1, \ldots, b_{k_i}$ arrive on i’s input channels at time t then the value output by the module at time $t+1$ is $\mathsf{F}_i(b_1, \ldots, b_{k_i})$. Let the SCA have wiring functions α and β as described in §4.1. Then for $i = 1, \ldots, m$ and all $t \geq 0$

$$\boldsymbol{V}_i(t+1, \boldsymbol{a}, \boldsymbol{x}) \;=\; \mathsf{F}_i(b_{i1}, \ldots, b_{ik_i}) \tag{4.3a}$$

where for $j = 1, \ldots, k_i$

$$b_{ij} = \begin{cases} \boldsymbol{V}_{\alpha(i,j)}(t, \boldsymbol{a}, \boldsymbol{x}) & \text{if } \beta(i,j) = \mathsf{M} \\ \\ x_{\alpha(i,j)}(t) & \text{if } \beta(i,j) = \mathsf{S}. \end{cases} \tag{4.3b}$$

Remark 4.4.1. The equations (4.2) and (4.3) together form a definition by ***simultaneous primitive recursion***.

Remark 4.4.2 (Stream transformation). We can rewrite the network value function $\boldsymbol{V}$ (4.1) as a *stream transformation* by "abstraction" or "currying"; *i.e.*, define

$$\widehat{\boldsymbol{V}}\colon A^m \times [\mathbb{T} \to A]^p \;\to\; [\mathbb{T} \to A]^m \tag{4.4a}$$

where

$$\widehat{\boldsymbol{V}}(\boldsymbol{a}, \boldsymbol{x})(t) \;=\; \boldsymbol{V}(t, \boldsymbol{a}, \boldsymbol{x}). \tag{4.4b}$$

We return to a consideration of these two forms, from a computational point of view, in §6.2.

4.5 Output specification

Note that the network value function $\boldsymbol{V}^N$ gives the values output by *every* module in the network. In many cases we are interested only in the values sent to the network's sinks. When the network has $q > 0$ sinks with $\mathbf{I}_{\mathsf{out}} = \{1, \ldots, q\}$ we use the function $\mathsf{out}\colon \mathbf{I}_{\mathsf{out}} \to \mathbf{I}$ (§4.1). Now define the ***network output function***

$$\boldsymbol{V}_{\mathsf{out}}\colon \mathbb{T} \times A^m \times [\mathbb{T} \to A]^p \;\to\; A^q \tag{4.5a}$$

by

$$\boldsymbol{V}_{\mathsf{out}}(t, \boldsymbol{a}, \boldsymbol{x}) \;=\; (\,\boldsymbol{V}_{\mathsf{out}(1)}(t, \boldsymbol{a}, \boldsymbol{x}), \ldots, \,\boldsymbol{V}_{\mathsf{out}(q)}(t, \boldsymbol{a}, \boldsymbol{x})), \tag{4.5b}$$

so that $\boldsymbol{V}_{\mathsf{out}}(t, \boldsymbol{a}, \boldsymbol{x})$ is the vector of q values at the sinks of N at time t.

Note (*cf.* Remark 4.4.2) that we can also reformulate $\boldsymbol{V}_{\mathsf{out}}$ as a stream transformation by abstraction:

$$\widehat{\boldsymbol{V}}_{\mathsf{out}}\colon A^m \times [\mathbb{T} \to A]^p \;\to\; [\mathbb{T} \to A]^q$$

where

$$\widehat{\boldsymbol{V}}_{\mathsf{out}}(\boldsymbol{a}, \boldsymbol{x})(t) = \boldsymbol{V}_{\mathsf{out}}(t, \boldsymbol{a}, \boldsymbol{x}).$$

4.6 Generalisation of the model: Partial algebra of data

There are many fruitful generalisations of our mathematical model. We wish to mention one, where we drop the assumption that the data algebra A is total. This is of great practical importance, in the case, for example, that A is an algebra of reals, that includes the operation of *real division*, and the boolean operations of *equality* and *order*. In order that these operations be *continuous*, we must make them *partial*, as explained in [TZ04].

In such a framework, the module functions will also be partial. as will the network value function. We will also have to work with *partial streams*.

If we maintain our assumption that the channels have unit bandwidth (§4.1), we will have to drop both the Unit Delay and Termination Assumptions (§§4.2,4.3). We will also have to replace our global clock model with a system of *local clocks*. We conjecture that this will be equivalent to the global clock model, with the Termination and Unit Delay Assumptions, in the special case that the algebra A, and the function modules, are total.

Details will be given in a forthcoming publication.

5 Specifications and Correctness

5.1 Indexed sets and operations

Let S be a finite non-empty set. An S-indexed set A is a family $A = \langle A_s \mid s \in S \rangle$.

Given two S-indexed sets $A = \langle A_s \mid s \in S \rangle$ and $B = \langle B_s \mid s \in S \rangle$, an *$S$-indexed mapping* from A to B is a family $f = \langle f_s \mid s \in S \rangle$ where $f_s \colon A_s \to B_s$ for each $s \in S$. In symbols we write $f : A \to B$.

5.2 Syntax: Variables, terms and equations

(a) $\boldsymbol{Var}(\Sigma)$ is a $\boldsymbol{Sort}(\Sigma)$-indexed set, the set of ***Σ-variables*** $(\mathtt{x}, \mathtt{x}_1, \mathtt{x}', \dots)$. $\boldsymbol{Var}_s(\Sigma)$ is the set of ***Σ-variables of sort*** s, denoted $\mathtt{x}^s, \dots$.

(b) $\boldsymbol{T}(\Sigma)$ is the $\boldsymbol{Sort}(\Sigma)$-indexed set of Σ-terms (denoted $t, \dots$), where the set $\boldsymbol{Tm}_s(\Sigma)$ of such terms of sort s (denoted $t : s$) is defined (simultaneously over S) by

$$t^s \ ::= \ \mathtt{x}^s \mid c \mid F(t_1^{s_1}, \dots, t_m^{s_m})$$

where c is a constant symbol of type s, and F is a Σ-function symbol of type $s_1, \dots, s_m \to s$ $(m > 0)$. We also use the notation $b, \dots$ for boolean terms, *i.e.*, terms of sort bool.

Let $\boldsymbol{var}(E)$ be the set of variables contained in any syntactic object E.

(c) $\boldsymbol{Eq}(\Sigma)$ is the set of *Σ-equations* $(t_1^s = t_2^s)$ between Σ-terms of the same Σ-sort. We also write equations as $e, e', \dots$.

5.3 Semantics: Assignments and term evaluations; Satisfaction

Let $\mathcal{A}$ be a Σ-algebra. A (Σ-)***assignment*** on $\mathcal{A}$ is an S-indexed map $\theta \colon \boldsymbol{Var}(\Sigma) \to A$.

Definition 5.3.1 (Term evaluation). Given an assignment $\theta : \boldsymbol{Var}(\Sigma) \to A$, we extend it to an S-indexed map $[\![\cdot]\!]^{A,\theta} \colon \boldsymbol{T}(\Sigma) \to A$, defined by structural induction on $\boldsymbol{T}(\Sigma)$:

(i) If $t \equiv \mathtt{x}$ then $[\![t]\!]^{A,\theta} = \theta(\mathtt{x})$.

(ii) If $t \equiv c$ then $[\![t]\!]^{A,\theta} = c^A$.

(iii) If $t \equiv F(t_1, \dots, t_m)$ then $[\![t]\!]^{A,\theta} = F^A([\![t_1]\!]^{A,\theta}, \dots, [\![t_m]\!]^{A,\theta})$.

Note that for a closed term t, $[\![t]\!]^{A,\theta}$ does not depend on θ. We therefore write it as t^A.

Definition 5.3.2 (Equational specification). A (Σ)-*equational specification* is a pair (Σ, E) where $E \subseteq \boldsymbol{Eq}(\Sigma)$.

Definition 5.3.3 (Equational satisfaction). Let A be a Σ-algebra.

(a) A *satisfies* the Σ-equation $(t_1 = t_2)$, written $A \models t_1 = t_2$, if for all Σ-assignments θ on A, $[\![t_1]\!]^{A,\theta} \ = \ [\![t_2]\!]^{A,\theta}$.

(b) A satisfies the equational specification (Σ, E), written $A \models E$, if $A \models e$ for all $e \in E$.

5.4 Correctness of an SCA

We develop the concept of *specification* of an algebra introduced above, and apply it particularly to stream algebras and SCAs. Hence we introduce the notion of *correctness* of an SCA. We will concentrate on ***relational correctness***.

Suppose that a computational task or behaviour is specified by a relation of the form

$$R \ \subseteq \ \mathbb{T} \times A^m \times [\mathbb{T} \to A]^p \times A^q \tag{5.1}$$

such that for each $t \in \mathbb{T}$, $\boldsymbol{a} \in A^m$, $\boldsymbol{x} \in [\mathbb{T} \to A]^p$, and $\boldsymbol{y} \in A^q$,

$$R(t, \boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y})$$

means that $\boldsymbol{y}$ is acceptable as an output at time t for and initial state $\boldsymbol{a}$ and input stream $\boldsymbol{x}$. We call R the ***specification relation***.

There are various ways of formulating correctness w.r.t. a specification relation R, depending on how we treat ***initialisations*** and ***inputs***: We can consider a ***particular initialisation***, or ***all initialisations from some subset of*** A^m (possibly all of A^m). Similarly, we can consider a ***particular input stream***, or ***all inputs from some subset of*** $[\mathbb{T} \to A]^p$ (possibly all of $[\mathbb{T} \to A]^p$). To take a typical (and useful) case:

Definition 5.4.1 (Correctness for initialisations and inputs from some set). For any sets $P \subseteq A^m$ of initialisations and $Q \subseteq [\mathbb{T} \to A]^p$ of inputs, the SCA is *correct w.r.t. P, Q and R* if

$$(\forall t \in \mathbb{T})\,(\forall \boldsymbol{a} \in P)\,(\forall \boldsymbol{x} \in Q)\ R(t, \boldsymbol{a}, \boldsymbol{x}, \boldsymbol{V}_{\mathsf{out}}(t, \boldsymbol{a}, \boldsymbol{x})). \tag{5.2}$$

Here the output value function $\boldsymbol{V}_{\mathsf{out}} \colon \mathbb{T} \times A^m \times [\mathbb{T} \to A]^p \ \to \ A^q$ (3.9) is a ***selection function*** for the relation R, relative to P and Q.

We will investigate the computational significance of such correctness assertions in the next section.

6 Primitive recursive computability on stream algebras

6.1 Primitive recursion on abstract algebras

In [TZ88] we developed a theory of ***abstract computability on standard abstract many-sorted algebras***. We formulated a ***generalised Church-Turing thesis***, which identifies a certain class of functions (namely, 'μPR' computable or '***While***' computable) with functions algorithmically computable on such structures.

We also developed a theory of ***generalised primitive recursion*** over T-standard algebras A. These generalise Kleene's primitive recursion functions on $\mathbb{N}$ [Kle52]. and form a proper subclass of the class μPR.

Briefly, we define a class PR(A) of PR (primitive recursive) functions on A, generated by schemes for (i) the initial functions and constants, *i.e.*, the interpretations on A of the Σ-functions, (ii) projections, (iii) definition by cases, (iv) composition, and (v) simultaneous primitive recursion.

Note that the class $\mu\mathrm{PR}(A)$ is formed from $\mathrm{PR}(A)$ by adding a scheme for the (constructive) least number operator.

Lemma 6.1.1 (PR computability and continuity). *Let A be a topological algebra. Then all functions in* $\mathrm{PR}(A)$ *are continuous.*

This is proved, in fact for all $\mu\mathrm{PR}$ functions, in [TZ88].

We now consider a class of relations on algebras broader than primitive recursiveness.

Definition 6.1.2 (Equationally PR definable relations). A relation $R \subseteq A^u$ on an algebra A is *equationally* PR *definable* on A ($\mathrm{EqPR}(A)$) if there are $\mathrm{PR}(A)$ functions $\mathsf{f}_R, \mathsf{g}_R \colon u \to s$ for some Σ-sorts u, s such that for all $\boldsymbol{a} \in A^u$

$$\boldsymbol{a} \in R \quad \Longleftrightarrow \quad \mathsf{f}_R(\boldsymbol{a}) = \mathsf{g}_R(\boldsymbol{a}). \tag{6.1}$$

We call the rhs of (6.1) a *PR defining equation* for R, and the pair $(\mathsf{f}_R, \mathsf{g}_R)$ PR *defining functions* for R.

Remark 6.1.3 (Comparison of PR and EqPR computability). Note that $\mathrm{EqPR}(A)$ is a broader concept than $\mathrm{PR}(A)$. For on the one hand, any $\mathrm{PR}(A)$ relation R is also $\mathrm{EqPR}(A)$, since (if χ_R is the characteristic function of R)

$$a \in R \quad \Longleftrightarrow \quad \chi_R(a) = \mathsf{true}$$

(a special case of (6.1)). But on the other hand, the range sort s (in Definition 6.1.2) need not be an equality sort (*cf.* §2.2), *i.e.*, equality at sort s is not necessarily PR.

6.2 Primitive recursion on stream algebras

Assume for simplicity (as stated in Section 4) that our T-standard Σ-algebra A contains (apart from $\mathbb{B}$ and $\mathbb{T}$) only one carrier A of data.

Consider now PR stream valued functions or ***stream transformers*** on $\bar{A}$:

$$f \colon [\mathbb{T} \to A]^m \times A^n \ \to \ [\mathbb{T} \to A]. \tag{6.2}$$

It has been shown [TZ94] that all PR stream transformers f of type as in (6.2) have the form

$$f(u_1, \ldots, u_m, a_1, \ldots, a_n) \ = \ u_{f_0(u_1,\ldots,u_m,a_1,\ldots,a_n)}$$

for some PR function

$$f_0 \colon [\mathbb{T} \to A]^m \times A^n \ \to \ \mathbb{T}.$$

In other words, PR stream transformers are not "interesting"; they only return one of the input streams (the choice of which one depending primitive recursively on the inputs).

We therefore consider a broader, more interesting class of stream transformers, namely the class $\lambda\mathrm{PR}(\bar{A})$ formed from $\mathrm{PR}(\bar{A})$ by adding a scheme for *stream (λ)-abstraction*. Note that a function f as in (6.2) will be in $\lambda\mathrm{PR}(\bar{A})$ if its "cartesian" or "uncurried" form

$$\check{f} \colon [\mathbb{T} \to A]^m \times A^n \ \times \mathbb{T} \ \to \ A$$

is in $\mathrm{PR}(\bar{A})$, where

$$\check{f}(\boldsymbol{u}, \boldsymbol{a}, t) = f(\boldsymbol{u}, \boldsymbol{a})(t).$$

Note also that we can define the class $\mathrm{Eq}\lambda\mathrm{PR}(A)$ of *equational* λPR *definable* relations on A, analogously to $\mathrm{EqPR}(A)$ (Definition 6.1.2).

Now assume A, and hence $\bar{A}$, are topological algebras.

Lemma 6.2.1. *For f as in (6.2), f is continuous iff $\check{f}$ is continuous.*

Hence, from Lemma 6.1.1:

Lemma 6.2.2. *All functions in $\lambda\mathrm{PR}(\bar{A})$ are continuous.*

Corollary 6.2.3. *Let A be Hausdorff T-standard Σ-algebra, and D a dense subalgebra of A. Let f and g be* λPR *functions on $\bar{A}$. Then the following are equivalent:*

(i) $f = g$ *on* $\bar{A}$

(ii) $f = g$ *on* $\overline{D}$

(iii) $f = g$ *on* $\bar{A}_{\mathsf{reg}}$

(iv) $f = g$ *on* $\overline{D}_{\mathsf{reg}}$.

Proof: From Lemmas 3.4.1, 3.4.3 and 6.2.2. □

6.3 Primitive recursiveness of SCA state function

Recall the module, network and output value functions (§§4.3, 4.5).

Theorem 1. *For any SCA over A with network N and module functions $\mathsf{F}_1, \ldots, \mathsf{F}_m$:*

(a) The module value functions $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_m$, network value function $\boldsymbol{V}^N$ and network value function $\boldsymbol{V}_{\mathsf{out}}$ are in $\mathrm{PR}(\bar{A})$.

(b) The abstracted forms $\widehat{\boldsymbol{V}}$ and $\widehat{\boldsymbol{V}}_{\mathsf{out}}$ are in $\lambda\mathrm{PR}(\bar{A})$.

Proof: The main step in (a) is to show that $\boldsymbol{V}^N$ is definable (uniquely) by simultaneous primitive recursion (equations 4.8) from the module functions $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_m$. This can be seen by a simple inductive argument, parallelling the PR definition. □

Remark 6.3.1 (Fixed point theory for network functions). In [TZ08] the question of the existence of a solution to network equations is investigated in a more general context, which includes, as special cases, (i) a ***discrete global clock***, characteristic of SCAs, and (ii) a ***continuous global clock***, characteristic of analog systems [TZ07]. This involves a *fixed point argument* as follows. Define, for all $\boldsymbol{a} \in A^m$ and $\boldsymbol{x} \in [\mathbb{T} \to A]^p$, the stream transformation

$$\Phi_{\boldsymbol{a},\boldsymbol{x}} : \ [\mathbb{T} \to A]^m \ \to \ [\mathbb{T} \to A]^m$$

by $\Phi_{\boldsymbol{a},\boldsymbol{x}}(\boldsymbol{u}) \ = \ \boldsymbol{v}$, where (*cf.* equations (4.2) and (4.3)) for $i = 1, \ldots, m$

$$v_i(0) \ = \ a_i$$

and for all $t \geq 0$

$$v_i(t+1) \;=\; \mathsf{F}_i(b_{i1}, \ldots, b_{ik_i})$$

where for $j = 1, \ldots, k_i$

$$b_{ij} = \begin{cases} u_{\alpha(i,j)}(t) & \text{if } \beta(i,j) = \mathsf{M} \\ \\ x_{\alpha(i,j)}(t) & \text{if } \beta(i,j) = \mathsf{S}. \end{cases}$$

Then $\Phi_{\boldsymbol{a},\boldsymbol{x}}$ is *contracting*, in the sense that for all $t \in \mathbb{T}$ and all $\boldsymbol{u}_1, \boldsymbol{u}_2 \in [\mathbb{T} \to A]^m$:

$$\boldsymbol{u}_1 \restriction_t \;=\; \boldsymbol{u}_2 \restriction_t \quad\Longrightarrow\quad \Phi_{\boldsymbol{a},\boldsymbol{x}}(\boldsymbol{u}_1) \restriction_{t+1} \;=\; \Phi_{\boldsymbol{a},\boldsymbol{x}}(\boldsymbol{u}_2) \restriction_{t+1}$$

(where '$\boldsymbol{u} \restriction_t$' means the restriction of $\boldsymbol{u}$ to the initial segment $\{0, 1, \ldots, t\}$ of $\mathbb{T}$.) From this we can derive the existence of a unique fixed point $\boldsymbol{u}$ of $\Phi_{\boldsymbol{a},\boldsymbol{x}}$, satisfying $\boldsymbol{u} = \widehat{\boldsymbol{V}}(\boldsymbol{a}, \boldsymbol{x})$.

6.4 Computability of relational correctness specification

Recall the definition (5.4.1) of correctness for a specification relation R with initialisations and input streams from sets $P \subseteq A^m$ and $Q \subseteq [\mathbb{T} \to A]^p$ respectively (repeating (5.2)):

$$(\forall t \in \mathbb{T})\,(\forall \boldsymbol{a} \in P)\,(\forall \boldsymbol{x} \in Q)\;\; R(t, \boldsymbol{a}, \boldsymbol{x}, \boldsymbol{V}_{\mathsf{out}}(t, \boldsymbol{a}, \boldsymbol{x})). \tag{6.3}$$

Theorem 2 (Computability of correctness specification). *Suppose*

(a) *P, Q and R are* EqλPR *on $\bar{A}$,*

(b) *A has a dense computable subalgebra D.*

Then we can effectively construct a computable algebra $C_{V,P,Q,R}$ with signature $\Sigma_{V,P,Q,R}$ that expands $\overline{D}_{\mathsf{reg}}$ by functions, and equations e_P, $e_Q, e_{V,R}$ over $\Sigma_{V,P,Q,R}$ such that the following are equivalent:

(i) *$\boldsymbol{V}$ is correct w.r.t. P, Q and R, i.e.,* (6.3) *holds;*

(ii) $C_{V,P,Q,R} \;\models\; e_P \wedge e_Q \;\to\; e_{V,R}$.

In consequence, correctness in the sense of (i) *can be effectively reduced to the validity of conditional equations in a computable algebra and is co-recursively enumerable.*

Proof: We prove (i)$\Rightarrow$(ii). Consider the statement

$$\boldsymbol{a} \in P \wedge \boldsymbol{x} \in Q \quad\longrightarrow\quad R(t, \boldsymbol{a}, \boldsymbol{x}, \boldsymbol{V}_{\mathsf{out}}(t, \boldsymbol{a}, \boldsymbol{x})). \tag{6.4}$$

Let $(\mathsf{f}_P, \mathsf{g}_P)$, $(\mathsf{f}_Q, \mathsf{g}_Q)$ and $(\mathsf{f}_R, \mathsf{g}_R)$ be λPR defining functions for the sets P, Q and R respectively. By assumption and Theorem 1, these functions, as well as $\boldsymbol{V}$, are all λPR on $\overline{D}_{\mathsf{reg}}$. By assumption ($i$), (6.4) holds on A, and therefore, by Corollary 6.2.3, on $\overline{D}_{\mathsf{reg}}$. Since D is a computable algebra, so is $\overline{D}_{\mathsf{reg}}$, by Lemma 3.5.4, with effective presentation (α, Ω) say (recall §3.5). Now expand $\overline{D}_{\mathsf{reg}}$ to the algebra

$$C_{V,P,Q,R} \;=_{df}\; (\overline{D}_{\mathsf{reg}};\; \boldsymbol{V},\, \mathsf{f}_P,\, \mathsf{g}_P,\, \mathsf{f}_Q,\, \mathsf{g}_Q,\, \mathsf{f}_R,\, \mathsf{g}_R) \tag{6.5}$$

with signature $\Sigma_{V,P,Q,R}$. Since the seven functions shown in (6.5) are all λPR over $\overline{D}_{\mathsf{reg}}$, they are "$\alpha$-computable" on $\overline{D}_{\mathsf{reg}}$. (This follows from the soundness theorem for abstract computability [TZ04]). Hence $C_{V,P,Q,R}$ is also a computable algebra. Moreover (6.4) has the form of a conditional equation $e_P \wedge e_Q \;\rightarrow\; e_{V,R}$ over $C_{V,P,Q,R}$. Hence (*ii*) follows.

That the correctness problem is co-r.e. follows from the α-computability of the functions noted above, together with the decidability of $\equiv_\alpha$. □

Example 6.4.1. Let A be the T-standard topological agebra $\mathcal{R}^T$ (Example 2.4.3(*b*)). A has a dense computable subalgebra $D = \mathcal{Q}^T$ consisting of the rationals $\mathbb{Q}$ with the same signature as A. As a (very simple) example of an equationally λPR (in fact, PR) specification relation, we could take

$$R(t, a, x_1, x_2, y) \quad\Longleftrightarrow\quad x_1(t)^2 + x_2(t)^2 = y^2.$$

A more interesting example would be something like

$$R'(t, a, x_1, x_2, y) \quad\Longleftrightarrow\quad (0 < x_1(t)) \wedge (0 < x_2(t)) \wedge (x_1(t)^2 + x_2(t)^2 < y^2),$$

i.e., a boolean combination of equalities and inequalities between λPR terms.

The problem here is that equality and order, as *total operations* on $\mathbb{R}$, are not computable [TZ99, TZ04, TZ05]. In this paper we have solved this problem for equality by using the *computable subalgebra* $\mathcal{Q}^T$ of $\mathcal{R}^T$, together with the concept of *equational PR definability* (Definition 6.1.2).

To handle '<', however, would seem to require a major extension of the theory, so as to incorporate *partial algebras*. This is planned for a future publication.

We could ask if condition (*ii*) in Theorem 2 could be replaced by a statement that the conditional equation is a *valid consequence* of a certain set of axioms, *i.e.*, a completeness result. However the correctness problem for conditional equations in stream algebras is complete Π^0_1 [BT87] and so completeness fails. In this direction, however, we can prove the following, using results of Bergstra and Tucker on initial algebra semantics [BT80, BT82, BT87, BT92].

Theorem 3 (Validity of correctness specification). *Given the hypotheses of Theorem 2, we can effectively construct a finite conditional equational specification* ($\Sigma_{V,P,Q,R}$), $E_{V,P,Q,R}$) *and equations* e_P, $e_Q, e_{V,R}$ *over* $\Sigma_{V,P,Q,R}$ *such that the following are equivalent:*

(*i*) $\boldsymbol{V}$ *is correct w.r.t.* P, Q *and* R, *i.e.*, (6.3) *holds;*

(*ii*) $T(\Sigma_{V,P,Q,R},\, E_{V,P,Q,R}) \models\; e_P \wedge e_Q \;\rightarrow\; e_{V,R}$,

where $T(\Sigma_{V,P,Q,R},\, E_{V,P,Q,R})$ *is the* $\Sigma_{V,P,Q,R}$*-term model generated by* $E_{V,P,Q,R}$.

In particular ($\Sigma_{V,P,Q,R}$, $E_{V,P,Q,R}$) *can be chosen to be either (a) a complete orthogonal term rewriting system; or (b) a small specification: if* A *has* $n+2$ *sorts then* $\Sigma_{V,P,Q,R}$ *has* $6(n+2)$ *hidden operations and* $E_{V,P,Q,R}$ *has* $4(n+2)$ *equations.*

7 Concluding remarks

Since the idea of a SCA is general, our methods and results apply to the various classes of algorithms, architectures and physical models mentioned in the Introduction, as well as to several others.

Consider abstract hardware, for example. Our mathematical notions can be used to clarify and improve methods developed in case studies and software tools for design, simulation and verification work on specific classes of SCAs; and to make comparisons and help transfer specific methods between different classes. A fundamental problem in computing is that of hierarchy and the relationship between levels of abstraction in specifications and programs. A theory of SCAs could be used as a starting point for a very general theory of hardware.

As another example, consider physical modelling. Our notions can be used to clarify and improve methods developed in case studies and software tools for modelling, simulation and analysis work on specific classes of SCAs; and to make comparisons and help transfer specific methods between different classes. Again, hierarchy and the relationship between levels of abstractions in physical models is a fundamental problem. As one example of this, in [PTH02] we discuss how ideas from hardware design can be applied to non-linear dynamical systems in the case of whole-heart modelling.

There are, however, many more mathematical questions to answer. For example, we want to investigate the theory of SCAs based on partial data algebras, with partial streams.

Acknowledgments. We thank the following colleagues for many useful and stimulating discussions on the subject: J.A. Bergstra, B.R.J. McConnell, M.J. Poole, R. Stephens, W.B. Yates, S.M. Eker, K. Hobley, A.R. Martin, and A.V. Holden. The research of the second and third authors was supported in part by a grant from EPSRC (Engineering and Physical Sciences Research Council, UK). The research of the third author was supported in part by a grant from NSERC (Natural Sciences and Engineering Research Council, Canada).

References

References

[BT80] J.A. Bergstra and J.V. Tucker. A characterisation of computable data types by means of a finite equational specification method. In J.W. de Bakker and J. van Leeuwen, editors, *7th International Colloquium on Automata, Languages and Programming, Noordwijkerhout, The Netherlands, July 1980*, volume 85 of *Lecture Notes in Computer Science*, pages 76–90. Springer-Verlag, 1980.

[BT82] J.A. Bergstra and J.V. Tucker. The completeness of the algebraic specification methods for data types. *Information and Control*, 54:186–200, 1982.

[BT87] J.A. Bergstra and J.V. Tucker. Algebraic specifications of computable and semicomputable data types. *Theoretical Computer Science*, 50:137–181, 1987.

[BT92] J.A. Bergstra and J.V. Tucker. Equational specifications, complete term rewriting systems and computable and semicomputable algebras. Technical Report CS-20-92, Department of Computer Science, Swansea University, Swansea, Wales, 1992.

[Har89] N.A. Harman. *Formal specifications for digital systems.* PhD Thesis, School of Computer Studies, University of Leeds, 1989.

[HPTZ92] A.V. Holden, M. Poole, J.V. Tucker, and H. Zhang. Coupled map lattices as computational systems. *Chaos*, 2:367–376, 1992.

[HT88a] N.A. Harman and J.V. Tucker. Clocks, retimings, and the formal specification of a uart. In G. Milne, editor, *The Fusion of Hardware Design and Verification (Proceedings of IFIP Working Group 10.2 Working Conference)*, pages 375–396. North Holland, 1988.

[HT88b] N.A. Harman and J.V. Tucker. Formal specifications and the design of verifiable computers. In *Proceedings of 1988 UK IT Conference, held under the auspices of the Information Engineering Directorate of the Department of Trade and Industry*, pages 500–503. Institute of Electrical Engineers, 1988.

[HT90] N.A. Harman and J.V. Tucker. The formal specification of a digital correlator i: user specification process. In K. McEvoy and J.V. Tucker, editors, *Theoretical Foundations of VLSI Design*, pages 161–262. Cambridge University Press, 1990.

[HT91] N.A. Harman and J.V. Tucker. Consistent refinements of specifications for digital systems. In P. Prinetto, editor, *Correct hardware design methodologies (Proceedings ESPRIT BRA 3216 Workshop)*, pages 281–304. Elsevier, 1991.

[HT93] N.A. Harman and J.V. Tucker. Algebraic methods and the correctness of microprocessors. In G.J. Milne and L. Pierre, editors, *Correct Hardware Design and Verification Methods*, volume 683 of *Lecture Notes in Computer Science*, pages 92–108. Springer-Verlag, 1993.

[HT96] N.A. Harman and J.V. Tucker. Algebraic models of microprocessors: architecture and organisation. *Acta Informatica*, 33:421–456, 1996.

[HTT90] A.V. Holden, B.C. Thompson, and J.V. Tucker. The computational structure of neural systems. In A.V. Holden and V.I. Kryukov, editors, *Neurocomputers and Attention I: Neurobiology, Synchronisation and Chaos*, pages 223–240. Manchester University Press, 1990.

[HTT91] A.V. Holden, B.C. Thompson, and J.V. Tucker. Can excitable media be considered as computational systems? *Physica D*, 49:240–246, 1991.

[Kle52] S.C. Kleene. *Introduction to Metamathematics.* North Holland, 1952.

[PER89] M.B. Pour-El and J.I. Richards. *Computability in Analysis and Physics.* Springer-Verlag, 1989.

[PTH98] M.J. Poole, J.V. Tucker, and A.V. Holden. Hierarchies of spatially extended systems and synchronous concurrent algorithms. In B. Möller and J.V. Tucker, editors, *Prospects for hardware foundations*, volume 1546 of *Lecture Notes in Computer Science*, pages 184–235. Springer-Verlag, 1998.

[PTH02] M.J. Poole, J.V. Tucker, and A.V. Holden. Hierarchical reconstructions of cardiac tissue. *Chaos, Solitons and Fractals*, 13:1581–1612, 2002.

[TT91] B.C. Thompson and J.V. Tucker. Algebraic specification of synchronous concurrent algorithms and architectures (Revised). Research Report 9-91, Department of Computer Science, Swansea University, Swansea, Wales, 1991.

[TZ88] J.V. Tucker and J.I. Zucker. *Program Correctness over Abstract Data Types, with Error-State Semantics*, volume 6 of *CWI Monographs.* North Holland, 1988.

[TZ94] J.V. Tucker and J.I. Zucker. Computable functions on stream algebras. In H. Schwichtenberg, editor, *Proof and Computation: NATO Advanced Study Institute International Summer School at Marktoberdorf, 1993*, pages 341–382. Springer-Verlag, 1994.

[TZ99] J.V. Tucker and J.I. Zucker. Computation by 'while' programs on topological partial algebras. *Theoretical Computer Science*, 219:379–420, 1999.

[TZ00] J.V. Tucker and J.I. Zucker. Computable functions and semicomputable sets on many-sorted algebras. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 5, pages 317–523. Oxford University Press, 2000.

[TZ02a] J.V. Tucker and J.I. Zucker. Abstract computability and algebraic specification. *ACM Transactions on Computational Logic*, 3:279–333, 2002.

[TZ02b] J.V. Tucker and J.I. Zucker. Infinitary initial algebra specifications for stream algebras. In W. Sieg, R. Sommer, and C. Talcott, editors, *Reflections on the Foundations of Mathematics: Essays in honor of Solomon Feferman*, volume 15 of *Lecture Notes in Logic*, pages 234–256. Association for Symbolic Logic, 2002.

[TZ04] J.V. Tucker and J.I. Zucker. Abstract versus concrete computation on metric partial algebras. *ACM Transactions on Computational Logic*, 5:611–668, 2004.

[TZ05] J.V. Tucker and J.I. Zucker. Computable total functions, algebraic specifications and dynamical systems. *Journal of Logic and Algebraic Programming*, 62:71–108, 2005.

[TZ07] J.V. Tucker and J.I. Zucker. Computability of analog networks. *Theoretical Computer Science*, 371:115–146, 2007.

[TZ08] J.V. Tucker and J.I. Zucker. Computation on algebras of continuous functions. In preparation, 2008.

[Wei00] K. Weihrauch. *Computable Analysis: An Introduction.* Springer-Verlag, 2000.

Uncomputability[☆] and Undecidability in Economic Theory

K. Vela Velupillai[a,*]

[a] *Department of Economics, University of Trento, Via Inama 5, 381 00 Trento, Italy*

Abstract

Economic theory, game theory and mathematical statistics have all increasingly become *algorithmic sciences*. *Computable Economics*, *Algorithmic Game Theory* ([28]) and *Algorithmic Statistics* ([13]) are frontier research subjects. All of them, each in its own way, are underpinned by (classical) recursion theory – and its applied branches, say computational complexity theory or algorithmic information theory – and, occasionally, proof theory. These research paradigms have posed new mathematical and metamathematical questions and, inadvertently, undermined the traditional mathematical foundations of economic theory. A concise, but partial, pathway into these new frontiers is the subject matter of this paper. Interpreting the core of *mathematical economic theory* to be defined by *General Equilibrium Theory* and *Game Theory*, a general – but concise – analysis of the *computable* and *decidable* content of the implications of these two areas are discussed. Issues at the frontiers of *macroeconomics*, now dominated by *Recursive Macroeconomic Theory*[1], are also tackled, albeit ultra briefly.

[☆] By 'uncomputability' I mean both that arising from (classical) recursion theoretic considerations, and from those due to formal non-constructivities (in any sense of constructive mathematics).

[*] Quite serendipitously, I am in the happy position of being able to pay long overdue acknowledgements to four of my "fellow-invitees" at this meeting: Ann Condon, Barry Cooper, Chico Doria and Karl Svozil - although they are, almost certainly, unaware of the kind of ways in which I have benefitted from their wisdom and scholarship, over the years (cf. in particular, [7], [8] and [46], respectively). Indeed, in the case of Barry Cooper, I am also deeply indebted to his own distinguished teacher, R.L. Goodstein, whose works have had a lasting influence in the way I think about the kind of mathematics that is suitable for mathematizing economics. In particular, it was from his outstanding calculus text ([14]) that I learned the felicitous phrase 'undecidable disjunction', which was instrumental in my understanding of the pernicious influence of the Bolzano-Weierstrass theorem in algorithmic mathematics and, *a fortiori*, in algorithmic economics.But, as always these days, it is to Chico Doria and Stefano Zambelli that I owe most - without the slightest implications for the remaining infelicities in the paper.

Email addresses: kvelupillai@gmail.com (K. Vela Velupillai)

[1] The qualification 'recursive' here has nothing to do with 'recursion theory'. Instead, this is a reference to the mathematical formalizations of the rational economic agent's intertemporal optimization problems, in terms of Markov Decision Processes, (Kalman) Filtering and Dynamic Programming, where a kind of 'recursion' is invoked in the solution methods. The metaphor of the rational economic agent as a 'signal processor' underpins the recursive macroe-

The point of view adopted is that of *classical recursion theory* and varieties of *constructive mathematics.*

Key words: General Equilibrium Theory, Game Theory, Recursive Macroeconomics, (Un)computability, (Un)decidability, Constructivity

1. A Mathematical and Metamathematical Preamble[2]

Distinguished pure mathematicians, applied mathematicians, philosophers and physicists have, with the innocence of integrity and the objectivity of their respective disciplines, observing the mathematical practice and analytical assumptions of economists, have emulated the 'little child' in Hans Christian Andersen's evocative tale to exclaim similar obvious verities, from the point of view of *algorithmic mathematics.* I have in mind the 'innocent', but obviously potent, observations made by Michael Rabin ([35]), Hilary Putnam ([34]) Maury Osborne ([29]), Jacob Schwartz ([41]), Steve Smale ([42]), Glenn Shafer & Vladimir Vovk ([39]) and David Ruelle ([36])[3], each tackling an important core issue in mathematical economics and finding it less than adequate from a serious mathematical and computable point of view – in addition to being contrived, even from the point of view of common sense economics[4]. Decidability in games, uncomputability in rational choice, inappropriateness of real analysis in the modelling of financial market dynamics, the gratuitous assumption of (topological) fix point formalizations in equilibrium economic theory, the question of the algorithmic solvability of supply-demand (diophantine) equation systems, finance theory without probability (but with an algorithmically underpinned theory of *martingagles*), are some of the issues these 'innocent' pioneers raised, against the naked economic theoretic emperor.

I hasten to add that there were pioneers even within the 'citadel' of economic theory. Their contributions have been discussed and documented in various of my writings over the past 20 years or so and, therefore, I shall not rehash that

conomic paradigm.

[2] I am deeply indebted to two anonymous referees for clarifying many obscure issues in an earlier version of this paper. They also contributed with deep and penetrating questions and suggestions that contributed considerably to improving the paper. Alas, they are not responsible for the remaining obscurities.

[3] In addition to the themes Ruelle broached in this '*Gibbs Lecture*', the first four, chapter 9 and the last four chapters of his elegant new book ([37]) are also relevant for the philosophical underpinnings of this paper. Although the two Ruelle references are not *directly* related to the subject matter of this paper, I include them because the mathematical themes of these works are deeply relevant to my approach here.

[4] Discerning scholars would notice that I have not included the absolutely pioneering work of Louis Bachelier in this list (cf. [9] or [12] for easily accessible English versions of Bachelier's Théorie de la Spéculation). This is only because he did not raise issues of computability, decidability and constructivity, that he could not possibly have done at the time he wrote, even though Hilbert's famous '*Paris Lecture*' was only five months away from when Bachelier defended his remarkable doctoral dissertation – also in Paris.

part of the story here[5]. Suffice it to mention just the more obvious pioneers who emerged from within the 'citadel': Peter Albin, Kenneth Arrow, Douglas Bridges[6], Alain Lewis, Herbert Scarf and Herbert Simon. Albin, Arrow, Lewis, Scarf and Simon considered seriously, to a greater and lesser extent, the issue of modelling economic behaviour, both in the case of individually rational and in cases of strategically rational interactions, the place of formal computability considerations and their implications. Bridges and Scarf were early contributors to what may be called '*constructive economics*', complementing the '*computable economics*' of the former contributors. Scarf, of course, straddled both divides, without – surprisingly – providing a unifying underpinning in what I have come to call '*algorithmic economics*'[7].

Economic theory, at every level and at almost all frontiers - be it microeconomics or macroeconomics, game theory or IO - is now almost irreversibly dominated by *computational*, *numerical*[8] and *experimental* considerations. Curiously, though, none of the frontier emphasis from any one of these three points of view - computational, numerical or experimental - is underpinned by the natural algorithmic mathematics of either computability theory or constructive analysis[9]. In particular, the much vaunted field of Computable General Equilibrium theory, with explicit claims that it is based on constructive and computable foundations is neither the one, nor the other[10]. Similarly, Newclassical Economics, the dominant strand in Macroeconomics, has as its formal core so-called *Recursive* Macroeconomic Theory. The dominance of computational and numerical analysis, powerfully underpinned by serious approximation theory, is totally devoid of computable or constructive foundations.

[5] The absence of any detailed discussion of honest priorities from within the 'citadel' in this paper is also for reasons of space constraints.

[6] Douglas Bridges is, of course, a distinguished mathematician who has made fundamental contributions - both at the research frontiers and at the level of cultured pedagogy - to constructive analysis, computability theory and their interdependence, too. However, I consider his contributions to 'constructive economics' to be at least as pioneering as Alain Lewis's to 'computable economics'. Alas, neither the one nor the other seems to have made the slightest difference to the orthodox, routine, practice of the mathematical economist.

[7] In this paper I shall not discuss the place of computational complexity theory in economics, which has an almost equally distinguished ancestry. I provide a fairly full discussion of the role of computational complexity theory, from the point of view of algorithmic economics in [54]

[8] By this I aim to refer to *classical numerical analysis*, which has only in recent years shown tendencies of merging with computability theory - for example through the work of Steve Smale and his many collaborators (cf. for example [2]). To the best of my knowledge the foundational work in computable analysis and constructive analysis was never properly integrated with classical numerical analysis.

[9] With the notable exception of the writings of the above mentioned pioneers, none of whom work - or worked - in any of these three areas, as conceived and understood these days. For excellent expositions of numerical and computational methods in economics, particularly macroeconomics, see [4], [18] and [23].

[10] A complete and detailed analysis of the false claims – from the point of view of computability and constructivity – of the proponents and practitioners of CGE modelling is given in my recent paper devoted explicitly to the topic (cf. [52]).

The reasons for this paradoxical lack of interest in computability or constructivity considerations, even while almost the whole of economic theory is almost completely dominated by numerical, computational and experimental considerations, are quite easy to discern: the reliance of every kind of mathematical economics on real analysis for formalization. I shall not go into too many details of this 'conjecture' in this paper, but once again the interested reader is referred to [51] and [53] for more comprehensive discussions and formal analysis (but see also § *3*, below).

Against this 'potted' background of pioneering innocence and core issues, the rest of this paper is structured as follows. Some of the key results on uncomputability and undecidability, mostly derived by this author are summarised in a fairly merciless telegraphic form (with adequate and detailed references to sources) in the next section. In section 3 some remarks on the mathematical underpinnings of these 'negative' results are discussed and, again, stated in the usual telegraphic form. The concluding section suggests a framework for invoking my 'version' of unconventional computation models for mathematical models of the economy.

Two distinguished pioneers of economic theory and, appropriately, national income accounting, Kenneth Arrow and Richard Stone (in collaboration with Alan Brown) – who also happened to be *Nobel Laureates* – almost delineated the subject matter of what I have come to call *Computable Economics*. The former conjectured, more than two decades ago, as a frontier research strategy for the mathematical economic theorist, that:

> "The next step in analysis, I would conjecture, is a more consistent assumption of computability in the formulation of economic hypothesis. This is likely to have its own difficulties because, of course, not everything is computable, and there will be in this sense an inherently unpredictable element in rational behavior." [1]

Richard Stone (together with Alan Brown), speaking as an applied economist, grappling with the conundrums of adapting an economic theory formulated in terms of a mathematics alien to the digital computer and to the nature of the data[11], confessed his own *credo* in characteristically perceptive

[11] Maury Osborne, with the clarity that can only come from a rank outsider to the internal paradoxes of the dissonance between economic theory and applied economics, noted pungently:

> "There are numerous other paradoxical beliefs of this society [of economists], consequent to the difference between discrete numbers. . . in which data is recorded, whereas the theoreticians of this society tend to think in terms of real numbers. ...No matter how hard I looked, I never could see any actual real [economic] data that showed that [these solid, smooth, lines of economic theory] ... actually could be observed in nature. . . . At this point a beady eyed Chicken Little might . . . say, 'Look here, you can't have solid lines on that picture because there is always a smallest unit of money . . . and in addition there is always a unit of something that you buy. . . . [I]n any event we should have just whole numbers of some sort on [the supply-demand] diagram on both axes. The lines should be

terms [12]

> “Our approach is quantitative because economic life is largely concerned with quantities. We use [digital] computers because they are the best means that exist for answering the questions we ask. It is our responsibility to formulate the questions and get together the data which the computer needs to answer them.” [3], p.viii

Economic analysis, as practised by the mathematical economist – whether as a microeconomist or a macroeconomist, or even as a game theorist or an IO theorist – continues, with princely unconcern for these conjectures and conundrums, to be mired in, and underpinned by, conventional real analysis. Therefore, it is a ‘cheap’ exercise to extract, discover and display varieties of uncomputabilities, undecidabilities and non-constructivities in the citadel of economic theory. Anyone with a modicum of expertise in recursion theory, constructive analysis or even nonstandard analysis in its constructive modes, would find, in any reading from these more algorithmically oriented perspectives, the citadel of economic theory, game theory and IO replete with uncomputabilities, undecidabilities and non-constructivities – even elements of incompleteness.

Against this ‘potted’ background of pioneering innocence and core issues, the rest of this paper is structured as follows. Some of the key results on uncomputability and undecidability, mostly derived by this author, are summarized in a fairly merciless telegraphic form (with adequate and detailed references to sources) in the next section. In section 3 some remarks on the mathematical underpinnings of these ‘negative’ results are discussed and, again, stated in the usual telegraphic form. The concluding section suggests a framework for invoking my ‘version’ of unconventional computation models for mathematical models of the economy[13].

> dotted.... Then our mathematician Zero will have an objection on the grounds that if we are going to have dotted lines instead of solid lines on the curve then there does not exist any such thing as a slope, or a derivative, or a logarithmic derivative either.....
>
> “If you think in terms of solid lines while the practice is in terms of dots and little steps up and down, this misbelief on your part is worth, I would say conservatively, to the governors of the exchange, at least eighty million dollars per year.” [29], pp.16-34.

[12] *Prefaced, elegantly and appositely, with a typically telling observation by Samuel Johnson*: ”Nothing amuses more harmlessly than computation, and nothing is oftener applicable to real business or speculative enquiries. A thousand stories which the ignorant tell, and believe, die away at once when the computist takes them in his grip”

ibid, p.vii

Surely, this is simply a more literary expression of that famous credo of Leibniz:

“..[W]hen a controversy arises, disputation will no more be needed between two philosophers than between two computers. It will suffice that, pen in hand, they sit down ... and say to each other: Let us calculate.” [19]

[13] An acute observation by one of the referees requires at least a nodding mention. The referee wondered why the paper did not consider the famous ‘Socialist Calculation Debate’,

2. Uncomputability and Undecidability in Economic Theory

Although many of the results described in this section may appear to have been obtained 'cheaply' – in the sense mentioned above – my own reasons for having worked with the aim of locating uncomputabilities, non-constructivities and undecidabilities in core areas of economic theory have always been a combination of intellectual curiosity – along the lines conjectured by Arrow, above –and the desire to make the subject meaningfully quantitative – in the sense suggested by Brown and Stone (op.cit). In the process an explicit research strategy has also emerged, on the strategy of making economic theory consistently algorithmic. The most convincing and admirably transparent example of this research strategy is the one adopted by Michael Rabin to transform the celebrated Gale-Stewart Game to an Algorithmic Game and, then, to characterise its effective content. A full discussion of this particular episode in the development of Computable Economics is given in [48] and [50], chapter 7. However, the various subsections below simply report some of the results I have obtained, on uncomputability, non-constructivity and undecidability in economic theory, without, in each case, describing the background motivation, the precise research and proof strategy that was developed to obtain the result and the full extent of the implications for Computable Economics.

2.1. Undecidability (and Uncomputability) of Maximizing Choice

All of mathematical economics and every kind of orthodox game theory rely on some form of formalized notion of individually 'rational behaviour' Two key results that I derived more than two decades ago, are the following, stated as theorems[14]:

Theorem 1. *Rational economic agents in the sense of economic theory are equivalent to suitably indexed Turing Machines; i.e, decision processes implemented by rational economic agents - viz., choice behaviour - is equivalent to the computing behaviour of a suitably indexed Turing Machine.*

emerging, initially, from careless remarks by Pareto about the computing capabilities of a decentralised market. This issue later – in the 1920s and 1930s, revisited by one of the protagonists as late as 1967 – became a full-blooded debated about the feasibility of a decentralised planning system, an oxymoron if ever there was one. However, the reason I am not considering the debate in this paper is twin-pronged: firstly, it was, essentially, about analog computing (although Oskar Lange muddied the issue in his revisit to the problem in 1967 in the *Dobb Festschrift*); secondly, it is less about computability than computational complexity. For reasons of space, I have had to refrain from any serious consideration of any kind of complexity issue - whether of the computational or algorithmic variety.

[14] A perceptive referee wondered '*why rational choice can be interpreted as an equivalent of the whole class of Turing machines, maybe we should consider in this context only some subclass of Turing machines (e.g. polynomial-time Turing machines)*'. This is an obviously important point, which I have addressed in other writings, in the spirit of Herbert Simon's research program on *boundedly rational choice* and *satisficing decision problems* by economic agents. However, the question of restricting the class of Turing machines to a 'relevant' subclass becomes pertinent when one begins to focus attention on the 'complexity of choice', implemented in empirically relevant contexts. Unfortunately, to go into details of this aspect will require me to expand this paper beyond the allocated constraints and its limited scope.

Put another way, this theorem states that the process of rational choice by an economic agent is equivalent to the computing activity of a suitably programmed Turing Machine.

Proof. Essentially by construction from first principles (no non-constructive assumptions are invoked). See [49]. ■

An essential, but mathematically trivial, implication of this Theorem is the following result:

Theorem 2. *Rational choice, understood as maximizing choice, is undecidable.*

Proof. The procedure is to show, again by construction, that preference ordering is effectively undecidable. See [50], §3.3 for the details. ■

Remark 3. *These kinds of results are the reasons for the introduction of formalized concepts of bounded rationality and satisficing by Herbert Simon. Current practice, particularly in varieties of experimental game theory, to identify boundedly rational choice with the computing activities of a Finite Automaton are completely contrary to the theoretical constructs and cognitive underpinnings of Herbert Simon's framework. The key mistake in current practice is to divorce the definition of bounded rationality from that of satisficing. Simon's framework does not refer to the orthodox maximizing paradigm; it refers to the recursion theorist' s and the combinatorial optimizer's framework of decision procedures.*

2.2. Computable and Decidable Paradoxes of Excess Demand Function

2.2.1. Algorithmic Undecidability of a Computable General Equilibrium

The excess demand function plays a crucial role in all aspects of computable general equilibrium theory and, indeed, in the foundation of microeconomics. Its significance in computable general equilibrium theory is due to the crucial role it plays in what has come to be called *Uzawa's Equivalence Theorem* (cf. [44], §11.4) – the equivalence between a Walrasian Equilibrium Existence Theorem (**WEET**) and the Brouwer Fixed Point Theorem. The finesse in one half of the equivalence theorem, i.e., that **WEET** implies the Brouwer fix point theorem, is to show the feasibility of devising a continuous excess demand function, $X(p)$, satisfying *Walras' Law*,(and homogeneity), from an arbitrary continuous function, say $f(.) : S \to S$, where S is the unit simplex in $\mathbb{R}^N$, such that the equilibrium price vector, p^*, implied by $X(p)$ is also the fix point for $f(.)$, from which it is 'constructed'. For the benefit of those whose memories may well require some rejuvenations,a simple, succinct, version of *Walras' Law* can be stated as follows:

$$\forall p,\ p \cdot X\left(p\right) = \sum_{i=1}^{N} p_i \cdot X_i\left(p\right) = 0. \tag{1}$$

I am concerned, firstly, with the recursion theoretic status of $X(p)$. Is this function computable for arbitrary $p \in S$? Obviously, if it is, then there is no need to use the alleged constructive procedure to determine the Brouwer fix point (or any of the other usual topological fix points that are invoked in general equilibrium theory and CGE Modelling) to locate the economic equilibrium implied by **WEET**.

The key step in proceeding from a given, arbitrary, $f(.) : S \rightarrow S$ to an excess demand function $X(p)$ is the definition of an appropriate scalar:

$$\mu(p) = \frac{\sum_{i=1}^{n} p_i f_i(\frac{p}{\lambda(p)})}{\sum_{i=1}^{n} p_i^2} = \frac{p.f(p)}{\|p\|^2} \tag{2}$$

Where:

$$\lambda(p) = \sum_{i=1}^{n} p_i \tag{3}$$

From (1) and (2), the following excess demand function, $X(p)$, is defined:

$$x_i(p) = f_i(\frac{p}{\lambda(p)}) - p_i \mu(p) \tag{4}$$

i.e.,

$$X(p) = f(p) - \mu(p)p \tag{5}$$

I claim that the procedure that leads to the definition of (3) [or, equivalently, (4)] to determine $p^*$is provably *undecidable*. In other words, the crucial scalar in (1) cannot be defined recursion theoretically (and, *a fortiori*, constructively) to effectivize a sequence of projections that would ensure convergence to the equilibrium price vector.

Clearly, given any $p \in S$, all the elements on the r.h.s of (1) and (2) *seem* to be well defined. However, $f(p)$ is not necessarily computable (nor meaningfully constructive) for arbitrary $p \in S$. Restricting the choice of $f(.)$ to the partial recursive functions may most obviously violate the assumption of *Walras' Law*. Therefore, I shall show that it is impossible to devise an algorithm to define (3) [or (4)] for an arbitrary $f(p)$, such that the equilibrium p^* for the defined excess demand function is also the fix point of $f(.)$. If it were possible, then the famous *Halting Problem for Turing Machines* can be solved, which is an impossibility.

Theorem 4. *$X(p^*)$, as defined in (3) [or (4)] above is undecidable; i.e., cannot be determined algorithmically.*

Proof. *Suppose, contrariwise, there is an algorithm which, given an arbitrary $f(.) : S \rightarrow S$, determines $X(p^*)$. This means, therefore, that the given algorithm determines the equilibrium p^* implied by **WEET**. In other words, given the arbitrary initial conditions $p \in S$ and $f(.) : S \rightarrow S$, the assumption of the existence of an algorithm to determine $X(p^*)$ implies that its* halting *configurations are decidable. But this violates the* undecidability of the Halting Problem for Turing Machines*. Hence, the assumption that there exists an algorithm to determine - i.e., to construct - $X(p^*)$ is untenable.* ■

Remark 5. *The algorithmically important content of the proof is the following. Starting with an arbitrary continuous function mapping the simplex into itself and an arbitrary price vector, the existence of an algorithm to determine*

$X(p^*)$ entails the feasibility of a procedure to choose price sequences in some determined way to check for p^* and to halt when such a price vector is found. Now, the two scalars, μ and λ are determined once $f(.)$ and p are given. But an arbitrary initial price vector p, except for flukes, will not be the equilibrium price vector p^*. Therefore the existence of an algorithm would imply that there is a systematic procedure to choose price vectors, determine the values of $f(.)$, μ and λ and the associated excess demand vector $X(p; \mu, \lambda)$. At each determination of such an excess demand vector, a projection of the given, arbitrary, $f(p)$, on the current $X(p)$, for the current p, will have to be tried. This procedure must continue till the projection for a price vector results in excess demands that vanish for some price. Unless severe recursive constraints are imposed on price sequences - constraints that will make very little economic sense - such a test is algorithmically infeasible. In other words, given an arbitrary, continuous, $f(.)$, there is no procedure - algorithm (constructive or recursion theoretic) - by which a sequence of price vectors, $p \in S$, can be systematically tested to find p^*.

Corollary 6. *The* Recursive Competitive Equilibrium *(**RCE**)of New Classical Macroeconomics – Recursive Macroeconomic Theory – is uncomputable.*

Remark 7. *See [55] (definition 2, p. 16) for a detailed definition of **RCE** and hints on proving this Corollary.*

Remark 8. *The proof procedure is almost exactly analogous to the one used above to show the* recursive undecidability *of a computable general equilibrium – with one significant difference. Instead of using the* unsolvability of the Halting problem for Turing Machines *to derive the contradiction, I use a version of* Rice's Theorem.

Remark 9. *The more empirically relevant question would be to consider the question of the feasibility of approximating $X(p^*)$. This, like the issue of considering a subclass of Turing machines to formalize empirically relevant rational choice procedures, falls under a burgeoning research program on the complexity of computing varieties of economic and game theoretic equilibria. Since I have had to limit the scope of my considerations in this paper to questions of computability and decidability in principle, I must - albeit reluctantly - refrain from going further into these issues. An excellent reference on the problem, via a discussion of the complexity of computing Nash equilibria, can be found in [30].*

2.2.2. Recursive Undecidability of the Excess Demand Function

The nature of economic data and the parameters underpinning the mechanisms generating the data – as noted by Stone and Osborne, if any substantiation of the obvious must be invoked via the wisdom of eminence – should imply that the excess demand function is a Diophantine relation. Suppose we take economic reality, Stone, Osborne and Smale seriously assume that all variables and parameters defining the excess demand functions are, in fact, integer or rational valued (with the former, in addition, being non-negative, as well).

Indeed, Smale has brilliantly articulated the perplexity of the Arrow-Debreu 'subversion' of the classic problem of supply-demand equilibrium as a system of equations to be solved for non-negative valued, rational-number variables, into a system of inequalities whose consistency is proved by blind appeals to non-constructive fix point theorems and, thereby, an existence of a set of equilibrium prices is asserted:

> "We return to the subject of equilibrium theory. The existence theory of the static approach is deeply rooted to the use of the mathematics of fixed point theory. Thus one step in the liberation from the static point of view would be to **use a mathematics of a different kind**. Furthermore, proofs of fixed point theorems traditionally use difficult ideas of algebraic topology, and this has obscured the economic phenomena underlying the existence of equilibria. Also the economic equilibrium problem presents itself most directly and with the most tradition not as a fixed point problem, but as an *equation*, supply equals demand. **Mathematical economists have translated the problem of solving this equation into a fixed point problem**.
>
> I think it is fair to say that for the main existence problems in the theory of economic equilibrium, **one can now bypass the fixed point approach and attack the equations directly to give existence of solutions, with a simpler kind of mathematics** and even mathematics with dynamic and **algorithmic** overtones."
>
> [42], p.290; bold emphasis added.

To 'attack the equations directly,' taking into account the obvious constraints on variables and parameters in economics - i.e., that the variables have to be non-negative, rational numbers and the parameters at least the latter (and if they are not the former, then there are feasible transformations to make them so, cf. [24], chapter 1) – is actually a very simple matter. I shall only indicate the skeleton of such an approach here. Full details are available in the author's other writings.

Now, dividing the vector of parameters and variables characterizing the excess demand function X into two parts, a vector a of parameters and the vector of prices, p, we can write a relation of the form (in supply-demand **equilibrium**)

$$X\left(a_1, a_2, \ldots.., a_n, x_1, x_2, \ldots, x_m\right) = 0$$

where:

Definition 10. *X is a polynomial*[15] *with integer (or rational number) coefficients with respect to the* ***parameters*** *$a_1, a_2, \ldots.., a_n$ and* ***variables***

[15] I am restricting the excess demand functions to be polynomials simply to be consistent with the traditional definition. The more mathematically satisfying approach may have been to consider, in the general case, arbitrary functions from $\mathbb{N}$ to $\mathbb{N}$. I am indebted to a referee's observation regarding the need to clarify this point.

*$x_1, x_2, ..., x_m$ (which are also non-negative) and is called a **parametric Diophantine equation**.*

Definition 11. *X in Definition 8 defines a set $\digamma$ of the parameters for which there are values of the unknowns such that:*

$$\langle a_1, a_2, ..., a_n \rangle \in \Longleftrightarrow \exists x_1, x_2, ..., x_m \left[X \left(a_1, a_2, ..., a_n, x_1, x_2, ..., x_m \right) = 0 \right] \quad (6)$$

Loosely speaking, the relations denoted in the above two definitions can be called *Diophantine representations.* Then sets, such as $\digamma$, having a Diophantine representation, are called simply *Diophantine.* With this much terminology at hand, it is possible to state the fundamental problem of a Diophantine system of excess demand equations as follows:

Problem 12. *A set, say $\langle a_1, a_2, ..., a_n \rangle \in G$, is given. Determine if this set is Diophantine. If it is, find a Diophantine representation for it.*

Of course, the set $\digamma$ may be so structured as to possess equivalence classes of properties, P and relations, R.Then it is possible also to talk, analogously, about a *Diophantine representation of a Property P* or a *Diophantine representation of a Relation R.* For example, in the latter case we have:

$$R \left(a_1, a_2, ..., a_n \right) \Longleftrightarrow \exists x_1, x_2, ..., x_m \left[X \left(a_1, a_2, ..., a_n, x_1, x_2, ..., x_m \right) = 0 \right]$$

Next, how can we talk about the *solvability* of a Diophantine representation of the excess demand relation? This is where undecidability (and uncomputability) will enter – through a remarkable connection with recursion theory, summarized in the next Proposition:

Proposition 13. *Given any parametric Diophantine equation, X, it is possible to construct a Turing Machine, M, such that M will eventually **Halt**, beginning with a representation of the parametric n-tuple, $\langle a_1, a_2, ..., a_n \rangle$, **iff** X in Definition 9 is solvable for the unknowns, $x_1, x_2, ..., x_m$.*

But, then, given the famous result on the *Unsolvability of the Halting problem for Turing Machines*, we are forced to come to terms with the algorithmic unsolvability of the excess demand function as a Diophantine equations.

2.3. Nonconstructivity of Welfare Theorems

Let me conclude this section by showing, in a very general way, the role played by the *Hahn-Banach Theorem* in proving the crucial '*Second Welfare Theorem*' in economics. I shall refer to the way it is presented, proved and discussed in [22] (although I could equally well have chosen the slightly simpler and clearer exposition in [44]). The *Second Welfare Theorem* establishes the proposition that *any Pareto optimum can, for suitably chosen prices, be supported as a competitive equilibrium.* The role of the Hahn-Banach theorem in this proposition is in establishing *the suitable price system.*

Lucas and Stokey state 'their' version of the Hahn-Banach Theorem in the following way[16]:

Theorem 14. *Geometric form of the Hahn-Banch Theorem.*

Let S be a normed vector space; let $A, B \subset S$ be convex sets. Assume:

(a). Either B has an interior point and $A \cap \overset{\circ}{B} = \emptyset$, $\left(\overset{\circ}{B}: \text{ closure of } B\right)$;

(b). Or, S is finite dimensional and $A \cap B = \emptyset$;

Then: $\exists$ a continuous linear functional ϕ, not identically zero on S, and a constant c s.t:

$\phi(y) \leq c \leq \phi(x)$, $\forall x \in A$ *and* $\forall y \in B$.

Next, I state the economic part of the problem in merciless telegraphic form as follows:

There are$I$ consumers, indexed $i = 1, ..., I$;

S is a vector space with the usual norm;

Consumer i chooses from commodity set $X_i \subseteq S$, evaluated according to the utility function $u_i : X_i \rightarrow \Re$;

There are J firms, indexed $j = 1, ..., J$;

Choice by firm j is from the technology possibility set, $Y_j \subseteq S$; (evaluated along profit maximizing lines);

The mathematical structure is represented by the following absolutely standard assumptions:

1. $\forall i, X_i$ *is convex*;
2. $\forall i$, *if* $x, x' \in C_i, u_i(x) > u_i(x')$, *and if* $\theta \in (0,1)$, *then* $u_i[\theta x + (1-\theta) x'] > u_i(x')$;
3. $\forall i, u_i : X_i \rightarrow \Re$ *is continuous*;
4. *The set* $Y = \sum_j Y_j$ *is convex*;
5. *Either the set* $Y = \sum_j Y_j$ *has an interior point*, or S *is finite dimensional*;

Then, the *Second Fundamental Theorem of Welfare Economics* is:

Theorem 15. *Let assumptions* $1-5$ *be satisfied; let* $\left[\left(x_i^0\right), \left(y_j^0\right)\right]$ *be a Pareto Optimal allocation; assume, for some* $h \in \left\{\bar{1}, ...\bar{I}\right\}$, $\exists \hat{x}_h \in X_h$ *with* $u_h(\hat{x}_h) >$

[16] Essentially, the 'classical' mathematician's Hahn-Banach theorem guarantees the extension of a bounded linear functional, say ρ, from a linear subset Y of a separable normed linear space, X, to a functional, η, on the whole space X, with exact preservation of norm; i.e., $|\rho| = |\eta|$. The constructive Hahn-Banach theorem, on the other hand, cannot deliver this pseudo-exactness and preserves the extension as: $|\rho| \leq |\eta| + \varepsilon$, $\forall \varepsilon > 0$. The role of the positive ε in the constructive version of the Hahn-Banach theorem is elegantly discussed by Nerode, Metakides and Constable in their beautiful piece in the Bishop Memorial Volume ([27], pp. 85-91). Again, compare the difference between the 'classical' IVT and the constructive IVT to get a feel for the role of ε.

$u_h(x_h^0)$. *Then* $\exists$ *a continuous linear functional* $\phi : S \to \Re$, *not identically zero on* S, *s.t:*
(a). $\forall i, x \in X_i$ *and* $u_i(x) \geq u_i(x^0) \Longrightarrow \phi(x) \geq \phi(x_i^0)$;
(b). $\forall j, y \in Y_j \Longrightarrow \phi(j) \leq \phi(y_i^0)$;

Anyone can see, as anyone would have seen and has seen for the last 70 years, that an economic problem has been 'mangled' into a mathematical form to conform to the structure and form of a mathematical theorem. This was the case with the way Nash formulated his problems; the way the Arrow-Debreu formulation of the general equilibrium problem was made famous; and legions of others.

It is a pure mechanical procedure to verify that the assumptions of the economic problem satisfy the conditions of the Hahn-Banach Theorem and, therefore, the powerful *Second Fundamental Theorem of Welfare Economics* is 'proved'[17].

The Hahn-Banach theorem does have a constructive version, but only on subspaces of *separable* normed spaces. The standard, 'classical' version, valid on nonseparable normed spaces depends on *Zorn's Lemma* which is, of course, equivalent to the axiom of choice, and is therefore, non-constructive[18].

Schechter's perceptive comment on the constructive Hahn-Banach theorem is the precept I wish economists with a numerical, computational or experimental bent should keep in mind (ibid, p. 135; italics in original; emphasis added).:

> "[O]ne of the fundamental theorems of classical functional analysis is the *Hahn-Banach Theorem; ...* some versions assert the existence of a certain type of linear functional on a normed space X. The theorem is inherently nonconstructive, but a constructive proof can be given for a variant involving normed spaces X that are *separable* – i.e., normed spaces that have a countable dense subset. *Little is lost in restricting one's attention to separable spaces*[19]*, for in applied math most or all normed spaces of interest are separable.* The constructive version of the Hahn-Banach Theorem is more complicated, but it has the advantage that it actually *finds* the linear functional in question."

So, one may be excused for wondering, why economists rely on the 'classical'

[17] To the best of my knowledge an equivalence between the two, analogous to that between the Brouwer fix point theorem and the Walrasian equilibrium existence theorem, proved by Uzawa ([47]), has not been shown.

[18] This is not a strictly accurate statement, although this is the way many advanced books on functional analysis tend to present the Hahn-Banach theorem. For a reasonably accessible discussion of the precise dependency of the Hahn-Banach theorem on the kind of axiom of choice (i.e., whether countable axiom of choice or the axiom of dependent choice), see [26]. For an even better and fuller discussion of the Hahn-Banach theorem, both from 'classical' and a constructive points of view, Schechter's encyclopedic treatise is unbeatable ([38]).

[19] However, it must be remembered that Ishihara, [17], has shown the constructive validity of the Hahn-Banach theorem also for uniformly convex spaces.

versions of these theorems? They are devoid of numerical meaning and computational content. Why go through the rigmarole of first formalizing in terms of numerically meaningless and computationally invalid concepts to then seek impossible and intractable approximations to determine uncomputable equilibria, undecidably efficient allocations, and so on?

Thus my question is: why should an economist *force* the economic domain to be a normed vector space? Why not a *separable normed vector space*? Isn't this because of pure ignorance of constructive mathematics and a carelessness about the nature and scope of fundamental economic entities and the domain over which they should be defined?

2.4. Noneffectivity of Games

The most celebrated exercise in Computable Economics or what has recently come to be called Algorithmic Game Theory is Michael Rabin's famous result:

Theorem 16. ***(Rabin, 1957)*** *There are games in which the player who* ***in theory*** *can always win cannot do so* ***in practice*** *because it is impossible to supply him with* ***effective instructions*** *regarding how he should play in order to win.*

Rabin's strategy to obtain this result is the paradigmatic example of what I conceive to be the typical research program of a Computable Economist. Essentially, the idea is to consider any formal, orthodox, game theoretic example and strip it away of all non-effective considerations and, then, ask whether the remaining scaffolding is capable of being algorithmically decidable in an empirically meaningful sense. A complete description and explanation of Rabin's strategy is fully discussed in [48].

But at the time I first studied Rabin's example – about twenty years ago – and extracted his implicit research strategy as a paradigmatic example for the work of a Computable Economist, I missed an important aspect: its place in a particular tradition of game theory. It was only in very recent times that I have been able to place it in the original tradition of game theory – the tradition that began with Zermelo, before it was 'subverted' by the von Neumann-Nash subjective approach which dominates all current frontiers of research in game theory, at least in the citadel of economic theory (including its computational and experimental branches). My starting point for the tradition that came to a transitory completion, therefore, would be Zermelo's celebrated lecture of 1912 ([58]) and his pioneering formulation of an adversarial situation into an *alternating game* and its subsequent formulation and solution as a mini-max problem by Jan Mycielski in terms of *alternating the existential and universal quantifiers.*

The Zermelo game has no subjective component of any sort. It is an entirely objective game of perfect information, although it is often considered part of the orthodox game theoretic tradition. Let me describe the gist of the kind of game considered by Zermelo, first. In a 2-player game of perfect information, alternative moves are made by the two players, say A and B. The game, say as in Chess, is played by each of the players 'moving' one of a finite number of counters available to him or her, according to specified rules, along a 'tree' - in the case of Chess, of course, on a board of fixed dimension, etc. Player A, say, makes the first move (perhaps determined by a chance mechanism) and places one of the counters, say $a_0 \in A_0$, on the designated 'tree' at some allowable position (again, for evocative purposes, say as in Chess or any other similar board game); player B, then, observes the move made by A - i.e., observes, with perfect recall, the placement of the counter a_1 - and makes the second move by placing, say $b_1 \in B_1$, on an allowable position on the 'board'; and so on. Let us suppose these alternating choices terminate after Player B's $n - th$ move; i.e., when $b_n \in B_n$ has been placed in an appropriate place on the 'board'.

Definition 17. *A **play** of such a game consists of a sequence of such alternative moves by the two players*

Suppose we label the alternating individual moves by the two players with the natural numbers in such a way that:

1. The even numbers, say, $a(0), a(2), ..., a(n-1)$ enumerate player A's moves;
2. The odd numbers, say, $b(1), b(3), ..., b(n)$ enumerate player B's moves;
    - Then, each (finite) play can be expressed as a sequence, say γ, of natural numbers.

Suppose we define the set α as the set of plays which are wins for player A; and, similarly, the set β as the set of plays which are wins for player B.

Definition 18. *A strategy is a function from any (finite) string of natural numbers as input generates a single natural number, say σ, as an output.*

Definition 19. *A game is **determined** if one of the players has a winning strategy; i.e., if either $\sigma \in \alpha$ or $\sigma \in \beta$.*

Theorem 20. *Zermelo's Theorem: $\exists$ a winning strategy for player A, whatever is the play chosen by B; and vice versa for B*[20]

Remark 21. *This is Zermelo's version of a minimax theorem in a perfect recall, perfect information, game.*

It is in connection with this result and the minimax form of it that Steinhaus observed, with considerable perplexity:

[20] One referee found this way of stating the celebrated 'Zermelo Theorem' somewhat 'unclear'. The best I can do, to respond to the referee's gentle – albeit indirect – admonition to state it more intuitively is to refer to the excellent pedagogical discussion, and a particularly lucid version, of the Zermelo Theorem in [45].

> “[My] inability [to prove the minimax theorem] was a consequence of the ignorance of Zermelo’s paper ([58]) in spite of its having been published in 1913. .... J von Neumann was aware of the importance of the minimax principle (cf. [57]); it is, however, *difficult to understand the absence of a quotation of Zermelo’s lecture in his publications.*”
>
> Steinhaus ([45], p. 460; italics added)

Why didn’t von Neumann refer, in 1928, to the Zermelo-tradition of alternating games? The tentative answer to such a question is a whole research program in itself and I will simply have to place it on an agenda and pass on. I have no doubts whatsoever that any serious study to answer this almost rhetorical question will reap a rich harvest of further cons perpetrated by the mathematical economists, perhaps inadvertently. The point I wish to make is something else and has to do with the axiom of choice and its place in *economic conning*. So, let me return to this theme.

Mycielski (cf., [45], pp. 460-1) formulated the Zermelo minimax theorem in terms of alternating logical quantifiers as follows[21]:

$$\sim \left\{ \bigcup_{a_0 \in A_0} \bigcap_{b_1 \in B_1} \cdots\cdots\cdots \bigcup_{a_n \in A_{n-1}} \bigcap_{b_n \in B_n} (a_0 b_1 a_2 b_3 \ldots.. a_{n-1} b_n) \right\} \in \alpha$$

$$\Longrightarrow \left\{ \bigcap_{a_0 \in A_0} \bigcup_{b_1 \in B_1} \cdots\cdots\cdots \bigcap_{a_n \in A_{n-1}} \bigcup_{b_n \in B_n} (a_0 b_1 a_2 b_3 \ldots.. a_{n-1} b_n) \right\} \notin \beta$$

Now, summarizing the structure of the game and taking into account Mycielski’s formulation in terms of alternating we can state as follows:

1. The sequential moves by the players can be modelled by alternating existential and universal quantifiers.
2. The existential quantifier moves first; if the total number of moves is odd, then an existential quantifier determines the last chosen integer; if not, the universal quantifier determines the final integer to be chosen.
3. One of the players tries to make a logical expression, preceded by these alternating quantifiers *true*; the other tries to make it *false*.
4. Thus, inside the braces the win condition in any play is stated as a proposition to be satisfied by generating a number belonging to a given set.
5. If, therefore, we can extract *an arithmetical form* - since we are dealing with sequences of natural numbers - for the win condition it will be possible to discuss recursive solvability, decidability and computability of winning strategies.

The above definitions, descriptions and structures define, therefore, an *Arithmetical Game* of length n (cf. [50], pp. 125-6 for a formal definition). Stating the Zermelo theorem in a more formal and general form, we have:

[21] Discerning and knowledgeable readers will recognize, in this formulation, the way Gödel derived undecidable sentences.

Theorem 22. *Arithmetical Games of finite length are determined.*

Remark 23. *However, the qualifications required by Harrop's Theorem (see below) have to be added as 'constructive' caveats to this result*[22].

The more general theorem, for games of arbitrary (non-finite) length, can be proved by standard diagonalization arguments and is[23]:

Theorem 24. *Arithmetical Games on any countable set or on any set which has a countable complement is determined.*

Now, enter the axiom of choice! Suppose we allow any unrestricted sets α and β. Then, for example if they are *imperfect sets*[24], the game is not determined. If we work within ZFC, then such sets are routinely acceptable and lead to games that cannot be determined - even if we assume perfect information and perfect recall. Surely, this is counter-intuitive? For this reason, this tradition in game theory chose to renounce the axiom of choice and work with an alternative axiom that restricts the class of sets within which arithmetical games are played. The alternative axiom is the *axiom of determinacy*, introduced by Steinhaus:

Axiom 25. *The **Axiom of Determinacy**: Arithmetical Games on every subset of the Baire line*[25] *is determined.*

The motivation given by Steinhaus ([45], pp. 464-5) is a salutary lesson for mathematically minded economists or economists who choose to accept the axiom of choice on 'democratic' principles or economists who are too lazy to study carefully the economic meaning of accepting a mathematical axiom:

> "It is known that [the Axiom of Choice] produces such consequences as the decomposition of a ball into five parts which can be put together to build up a new ball of twice the volume of the old one [the Banach-Tarski paradox], a result considered as paradoxical by many scientists. There is another objection: how are we to speak of perfect information for [players] A and B if it is impossible to verify whether both of them think of the same set when they speak of [α]? *This impossibility is inherent in every set having only [the Axiom of Choice] as its certificate of birth.* In such circumstances it is doubtful whether human beings will ever play really [an infinite game].
>
> *All these considerations impelled me to place the blame on the Axiom of Choice.* Sixty years of the theory of sets have elapsed since

[22] I am indebted to a referee for making me think about this important point.

[23] The real time paradox of implementing an infinite play is easily resolved (cf., [45], pp. 465; [50], chapter 7).

[24] A set G is a *perfect set* if it is a *closed set in which every point is a limit point.*

[25] A Baire line is an irrational line which, in turn, is a line obtainable from a continuum by removing a countable dense subset.

> this Axiom was proclaimed, and some ideas have .... convinced me that a purely negative attitude against [the Axiom of Choice] would be dangerous to propose. Thus I have chosen the idea of replacing [the Axiom of Choice] by the [above Axiom of Determinacy].
> italics added.

There is a whole tradition of game theory, beginning at the beginning, so to speak, with Zermelo, linking up, via Rabin's modification of the Gale-Stewart infinite game, to recursion theoretic formulations of arithmetical games underpinned by the *axiom of determinacy* and completely independent of the *axiom of choice* and **eschewing all subjective considerations**. In this tradition notions of *effective playability*, *solvability* and *decidability* questions take on fully meaningful computational and computable form where one can investigate whether it is feasible to instruct a player, who is known to have a winning strategy, to actually select a sequence to achieve the win.

3. *Towards* Unconventional Computational Models in Economic Theory

My personal view – indeed, *vision* – has been evolving, very gradually, towards a mathematical economics that is formalized exclusively in terms of strict Brouwerian Constructive Mathematics. To make this case, from methodological and epistemological points of view would require more space than I have at my disposal and, moreover, would necessitate a widening of the scope of the paper. Therefore, I shall only make a few salient observations that may indicate the reasons for this *vision.*

Even more than a justification from an algorithmic point of view, for which it is possible to make almost equally strong cases for either a classical computability or Bishop-style constructivist approaches towards a consistent quantitative formalization of economic theory, there is the epistemological question of the meaning of proving economic propositions using classical, non-constructive, logic. To the best of my knowledge, there is only one serious, fundamental, work in economic theory that is consistently constructive in the way the economic propositions in it are proved: Piero Sraffa's *magnum opus*, [43]. I have dissected the methods of proof in this elegant, terse, text and made my case for a constructive epistemology in the formalization of economic theory (see [56]). However, from a strictly methodological point of view, I remain indifferent between formalizing economic entities and processes recursion theoretically or constructively, either of them in any of the many variants in which they are being developed these days.

Whether methodologically or epistemologically, a formalization of economic theory via classical recursion theory or any variant of constructive mathematics, will have to lead to fairly drastic rethinking of fundamental issues in economics from policy and empirical points of view. This is primarily because the welfare theorems and computable properties of equilibria lose their quantitative underpinnings and become almost mystically sustained. The issues I have chosen to dissect in this paper suggest this implication, sometimes explicitly, but more than often only implicitly. The following remarks are somewhat limited further elaborations of these issues, but primarily from a methodological standpoint.[26]

I am not in any way competent in any form of *unconventional models of computation.* The remarks below should, therefore, be taken as reflections of a Computable Economist who is deeply committed to making economics algorithmically meaningful – from a methodological point of view – so that computation and experimentation can be seriously and rigorously underpinned in the honest mathematics of the computer, whether digital or analog.

My original motivation for coining the term *Computable Economics*, to encapsulate the kind of issues raised by Arrow and Stone, mentioned in the opening section, was quite a different kind of perplexity. It was a perplexity grounded in *proof theory* and *model theory*[27]. There is no better way to summarize these originating concerns, from the perspective of a *Computable Economist*, than to recall two deep and subtle caveats added by two of the pioneers of computability theory, Alonzo Church and Emil Post, in their pathbreaking contributions to the origins of classical and higher recursion theory. Their insights suggested a deeper interplay between computability and constructivity than is normally understood or acknowledged by any of the social scientists now deeply immersed in developing the frontiers of computable economics, algorithmic game theory and algorithmic statistics.

To place these insights in the context of the present paper, let me state a conjecture in the form of a theorem[28]:

Theorem 26. *Nash equilibria of finite games are constructively indeterminate*[29]*.*

Proof. Apply Harrop's Theorem ([15], p.136). ■

To make sense of this 'theorem', and its proof using 'Harrop's Theorem', it is necessary to understand the subtle differences between computability as understood in (classical) recursion theory, accepting Church's Thesis[30] and computations by algorithmic mathematics as specified in varieties of constructive mathematics, particularly intuitionistic constructive mathematics. I shall not

[26] The brief few opening paragraphs of this concluding section were added in response to observations made by two very helpful referees.

[27] To a large extent in their incarnation as *constructive* and *non-standard* analysis.

[28] These thoughts were inspired entirely by a reading of a fundamental series of results by Francisco Doria and his collaborators (cf. [10] and [11]), which introduced me to Harrop's important work ([15]). These papers put in proper perspective my initial proof- and model-theoretic perplexities when faced with assumptions and proofs in mathematical economics.

[29] The proof of the existence of Nash equilibria, given in standard textbooks, rely on one or another of the nonconstructive fix point theorems (Brouwer's, Kakutani's etc.). Since these, in turn, are proved invoking the Bolzano-Weierstrass theorem, which is intrinsically nonconstructive, due to essentially undecidable disjunctions, the proof of the non-constructivity of Nash equilibria for general infinite games is 'easy' – and 'cheap' – at least from one point of view.

[30] Or the Church-Turing Thesis.

enter into the deep domain of the foundations of mathematics and its thorny controversies here - although I, too, have my view and take my 'sides' and find myself, as always, in the minority! The subtle issues that have to be clarified, to make sense of the above almost counter-intuitive 'theorem', were made clear in Charles Parson's 'review', [31] of Harrop's result[31]. The issues are the bearing of Harrop's Theorem on, whether:

1. Every finite set is recursive;
2. Every recursive set is effectively decidable;
3. Every finite set is effectively decidable;

This neat three-point characterization of Harrop's theorem, by Parsons, is a summary of the following explanation of the implications of his Theorem by Harrop himself:

> "Although it is correct to classically to state that the values of a partial function computed by a machine of arbitrary Gödel number form of a finite set or an infinite set, this statement should *not* be used together with the statements that *every finite set is recursive* and that *every recursive set has an intuitively effective test for membership* (*converse of Church's thesis*) to conclude that if we know classically that a certain integer is the Gödel number of a machine which computes a function with a finite range then automatically there is an intuitively effective test for membership of that range. Our theorem shown that as far as the general case is concerned there is no recursive method for obtaining machines which will compute the characteristic functions which would all individually be obtainable if there were such intuitively effective tests. There may in any particular case be an intuitively effective test."
>
> ibid, p.139; italics added.

Now how this links up with the early contributions by Church and Post to the defining frameworks for classical and higher recursion theory, can easily be gauged by two caveats they made in two of their classic writings ([5], [33], respectively). Church observed (*ibid*, p. 351):

> "It is clear that for any recursive function of positive integers there exists an algorithm using which any required particular value of the function can be effectively calculated."

To this almost innocuous observation – 'innocuous' at least to the modern 'classical' recursion theorist – Church added the subtle (I almost wrote 'slightly devious' – but no one can possibly accuse Alonzo Church of being 'devious'!!) caveat:

[31] And the subsequent development and simplification of Harrop's result and proof by Jiří Hořješ ([16]).

> "The reader may object that this algorithm cannot be held to provide an effective calculation of the required particular values of F^i *unless the proof is constructive* that the required equation $f_{n_i}{}^i(k_1{}^i, , k_2{}^i, ..., k_n{}^i) = k^i$ will ultimately be found. But if so this merely means that he should take the existential quantifier which appears in our definition of a set of recursion equations *in a constructive sense. What the criterion of constructiveness shall be is left to the reader.*" ibid, p.351, footnote 10; italics added.

Post, analogously, first states what may seem obvious to a modern 'classical' recursion theorist *([33]*, p. 469; italics in the original):

> "Clearly, *any finite set of positive integers is recursive.* For if $n_1, n_2, ..., n_\nu$ are the integers in question, we can test n being, not being, in the set by directly comparing it with $n_1, n_2, ..., n_\nu$."

But, then, goes on with *his* caveat:

> "*The mere*[32] *existence of a general recursive function defining the finite set is in question.* Whether, given some definition of the set, we can *actually discover* what the members thereof are, is *a question for a theory of proof* rather than for the present theory of finite processes. For sets of finite sets the situation is otherwise, ..."
>
> ibid, p.469, footnote 10; italics added.

Harrop's theorem's clarifies these caveats and drives the wedge between computabilty, classically understood, and algorithms as understood by (at least) some constructivists. I believe this 'wedge' allows the Computable Economist to seek 'unconventional models of computation' – i.e., going beyond or, at least, sideways from, the phenomenological limits imposed by the Church-Turing Thesis.

Whether the institutions and mechanisms of a market economy make feasible such 'unconventional models of computation', depending on the 'wedge' between the Church-Turing Thesis and its 'converse', is not something that is formally decidable – let alone algorithmically decidable. But that does not mean the market mechanism is not actually involved in 'unconventional computation'. However, to make sense of this question it will be necessary to algorithmise orthodox economic theory - or, even better, develop an algorithmic economics, *ab initio*. To this extent I find it sobering to contemplate on an analogy between the lessons to be learned from Abel's impossibilities – and Gödel's:

> "Why was it that, in His infinite wisdom, God should have created algebraic solutions for general equations of the first four degrees, but not for

[32] I suspect a reading of this sentence by replacing 'mere' with 'very' will make the sense more accurate!

> the equation $ax^5 + bx^4 + cx^3 + dx^2 + ex + f = 0$? Is it the case that human powers are too limited to understand such a transcendent matter? Or have we simply not yet ascended to the 'meta-mathematical' level in which comprehension will be forthcoming? If Abel's proof was spared such conundrums Gödel's theorem unfortunately was not;.... . For while Gödel's theorem looks like – and was initially intended to be seen as – a *closure*, it has been widely interpreted as a *transitional* impossibility proof."
>
> [40], p.167; italics in original

The 'fallacy of composition' that drives a felicitous wedge between micro and macro, between the individual and the aggregate, and gives rise to *emergent* phenomena in economics, non-algorithmic ways – as conjectured, originally more than a century and a half ago – by John Stuart Mill ([25]) and George Herbert Lewes ([20]), and codified by Lloyd Morgan in his *Gifford Lectures* ([21]) – may yet be tamed by *unconventional models of computation.*

References

[1] Arrow, Kenneth J (1986), "*Rationality of Self and Others in an Economic System*", The Journal of Business, Vol. 59, #4, pt. 2, October, pp. $S385 - S399$.

[2] Blum, Lenore, Felipe Cucker, Michael Shub and Steve Smale (1998), Complexity and Real Computation, Springer-Verlag, New York and Berlin.

[3] Brown, Alan and Richard Stone (1962), A Computable Model of Economic Growth, *A Programme for Growth: Volume 1*, Department of Applied Economics, University of Cambridge, Chapman and Hall, London.

[4] Canova, Fabio (2007), Methods for Applied Macroeconomic Research, Princeton University Press, Princeton, NJ.

[5] Church, Alonzo (1936), "*An Unsolvable Problem of Elementary Number Theory*", The American Journal of Mathematics, Vol. 58, pp. 345 – 364.

[6] Clower, Robert W (1994), "*Economics as an Inductive Science*", Southern Economic Journal, Vol. 60, #4, PP. 805 – 814.

[7] Condon, Ann (1989), Computational Models of Games (*An ACM Distinguished Dissertation*), The MIT Press, Cambridge, Mass.

[8] Cooper, S Barry (2004), Computability Theory, Chapman & Hall/CRC Mathematics, Boca Raton and London.

[9] Cootner, Paul H (Editor) (1964, [2000]), The Random Character of Stock Market Prices, Risk Publications, London.

[10] da Costa, N. C. A, Francisco A. Doria and Marcelo Tsuji (1995), "*The Undecidability of Formal Definitions in the Theory of Finite Groups*", Bulletin of the Section of Logic, Polish Academy of Sciences, Vol. 24, pp. 56 – 63.

[11] da Costa, N. C. A, Francisco A. Doria and Marcelo Tsuji (1998), "*The Incompleteness Theories of Games*", Journal of Philosophical Logic, Vol. 27, pp. 553 – 568.

[12] Davis, Mark and Alison Etheridge (2006), Louis Bachelier's Theory of Speculation: The Origins of Modern Finance, Translated and with a Commentary by Mark Davis & Alison Etheridge, Foreward by Paul A. Samuelson, Princeton University Press, Princeton, NJ.

[13] Gács, Péter, John T. Tromp and Paul M. B. Vitányi, (2001), "*Algorithmic Statistics*", IEEE Transactions on Information Theory, Vol.47, #6, September, pp.2443 – 63.

[14] Goodstein, R. L (1948), A Text-Book of Mathematical Analysis: The Uniform Calculus and its Applications, The Clarendon Press, Oxford.

[15] Harrop, Ronald (1961), "*On the Recursivity of Finite Sets*", Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, Bd, 7, pp. 136 – 140.

[16] Hořejš, Jiří, (1964), "*Note on Definition of Recursiveness*", Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, Bd 10, pp. 119 – 120.

[17] Ishihara, Hajime (1989), "*On the Constructive Hahn-Banach Theorem*", Bulletin of the London Mathematical Society, Vol. 21, pp.79-81.

[18] Judd, Kenneth L (1998), Numerical Methods in Economics, The MIT Press, Cambridge, Massachusetts.

[19] Leibniz, G. W (1686, [1965]), "*Universal Science: Characteristic XIV, XV*", in: Monadology and other Philosophical Essays, translated by P. Schrekker, Bobbs-Merril, Indianapolis, IN.

[20] Lewes, George Henry (1891), Problems of Life and Mind, *First Series: The Foundations of a Creed,* The Riverside Press, Cambridge.

[21] Lloyd Morgan, C (1927), Emergent Evolution, *The Gifford Lectures*, Williams and Norgate, London.

[22] Lucas, Robert. E, Jr., and Nancy L. Stokey with Edward C. Prescott, (1989), Recursive Methods in Economic Dynamics, Harvard University Press, Cambridge, Massachusetts.

[23] Marimon, Ramon and Andrew Scott (Editors), (1999), Computational Methods for the Study of Dynamic Economies, Oxford University Press, Oxford.

[24] Matiyasevich, Yuri M (1993), Hilbert's Tenth Problem, The MIT Press, Cambridge, Mass.

[25] Mill, John Stuart (1843, [1890]), A System of Logic, *Ratiocinative and Inductive - being a Conected View of the Principles of Evidence and the methods of Scientific Investigation*, Eighth Edition, Harper & Brothers Publishers, New York.

[26] Narici, Lawrence & Edward Beckenstein (1997), "*The Hahn-Banach Theorem*", Topology and its Applications, Vol. 77, pp. 193-211.

[27] Nerode, Anil, George Metakides and Robert Constable, (1985), *Recursive Limits on the Hahn-Banach Theorem*, in: Errett Bishop - Reflections on Him and His Research, pp.85-91, edited by Murray Rosenblatt, *Contemporary Mathematics*, Vol. 39, American Mathematical Society, Providence, Rhode Island.

[28] Nisan, Noam, Tim Roiughgarden, Éva Tardos and Vijay V. Vazirani (editors), (2007), Algorithmic Game Theory, Cambridge University Press, Cambridge.

[29] Osborne, M. S. M (1977, [1995]), The Stock Market and Finance from a Physicist's Viewpoint, Crossgar Press, Minneapolis, MN.

[30] Papadimitriou, Christos, H (2007), "*The Complexity of Finding Nash Equilibria*", Chapter 2, pp. 29-51, in: Algorithmic Game Theory, edited by Noam Nisan, Tim Roughgarden, Éva Tardos and Vijay V. Vazirani, Cambridge University Press, Cambridge.

[31] Parsons, Charles (1968), "*On the Recursivity of Finite Sets: A Review*", The Journal of Symbolic Logic, Vol. 33, No. 1, March, p. 115.

[32] Penrose, Roger (1989), The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics, Oxford University Press, Oxford.

[33] Post, Emil L (1944), "*Recursively Enumerable Sets of Positive Integers and their Decision Problems*", Bulletin of the American Mathematical Society, Vol. 50, No. 5, may, pp. 284 – 316.

[34] Putnam, Hilary (1967, [1975]), "*The Mental Life of Some Machines*", in H. Castaneda (ed.), Intensionality, Minds and Perception, Wayne University Press, Detroit; reprinted in: Mind, Language and Reality - Philosophical Papers: Vol. 2, by Hilary Putnam, chapter 20, pp. 408 – 428, Cambridge University Press, Cambridge.

[35] Rabin, Michael O, (1957), "*Effective Computability of Winning Strategies*", in: Annals of Mathematics Studies, No. 39: Contributions to the Theory of Games, Vol. III, edited by M. Dresher, A. W. Tucker and P. Wolfe, pp. 147–157, Princeton University Press, princeton, NJ.

[36] Ruelle, David (1988), "*Is Our Mathematics Natural? The Case of Equilibrium Statistical Mechanics*", Bulletin (New Series) of The American Mathemtical Society, Vol. 19, #1, Juy, pp. 259 – 268.

[37] Ruelle, David (2007), The Mathematician's Brian, Princeton University Press, Princeton, NJ.

[38] Schechter, Eric, (1997), Handbook of Analysis and Its Foundations, Academic Press, San Diego.

[39] Shafer, Glenn and Vladimir Vovk (2001), Probability and Finance – It's Only a Game!, John Wiley & Sons, Inc., New York.

[40] Shenker, S. G (1989), "*Wittgenstein's Remarks on the Significance of Gödel's Theorem*", Chapter VIII, pp. 155 – 256, in: Gödel's Theorem in Focus, edited by S.G. Shenker, Routledge, London.

[41] Schwartz, Jacob. T (1986), *The Pernicious Influence of Mathematics on Science*, in: Discrete Thoughts - Essays on Mathematics, Science, and Philosophy by Mark Kac, Gian-Carlo Rota and Jacob T. Schwartz, Birkhaüser, Boston.

[42] Smale, Steve (1976), "*Dynamics in General Equilibrium Theory*", American Economic Review, Vol. 66, No.2, May, pp.288 – 94.

[43] Sraffa, Piero (1960), Production of Commodities by Means of Commodities: Prelude to a Critique of Economic Theory, Cambridge University Press, Cambridge.

[44] Starr, Ross M (1977), General Equilibrium Theory: An Introduction, Cambridge University Press, Cambridge.

[45] Steinhaus, H (1965), *Games, An Informal Talk*, The American Mathematical Monthly, Vol. 72, No. 5, May, pp. 457– 468.

[46] Svozil, Karl (1993), Randomness & Undecidability in Physics, World Scientific, Singapore.

[47] Uzawa, Hirofumi (1962), "*Walras' Existence Theorem and Brouwer's Fixed Point Theorem*", The Economic Studies Quarterly, Vol. 8, No. 1, pp. 59 – 62.

[48] Velupillai, K. Vela (1997), "*Expository Notes on Computability and Complexity in (Arithmetical) Games*", Journal of Economic Dynamics and Control, Vol.21, No.6, June, pp 955 – 979.

[49] Velupillai, K. Vela (1999), "*Non-maximum Disequilibrium Rational Behaviour*", Economic Systems Research, Vol. 11, # 2, pp. 113 – 126.

[50] Velupillai, K. Vela (2000), Computable Economics, Oxford University Press, Oxford.

[51] Velupillai, K. Vela (2005), "*The Unreasonable Ineffectiveness of Mathematics in Economics*", Cambridge Journal of Economics, Vol. 29, #.6, November, PP. 849 – 872.

[52] Velupillai, K. Vela (2006), "*Algorithmic Foundations of* Computable *General Equilibrium Theory*", Applied Mathematics and Computation, Vol. 179, # 1, August, pp. 360 – 369.

[53] Velupillai, K. Vela (2007), "*Taming the Incomputable, Reconstructing the Nonconstructive and Deciding the Undecidable in Mathematical Economics*", Forthcoming in: New Mathematics and Natural Computation, Vol. 4.

[54] Velupillai, K. Vela (2008), "*A Computable Economist's Perspective on Computational Complexity*", Forthcoming in: The Handbook of Complexity Research, edited by J. Barkley Rosser, Jr., Edward Elgar Publishing Ltd., Cheltenham.

[55] "*The Mathematization of Macroeconomics - A* Recursive Revolution*"*, forthcoming in: Economia Politica, Vol. XXV, #2.

[56] "*Sraffa's Mathematical Economics - A* Constructive *Interpretation*", forthcoming in: The Journal of Economic Methodology, Vol. 15, #4.

[57] von Neumann, J (1928), *Zur Theorie der Gesellsschaftsspiele*, Mathematische Annalen, Vol. 100, pp. 295 – 320.

[58] Zermelo, Ernst (1912), "*Über ein Anwendung der Mengenlehre auf die Theorie des Schachspiels*", Proceedings of the Fifth International Congress of Mathematicians, Cambridge, Vol. 2, pp. 501– 4.

Optical computing [⋆]

Damien Woods [a]

[a] *Department of Computer Science and Artificial Intelligence*
University of Seville, Spain

Thomas J. Naughton [b,c]

[b] *Department of Computer Science*
National University of Ireland, Maynooth, County Kildare, Ireland

[c] *University of Oulu, RFMedia Laboratory, Oulu Southern Institute, Vierimaantie 5, 84100 Ylivieska, Finland*

Abstract

We consider optical computers that encode data using images and compute by transforming such images. We give an overview of a number of such optical computing architectures, including descriptions of the type of hardware commonly used in optical computing, as well as some of the computational efficiencies of optical devices. We go on to discuss optical computing from the point of view of computational complexity theory, with the aim of putting some old, and some very recent, results in context. Finally, we focus on a particular optical model of computation called the continuous space machine. We describe some results for this model including characterisations in terms of well-known complexity classes.

Key words:
PACS:

1. Introduction

In this survey we consider optical computers that encode data using images and compute by transforming such images. We try to bring together, and thus give context to, a large range of architectures and algorithms that come under the term optical computing.

[⋆] DW acknowledges support from Junta de Andalucía grant TIC-581. TN acknowledges support from the European Commission Framework Programme 6 through a Marie Curie Intra-European Fellowship.

Email addresses: d.woods@cs.ucc.ie (Damien Woods), tomn@cs.nuim.ie (Thomas J. Naughton).
URLs: http://www.cs.ucc.ie/∼dw5 (Damien Woods), http://www.cs.nuim.ie/∼tnaughton/ (Thomas J. Naughton).

In Section 2 we begin by stating what we mean by the term optical computing, and we discuss some common optical computing architectures. Unlike a number of other areas of nature inspired computing, optical computers have existed (mostly in laboratories) for many years, and so in Section 3 we describe hardware components that are found in many optical computing systems. In Section 4 we describe some of the physical principles that underlie the parallel abilities and efficiencies found in optical computing. These include high fan-in, high interconnection densities, and low energy consumption.

In order to understand the abilities of optics as a computing medium we would argue that computational complexity theory is an indispensable tool. So in Section 5 we collect a number of existing algorithmic results for optical computers. In particular we focus on algorithms and models for matrix-vector multipliers, and some other architectures. We also offer suggestions for future work directions for optical algorithm designers. In Section 6, we take a detailed look at a particular model of optical computing (the continuous space machine, or CSM) that encompasses most of the functionality that coherent optical information processing has to offer. We begin by defining the CSM and a total of seven complexity measures that are inspired by real-world (optical) resources. We go on to discuss how the CSM's operations could be carried out physically. Section 7 contains some example datastructures and algorithms for the CSM. In Section 8 we motivate and introduce an important restriction of the model called the $\mathcal{C}_2$-CSM, and in Section 9 we briefly describe a number of $\mathcal{C}_2$-CSM computational complexity results, and their implications.

2. Brief overview of optical computing architectures

Traditionally, in optical information processing a distinction was made between signal/image processing through optics and numerical processing through optics, with only the latter (and often only the digital version of the latter) being called optical computing [57,48,29,101]. However, it was always difficult to clearly delineate between the two, since it was largely a question of the interpretation the programmer attached to the output optical signals. The most important argument for referring to the latter only as optical computing had to do with the fact that the perceived limits (or at least, ambitions) of the former was simply for special-purpose signal/image processing devices while the ambitions for the latter was general-purpose computation. Given recent results on the computational power of optical image processing architectures [63,98,93], it is not the case that such architectures are limited to special-purpose tasks.

Furthermore, as the field become increasingly multidisciplinary, and in particular as computer scientists play a more prominent role, it is necessary to bring the definition of optical computing in line with the broad definition of computing. In particular, this facilitates analysis from the theoretical computer science point of view. The distinction between analog optical computing and digital optical computing is similarly blurred given the prevalence of digital multiplication schemes effected through analog convolution [48]. Our broad interpretation of the term optical computing has been espoused before [19].

2.1. *Optical pattern recognition*

Arguably, optical computing began with the design and implementation of optical systems to arbitrarily modify the complex valued spatial frequencies of an image. This concept, spatial filtering [67,86,89], is at the root of optics' ability to perform efficient convolution and correlation operations. In a basic pattern recognition application, spatial filtering is called matched filtering, where a filter is chosen that matches (i.e. conjugates) the spectrum of the sought object. Employing this operation for advanced pattern recognition [12,18,43,47], effort focused on achieving systems invariant to scaling, rotation, out-of-plane rotation, deformation, and signal dependent noise, while retaining the existing invariance to translating, adding noise to, and obscuring parts of the input. Effort also went into injecting nonlinearities into these inherently linear systems to achieve wider functionality [46]. Improvements were made to the fundamental limitations of the basic matched filter design, most notably the joint transform correlator architecture [92].

Optical correlators that use incoherent sources of illumination (both spatially and temporally) rather than lasers are also possible [11,71]. The simplest incoherent correlator would have the same basic spatial filtering architecture as that used for matched filtering. While coherent systems in principle are more capable than incoherent systems (principally because the former naturally represents complex functions while the latter naturally represents real functions), incoherent systems require less precise positioning when it comes to system construction and are less susceptible to noise.

Trade-offs between space and time were proposed and demonstrated. These included time integrating correlators [90] (architecturally simpler linear time variants of the constant time space integrating matched filter mentioned already) and systolic architectures [21]. In addition to pattern recognition, a common application for these classes of architectures was numerical calculation.

2.2. *Analog optical numerical computation*

An important strand of image-based optical computation involved numerical calculations: analog computation as well as multi-level discrete computation. Matrix-vector and matrix-matrix multiplication systems were proposed and demonstrated [42,57,90,48,4,29,51]. The capability to expand a beam of light and to focus many beams of light to a common point directly corresponded to high fan-out and fan-in capabilities, respectively. The limitations of encoding a number simply as an intensity value (finite dynamic range and finite intensity resolution in the modulators and detectors) could be overcome by representing the numbers in some base. Significant effort went into dealing with carry operations so that in additions, subtractions, and multiplications each digit could be processed in parallel. Algorithms based on convolution to multiply numbers in this representation were demonstrated [48], with a single post-processing step to combine the sub-calculations and deal with the carry operations.

An application that benefited greatly from the tightly-coupled parallelism afforded by optics was the solving of sets of simultaneous equations and matrix inversion [17,1]. An application that, further, was tolerant to the inherent inaccuracies and noise of analog optics was optical neural networks [28,20] including online neural learning in the presence of noise [62].

2.3. *Digital optical computing*

The next major advances came in the form of optical equivalents of digital computers [44]. The flexibility of digital systems over analog systems in general was a major factor behind the interest in this form of optical computation [78]. Specific drawbacks of the analog computing paradigm in optics that this new paradigm addressed included no perceived ability to perform general purpose computation, accumulation of noise from one computation step to another, and systematic errors introduced by imperfect analog components. The aim was to design digital optical computers that followed the same principles as conventional electronic processors but which could perform many binary operations in parallel. These systems were designed from logic gates using nonlinear optical elements: semitransparent materials whose transmitted intensity has a nonlinear dependence on the input intensity.

Digital optical computing was also proposed as an application of architectures designed originally for image-based processing, for example logic effected through symbolic substitution [10]. At the confluence of computing and communication, optical techniques were proposed for the routing of signals in long-haul networks [29,101].

3. Optical computing hardware

The three most basic hardware components of an optical information processing system are a source, a modulator, and a detector. A source generates the light, a modulator multiplies the light by a (usually, spatially varying) function, and a detector senses the resulting light. These and others are introduced in this section.

3.1. *Sources*

Lasers are a common source of illumination because at some levels they are mathematically simpler to understand, but incoherent sources such as light-emitting diodes are also used frequently for increased tolerance to noise and when nonnegative functions are sufficient for the computation. Usually, the source is monochromatic to avoid the problem of colour dispersion as the light passes through refracting optical components, unless this dispersion is itself the basis for the computation.

3.2. *Spatial light modulators (SLMs)*

It is possible to encode a spatial function (a 2D image) in an optical wavefront. A page of text when illuminated with sunlight, for example, does this job perfectly. This would be called an amplitude-modulating reflective SLM. Modulation is a multiplicative effect, so an image encoded in the incoming wavefront will be pointwise multiplied by the image on the SLM. Modulators can also act on phase and polarisation, and can be transmissive rather than reflective. They include photographic film, and electro-optic, magneto-optic, and acousto-optic devices [57,4,48,29,35]. One class of note are the optically-addressed SLMs, in which, typically, a 2D light pattern falling on a photosensitive layer on one side of the SLM spatially varies (with an identical pattern) the reflective properties of the other side of the SLM. A beam splitter then allows one to read out this spatially-varying reflectance pattern. The liquid-crystal light valve [45] is one instance of this class. Other classes of SLMs such as liquid-crystal display panels and acousto-optic modulators allow one to dynamically alter the pattern using electronics. It is possible for a single device (such as an electronically programmed array of individual sources) to act as both source and modulator.

3.3. *Detectors and Nature's square law*

Optical signals can be regarded as having both an amplitude and phase. However, detectors will measure only the square of the amplitude of the signal (referred to as its intensity). This phenomenon is known as Nature's detector square law and applies to detectors from photographic film to digital cameras to the human eye. Detectors that obey this law are referred to as square-law detectors. This law is evident in many physical theories of light. In quantum theory, the measurement of a complex probability function is formalised as a projection onto the set of real numbers through a squaring operation. Square-law detectors need to be augmented with a interferometric or holographic arrangement to measure both amplitude and phase rather than intensity [13], or need to be used for multiple captures in different domains to heuristically infer the phase.

Since it squares the absolute value of a complex function, this square law can be used for some useful computation (for example, in the joint transform correlator [92]). Detectors most commonly used include high range point (single pixel) detectors such as photodiodes, highly sensitive photon detectors such as photomultiplier tubes, and 1D and 2D array detectors such as CCD- or CMOS-digital cameras. Intensity values outside the range of a detector (outside the lowest and highest intensities that the detector can record) are thresholded accordingly. The integration time of some detectors can be adjusted to sum all of the light intensity falling on them over a period of time. Other detectors can have quite large light sensitive areas and can sum all of the light intensity falling in a region of space.

3.4. *Other optical computing hardware*

Lenses can be used to effect high fan-in and fan-out interconnections, to rescale images linearly in either one or two dimensions, and for taking Fourier transforms. In fact, a coherent optical wavefront naturally evolves into its Fresnel transform, and subsequently into its Fourier transform at infinity, and the lens simply images those frequency components at a finite fixed distance.

A mirror changes the direction of the wavefront and simultaneously reflects it along some axis. A phase conjugate mirror [25] returns an image along the same path at which it approached the mirror.

Prisms can be used to for in-plane flipping (mirror image), in-plane rotations, and out-of-plane tilting. A prism or diffraction grating can be used to separate by wavelength the components of a multi-wavelength optical channel. For optical fiber communications applications, more practical (robust, economical, and scalable) alternatives exist to achieve the same effect [101].

Polarisation is an important property of wavefronts, in particular in coherent optical computing, and is the basis for how liquid crystal displays work. At each point, an optical wavefront has an independent polarisation value dependent on the angle, in the range $[0, 2\pi)$, of its electrical field. This can be independent of its successor (in the case of randomly polarised wavefronts), or dependent (as in the case of linear polarisation), or dependent and time varying (as in the case of circular or elliptical polarisation). Mathematically, a polarisation state, and the transition from one polarisation state to another, can be described using the Mueller calculus or the Jones calculus.

Photons have properties such as phase, polarisation, and quantum state that can be used for computation. For example, quantum computers using linear optical elements (such as mirrors, polarisers, beam splitters, and phase shifters) have been proposed and demonstrated [50].

4. Efficiencies in optical computing

Optical computing is an inherently multidisciplinary subject whose study routinely involves a spectrum of expertise that threads optical physics, materials science, optical engineering, electrical engineering, computer architecture, computer programming, and computer theory. Applying ideas from theoretical computer science, such as analysis of algorithms and computational complexity, enables us to place optical computing in a framework where we can try to answer a number of important questions. For example, which problems are optical computers suitable for solving? Also, how does the resource usage on optical computers compare with more standard (e.g. digital electronic) architectures? The physical principles behind some efficiencies in optical computing are outlined here.

4.1. *Fan-in efficiency*

Kirchoff's Law is well understood in analog electronics as a natural and constant-time means of summing the current at the intersection of an arbitrary number of wires [58]. In optics, the same thing is possible by directing several light beams towards a point detector with a linear response to incident light. Such an optical arrangement sums n nonnegative integers in $O(1)$ addition steps. On a model of a sequential digital electronic computer this would require $n - 1$ addition operations and even many typical (bounded fan-in) parallel models, with n or more processors, take $O(\log n)$ timesteps. Tasks that rely on scalar summation operations (such as matrix multiplication) would benefit greatly from an optical implementation of the scalar sum operation. Similarly, $O(1)$ multiplication and $O(1)$ convolution operations can be realised optically. Very recently, an optics-based digital signal processing platform has been marketed that claims digital processing speeds of tera (10^{12}) operations per second [53].

4.2. *Efficiency in interconnection complexity*

As optical pathways can cross in free space without measurable effect on the information in either channel, high interconnection densities are possible with optics [20].

Architectures with highly parallel many-to-many interconnections between parallel surfaces have already been proposed for common tasks such as sorting [7]. Currently, intra-chip, inter-chip, and inter-board connections are being investigated for manufacturing feasibility [59].

4.3. *Energy efficiency*

Electrical wires suffer from induced noise and heat, which increases dramatically whenever wires are made thinner or placed closer together, or whenever the data throughput is increased [59]. As a direct consequence of their resistance-free pathways and noise-reduced environments, optical systems have the potential to generate less waste heat and so consume less energy per computation step than electronic systems [14]. This has been demonstrated experimentally with general-purpose digital optical processors [38].

5. Optical models of computation and computational complexity

There has been a lot of effort put into designing optical computers to emulate conventional microprocessors (digital optical computing), and to image processing over continuous wavefronts (analog optical computing and pattern recognition). Numerous physical implementations of the latter class exist, and example applications include fast pattern recognition and matrix-vector algebra [35,90]. There have been much resources devoted to designs, implementations and algorithms for such optical information processing architectures (for example see [4,15,29,35,52,55,57,62,77,90,101,27] and their references).

However the computational complexity theory of optical computers (that is, finding lower and upper bounds on computational power in terms of known complexity classes) has received relatively little attention when compared with other nature-insired computing paradigms. Some authors have even complained about the lack of suitable models [29,55]. Many other areas of natural computing (e.g. [41,2,54,100,82,37,70,60,61]) have not suffered from this problem. Despite this, we review a number of algorithmically orientated results related to optical computing. We then go on to suggest classes of problems where optical computers might be usefully applied.

Reif and Tyagi [77] study two optically inspired models. One model is a 3D VLSI model augmented with a 2D discrete Fourier transform (DFT) primitive and parallel optical interconnections. The other is a DFT circuit with operations (multiplication, addition, comparison of two inputs, DFT) that compute over an ordered ring. Time complexity is defined for both models as number of (parallel) steps. For the first model, volume complexity is defined as the volume of the smallest convex box enclosing an instance of the model. For the DFT circuit, size is defined as the number of edges plus gates. Constant time, polynomial size/volume, algorithms for a number of problems are reported including matrix multiplication, sorting and string matching [77]. These interesting results are built upon the ability of their models to compute the 2D DFT in one step. The authors suggest that the algorithm designer and optical computing architecture communities should identify other primitive optical operations, besides the DFT, that might result in efficient parallel algorithms. Barakat and Reif [6], and Tyagi and Reif [76] have also shown lower bounds on the optical VLSI model.

Reif, Tygar and Yoshida [75] examined the computational complexity of ray tracing problems. In such problems we are concerned about the geometry of an optical system where diffraction is ignored and we wish to predict the position of light rays after passing through some system of mirrors and lenses. They gave undecidability and PSPACE hardness results, which gives an indication of the power of these systems as computational models.

Feitelson [29] gives a call to theoretical computer scientists to apply their knowledge and techniques to optical computing. He then goes on to generalise the concurrent read, concurrent write parallel random access machine, by augmenting it with two optically inspired operations. The first is the ability to write the same piece of data to many global memory locations at once. Secondly, if many values are concurrently written to a single memory location then a summation of those values is computed in a single timestep. Essentially Feitelson is using 'unbounded fan-in with summation' and 'unbounded fan-out'. His architecture mixes a well known discrete model with some optical capabilities.

A symbolic substitution model of computation has been proposed by Huang and Brenner, and a proof sketched of its universality [10]. This model of digital computation operates over discrete binary images and derives its efficiency by performing logical operations on each pixel in the image in parallel. It has the functionality to copy, invert, and shift laterally individual images, and OR and AND pairs of images. Suggested techniques for its optical implementation are outlined.

In computer science there are two famous classes of problems called P and NP [68]. P contains those problems that are solvable in polynomial time on a standard sequential computer, while NP is the class of problems that are solvable in polynomial time on a nondeterministic computer. NP contains P, and it is widely conjectured that they are not equal. A direct consequence of this conjecture is that there are (NP-hard) problems for which we strongly believe there is no polynomial time algorithm on a standard sequential computer.

It is known that it is possible to solve any NP (and even any PSPACE) problem in polynomial time on optical computers, albeit with exponential use of some other, space-like, resources [97,93,95]. These results were shown on the CSM, a general model of a wide range of optical computers. The lower bound results were shown by generating appropriate Boolean masks, of exponential size, and manipulating the masks via parallel multiplications and additions to simulate space bounded Turing machines in a time-efficient way. The model was designed on the one hand to be close to the realities of optical computing, and on the other hand to be relatively straightforward to analyse from the point of view of computational complexity theory (e.g. see Section 6). In Section 9.1 we discuss the computational abilities of this computational model.

Since these general results, there have been a number of specific examples of optical systems (with exponential resource usage) for NP-hard problems.

Shaked et al. [79–81] design an optical system for solving the NP-hard travelling salesman problem in polynomial time. Basically they use an optical matrix-vector multiplier to generate the (exponentially large) matrix of all possible tours, then they multiply this tour matrix by the vector of intercity weights, and finally the lowest value in the resulting vector corresponds to the shortest tour. Interestingly, they give both optical experiments and simulations. They note that solving travelling salesman problems (or Hamiltonian path problems) with more than 15 nodes is problematic. However they argue that for less nodes (e.g. 5) their system works in real-time, which is faster than digital-electronic architectures. Problems with such bounds on input size (i.e. constant) lie in the class NC^1, and moreover in AC^0. As argued below, perhaps this suggests that the optical computing community should be focusing on problems where optics excels over digital-electronic architectures, such as problems in P or NC, rather than NP-hard problems.

Dolev and Fitoussi [26] give optical algorithms that make use of (exponentially large) masks to solve a number of NP-hard problems. Oltean [66], and Haist and Osten [39], give architectures for Hamiltonian path, and travelling salesman problem, respectively, via light travelling through optical cables. As is to be expected, both suffer from exponential resource use. The paper by MacKenzie and Ramachandran [56] is an example of algorithmic work, and lower bounds, on dynamically reconfigurable optical networks.

5.1. *A possible way forward*

Nature-inspired systems that apparently solve NP-hard problems in polynomial time, while using an exponential amount of some other resource(s), have been around for many years. So the existence of massively parallel optical systems for NP-hard problems should not really suprise the reader.

One could argue that it is interesting to know the computational abilities, limitations, and resource trade-offs of such optical architectures, as well as to find particular (tractable or intractable) problems which are particularly suited to optical algorithms. However, "algorithms" that use exponential space-like resources (such as number of pixels, number of images, number of amplitude levels, etc.) are not realistic to implement for large input instances. What happens to highly parallel optical architectures if add the requirement that the amount of space-like resources are bounded in some reasonable way? We could, for example, stipulate that the optical machine use no more than a polynomially bounded amount of space-like resources. If the machine runs in polynomial time, then it is not difficult to see that it characterises P [99] (by characterise we mean that the model solves exactly those problems in P), for a wide range of reasonable parallel and sequential optical models. Many argue that the reason for using parallel architectures is to speed-up computations. Asking for an exponential speed-up motivates the complexity class NC. The class NC can be thought of as those problems in P that can be solved exponentially faster on parallel computers than on sequential computers. NC is contained in P and it is an major open question whether this containment is strict: it is widely conjectured that this is indeed the case [36].

How does this relate to optics? It turns out that a wide range of optical computers that run for at most polylogarithmic time, and use at most polynomial space-like resources, solve exactly NC [97,93,95] (this can be shown to be a corollary of the PSPACE characterisation cited earlier in Section 5). In effect this means that we have an algorithmic way (in other words, a compiler) to convert existing NC algorithms into optical algorithms that use similar amounts of resources. There is scope for further work here, on the CSM in particular, in order to find exact characterisations, or as close as possible for NC^k for given k. On a technical note, NC can be defined as $\cup_{k=0}^{\infty}\mathrm{NC}^k$, where NC^k is the class of problems solvable on a PRAM that runs for $O(\log n)^k$ time and uses polynomial processors/space, in input length n. Equivalently NC^k can be defined as those problems solvable by circuits of $O(\log n)^k$ depth (parallel time), and polynomial size. From the practical side of things, perhaps we can use these kinds of results to find problems within

NC, where optical architectures can be shown to excel. Obvious examples for which this is already known are matrix-vector multiplication (which lies in NC^2), or Boolean matrix multiplication (which is in NC^1). Another example is the NC^1 unordered search problem [99,98]. Another closely related idea is to exploit the potential unbounded fan-in of optics to compute problems in the AC, and TC, (parallel) circuit classes. These are defined similarly to NC circuits except we allow unbounded fan-in gates, and threshold gates, respectively. The results in the above mentioned paper of Reif and Tyagi [77], and Caulfield's observation on the benefits of unbounded fan-in [16], can be interpreted as exploiting this important and efficient aspect of optics.

6. Continuous space machine (CSM)

For the remainder of this paper we focus on an optical model of computation called the CSM. The model was originally proposed by Naughton [63,64]. The CSM is inspired by analog Fourier optical computing architectures, specifically pattern recognition and matrix algebra processors [35,62]. For example, these architectures have the ability to do unit time Fourier transformation using coherent (laser) light and lenses. The CSM computes in discrete timesteps over a number of two-dimensional images of fixed size and arbitrary spatial resolution. The data and program are stored as images. The (constant time) operations on images include Fourier transformation, multiplication, addition, thresholding, copying and scaling. The model is designed to capture much of the important features of optical computers, while at the same time be amenable to analysis from a computer theory point of view. Towards these goals we give an overview of how the model relates to optics as well as giving a number of computational complexity results for the model.

Section 6.1 begins by defining the model. We analyse the model in terms of seven complexity measures inspired by real-world resources, these are described in Section 6.2. In Section 6.3 we discuss possible optical implementations for the model. We then go on to give example algorithms and datastructures in Section 7. The CSM definition is rather general, and so in Section 8 we define a more restricted model called the $\mathcal{C}_2$-CSM. Compared to the CSM, the $\mathcal{C}_2$-CSM is somewhat closer to optical computing as it happens in the laboratory. Finally, in Section 9 we show the power and limitations of optical computing, as embodied by the $\mathcal{C}_2$-CSM, in terms computational complexity theory. Optical information processing is a highly parallel form of computing and we make this intuition more concrete by relating the $\mathcal{C}_2$-CSM to parallel complexity theory by characterising the parallel complexity class NC. For example, this shows the kind of worst case resource usage one would expect when applying CSM algorithms to problems that are known to be suited to parallel solutions.

6.1. *CSM definition*

We begin this section by describing the CSM model in its most general setting, this brief overview is not intended to be complete and more details are to be found in [93].

A complex-valued image (or simply, image) is a function $f : [0,1) \times [0,1) \to \mathbb{C}$, where $[0,1)$ is the half-open real unit interval. We let $\mathcal{I}$ denote the set of complex-valued images. Let $\mathbb{N}^+ = \{1,2,3,\ldots\}$, $\mathbb{N} = \mathbb{N}^+ \cup \{0\}$, and for a given CSM M let $\mathcal{N}$ be a countable set of images that encode M's addresses. An address is an element of $\mathbb{N} \times \mathbb{N}$. Additionally, for a given M there is an *address encoding function* $\mathfrak{E} : \mathbb{N} \rightarrow \mathcal{N}$ such that $\mathfrak{E}$ is Turing machine decidable, under some *reasonable* representation of images as words.

Definition 1 (CSM) *A CSM is a quintuple* $M = (\mathfrak{E}, L, I, P, O)$, *where*
$\mathfrak{E} : \mathbb{N} \rightarrow \mathcal{N}$ *is the address encoding function,*
$L = ((s_\xi, s_\eta), (a_\xi, a_\eta), (b_\xi, b_\eta))$ *are the addresses: sta, a and b, where* $a \neq b$,
I *and* O *are finite sets of input and output addresses, respectively,*
$P = \{(\zeta_1, p_{1_\xi}, p_{1_\eta}), \ldots, (\zeta_r, p_{r_\xi}, p_{r_\eta})\}$ *are the r programming symbols* ζ_j *and their addresses* (p_{j_ξ}, p_{j_η}) *where* $\zeta_j \in (\{h, v, *, \cdot, +, \rho, st, ld, br, hlt\} \cup \mathcal{N}) \subset \mathcal{I}$.
Each address is an element from $\{0, \ldots, \Xi - 1\} \times \{0, \ldots, ⅄ - 1\}$ *where* $\Xi, ⅄ \in \mathbb{N}^+$.

Addresses whose contents are not specified by P in a CSM definition are assumed to contain the constant image $f(x, y) = 0$. We interpret this definition to mean that M is (initially) defined on a grid of images bounded by the constants Ξ and ⅄, in the horizontal and vertical directions respectively. The grid of images may grow in size as the computation progresses. In our grid notation the first and second elements of an address tuple refer to the horizontal and vertical axes of the grid respectively, and image $(0, 0)$ is located at the lower left-hand corner of the grid. The images have the same orientation as the grid. For example the value $f(0, 0)$ is located at the lower left-hand corner of the image f.

In Definition 1 the tuple P specifies the CSM program using programming symbol images ζ_j that are from the (low-level) CSM programing language [93,98]. We refrain from giving a description of this programming language and instead describe a less cumbersome high-level language [93]. Figure 1 gives the basic instructions of this high-level language. The copy instruction is illustrated in Figure 2. There are also **if/else** and **while** control flow instructions with conditional equality tests of the form $(f_\psi == f_\phi)$ where f_ψ and f_ϕ are *binary symbol images* (see Figures 3(a) and 3(b)).

Address *sta* is the start location for the program so the programmer should write the first program instruction at *sta*. Addresses *a* and *b* define special images that are frequently used by some program instructions. The function $\mathfrak{E}$ is specified by the programmer and is used to map addresses to image pairs. This enables the programmer to choose her own address encoding scheme. We typically don't want $\mathfrak{E}$ to hide complicated behaviour thus the computational power of this function should be somewhat restricted. For example we put such a restriction on $\mathfrak{E}$ in Definition 7. At any given timestep, a configuration is defined in a straightforward way as a tuple $\langle c, e \rangle$ where c is an address called the control and e represents the grid contents.

6.2. *Complexity measures*

In this section we define a number of CSM complexity measures. As is standard, all resource bounding functions map from $\mathbb{N}$ into $\mathbb{N}$ and are assumed to have the usual properties [5]. We begin by defining CSM TIME complexity in a manner that is standard among parallel models of computation.

Definition 2 *The* TIME *complexity of a CSM M is the number of configurations in the computation sequence of M, beginning with the initial configuration and ending with the first final configuration.*

$\mathrm{h}(i_1;i_2)$: replace image at i_2 with horizontal 1D Fourier transform of i_1.

$\mathrm{v}(i_1;i_2)$: replace image at i_2 with vertical 1D Fourier transform of image at i_1.

$*(i_1;i_2)$: replace image at i_2 with the complex conjugate of image at i_1.

$\cdot\,(i_1,i_2;i_3)$: pointwise multiply the two images at i_1 and i_2. Store result at i_3.

$+(i_1,i_2;i_3)$: pointwise addition of the two images at i_1 and i_2. Store result at i_3.

$\rho(i_1,z_{\mathrm{l}},z_{\mathrm{u}};i_2)$: filter the image at i_1 by amplitude using z_{l} and z_{u} as lower and upper amplitude threshold images, respectively. Place result at i_2.

$[\xi_1',\xi_2',\eta_1',\eta_2'] \leftarrow [\xi_1,\xi_2,\eta_1,\eta_2]$: copy the rectangle of images whose bottom left-hand address is (ξ_1,η_1) and whose top right-hand address is (ξ_2,η_2) to the rectangle of images whose bottom left-hand address is (ξ_1',η_1') and whose top right-hand address is (ξ_2',η_2'). See illustration in Figure 2.

Fig. 1. CSM high-level programming language instructions. In these instructions $i, z_{\mathrm{l}}, z_{\mathrm{u}} \in \mathbb{N}\times\mathbb{N}$ are image addresses and $\xi,\eta \in \mathbb{N}$. The control flow instructions are described in the main text.

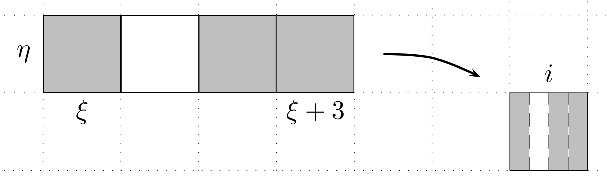


Fig. 2. Illustration of the instruction $i \leftarrow [\xi,\xi+3,\eta,\eta]$ that copies four images to a single image that is denoted i.

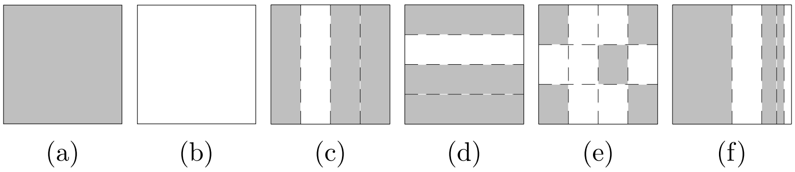


Fig. 3. Representing binary data. The shaded areas denote value 1 and the white areas denote value 0. (a) Binary symbol image representation of 1 and (b) of 0, (c) list (or row) image representation of the word 1011, (d) column image representation of 1011, (e) 3×4 matrix image, (f) binary stack image representation of 1101. Dashed lines are for illustration purposes only.

The first of our six space-like resources is called GRID.

Definition 3 *The* GRID *complexity of a CSM M is the minimum number of images, arranged in a rectangular grid, for M to compute correctly on all inputs.*

Let $S : \mathcal{I}\times(\mathbb{N}\times\mathbb{N}) \to \mathcal{I}$, where $S(f(x,y),(\Phi,\Psi))$ is a raster image, with $\Phi\Psi$ constant-valued pixels arranged in Φ columns and Ψ rows, that approximates $f(x,y)$. If we choose a reasonable and realistic S then the details of S are not important.

Definition 4 *The* SPATIALRES *complexity of a CSM M is the minimum $\Phi\Psi$ such that if each image $f(x,y)$ in the computation of M is replaced with $S(f(x,y),(\Phi,\Psi))$ then M computes correctly on all inputs.*

One can think of spatialRes as a measure of the number of pixels needed during a computation. In optical image processing terms, and given the fixed size of our images, spatialRes corresponds to the space-bandwidth product of a detector or SLM.

Definition 5 *The* dyRange *complexity of a CSM M is the ceiling of the maximum of all the amplitude values stored in all of M's images during M's computation.*

In optical processing terms dyRange corresponds to the dynamic range of a signal.

We also use complexity measures called amplRes, phaseRes and freq [93,98]. Roughly speaking, the amplRes of a CSM M is the number of discrete, evenly spaced, amplitude values per unit amplitude of the complex numbers in M's images. For example, we would need amplRes of 3 to directly store values from the set $\{0, \pm 1/3, \pm 2/3, \pm 1, \pm 4/3, \ldots\}$ as complex values in an image. Thus amplRes corresponds to the amplitude quantisation of a signal. The phaseRes of M is the total number (per 2π) of discrete evenly spaced phase values in M's images, and so phaseRes corresponds to the phase quantisation of a signal. For example, we would need phaseRes of 3 to directly store values from the set $\{\mathrm{e}^{\mathrm{i}x} \mid x \in \{0, 2/3\pi, 4/3\pi\}\}$ as complex values in an image. Finally, the freq complexity of a CSM M is the minimum optical frequency necessary for M to compute correctly, this concept is explained further in [98].

Often we wish to make analogies between space on some well-known model and CSM 'space-like' resources. Thus we define the following convenient term.

Definition 6 *The* space *complexity of a CSM M is the product of the* grid, spatialRes, dyRange, amplRes, phaseRes *and* freq *complexities of M.*

6.3. *Optical realisation*

In this section, we outline how some of the elementary operations of the CSM could be carried out physically. We do not intend to specify the definitive realisation of any of the operations, but simply convince the reader that the model's operations have physical interpretations. Furthermore, although we concentrate on implementations employing visible light (optical frequencies detectable to the human eye) the CSM definition does not preclude employing other portion(s) of the electromagnetic spectrum.

A complex-valued image could be represented physically by a spatially coherent optical wavefront. Spatially coherent illumination (light of a single wavelength and emitted with the same phase angle) can be produced by a laser. A SLM could be used to encode the image onto the expanded and collimated laser beam. One could write to a SLM offline (expose photographic film, or laser print or relief etch a transparency) or online (in the case of a liquid-crystal display [102,62,91] or holographic material [24,74]). The functions h and v could be effected using two convex cylindrical lenses, oriented horizontally and vertically, respectively [90,35,62,34].

A coherent optical wavefront will naturally evolve into its own Fourier spectrum as it propagates to infinity. What we do with a convex lens is simply image, at a finite distance, this spectrum. This finite distance is called the focal length of the lens. The constant θ used in the definitions of h and v could be effected using Fourier spectrum size reduction techniques [90,35] such as varying the focal length of the lens, varying the separation of the lens and SLM, employing cascaded Fourier transformation, increasing the dimensions/reducing the spatial resolution of the SLM, or using light with a shorter wavelength.

The function $*$ could be implemented using a phase conjugate mirror [25]. The function $\cdot$ could be realised by placing a SLM encoding an image f in the path of a wavefront encoding another image g [90,35,89]. The wavefront immediately behind the SLM would then be $\cdot(f, g)$. The function $+$ describes the superposition of two optical wavefronts. This could be achieved using a 50:50 beam splitter [90,25,92]. The function ρ could be implemented using an electronic camera or a liquid-crystal light valve [91]. The parameters z_l and z_u would then be physical characteristics of the particular camera/light valve used. z_l corresponds to the minimum intensity value that the device responds to, known as the dark current signal, and z_u corresponds to the maximum intensity (the saturation level).

A note will be made about the possibility of automating these operations. If suitable SLMs can be prepared with the appropriate 2D pattern(s), each of the operations h, v, $*$, $\cdot$, and $+$ could be effected autonomously and without user intervention using appropriately positioned lenses and free space propagation. The time to effect these operations would be the sum of the flight time of the image (distance divided by velocity of light) and the response time of the analog 2D detector; both of which are constants independent of the size or resolution of the images if an appropriate 2D detector is chosen. Examples of appropriate detectors would be holographic material [24,74] and a liquid-crystal light valve with a continuous (not pixellated) area [91]. Since these analog detectors are also optically-addressed SLMs, we can very easily arrange for the output of one function to act as the input to another, again in constant time independent of the size or resolution of the image. A set of angled mirrors will allow the optical image to be fed back to the first SLM in the sequence, also in constant time. It is not known, however, if ρ can be carried out completely autonomously for arbitrary parameters. Setting arbitrary parameters might fundamentally require offline user intervention (adjusting the gain of the camera, and so on), but at least for a small range of values this can be simulated online using a pair of liquid-crystal intensity filters.

We have outlined some optics principles that could be employed to implement the operations of the model. The simplicity of the implementations hides some imperfections in our suggested realisations. For example, the implementation of the $+$ operation outlined above results in an output image that has been unnecessarily multiplied by the constant factor 0.5 due to the operation of the beam splitter. Also, in our suggested technique, the output of the ρ function is squared unnecessarily. However, each of these effects can be compensated for with a more elaborate optical setup and/or at the algorithm design stage.

A more important issue concerns the quantum nature of light. According to our current understanding, light exists as individual packets called photons. As such, in order to physically realise the CSM one would have to modify it such that images would have discrete, instead of continuous, amplitudes. The atomic operations outlined above, in particular the Fourier transform, are not affected by the restriction to quantised amplitudes, as the many experiments with electron interference patterns indicate. We would still assume, however, that in the physical world space is continuous.

A final issue concerns how a theoretically infinite Fourier spectrum could be represented by an image (or encoded by a SLM) of finite extent. This difficulty is addressed with the FREQ complexity measure [98].

7. Example CSM datastructures and algorithms

In this section we give some example data representations. We then to go on to give an example CSM algorithm that efficiently squares a binary matrix.

7.1. *Representing data as images*

There are many ways to represent data as images and interesting new algorithms sometimes depend on a new data representation. Data representations should be in some sense reasonable, for example it is unreasonable that the input to an algorithm could (non-uniformly) encode solutions to NP-hard or even undecidable problems. From Section 8.1, the CSM address encoding function gives the programmer room to be creative, so long as the representation is logspace computable (assuming a reasonable representation of images as words).

Here we mention some data representations that are commonly used. Figures 3(a) and 3(b) are the binary symbol image representations of 1 and 0 respectively. These images have an everywhere constant value of 1 and 0 respectively, and both have SPATIALRES of 1. The row and column image representations of the word 1011 are respectively given in Figures 3(c) and 3(d). These row and column images both have SPATIALRES of 4. In the matrix image representation in Figure 3(e), the first matrix element is represented at the top left corner and elements are ordered in the usual matrix way. This 3×4 matrix image has SPATIALRES of 12. Finally the binary stack image representation, which has exponential SPATIALRES of 16, is given in Figure 3(f).

Figure 2 shows how we might form a list image by copying four images to one in a single timestep. All of the above mentioned images have DYRANGE, AMPLRES and PHASERES of 1.

Another useful representation is where the value of a pixel directly encodes a number, in this case DYRANGE becomes crucial. We can also encode values as phase values, and naturally PHASERES becomes a useful measure of the resources needed to store such values.

7.2. *A matrix squaring algorithm*

Here we give an example CSM algorithm (taken from [95]) that makes use of the data representations described above. The algorithm squares a $n \times n$ matrix in $O(\log n)$ TIME and $O(n^3)$ SPATIALRES (number of pixels), while all other CSM resources are constant.

Lemma 1 *Let n be a power of 2 and let A be a $n \times n$ binary matrix. The matrix A^2 is computed by a $\mathcal{C}_2$-CSM, using the matrix image representation, in* TIME $O(\log n)$*,* SPATIALRES $O(n^3)$*,* GRID $O(1)$*,* DYRANGE $O(1)$*,* AMPLRES 1 *and* PHASERES 1*.*

Proof. *(Sketch)* In this proof the matrix and its matrix image representation (see Figure 3(e)) are both denoted A. We begin with some precomputation, then one parallel pointwise multiplication step, followed by $\log n$ additions to complete the algorithm.

We generate the matrix image A_1 that consists of n vertically juxtaposed copies of A. This is computed by placing one copy of A above the other, scaling to one image, and repeating to give a total of $\log n$ iterations. The image A_1 is constructed in TIME $O(\log n)$, GRID $O(1)$ and SPATIALRES $O(n^3)$.

Next we transpose A to the column image A_2. The first n elements of A_2 are row 1 of A, the second n elements of A_2 are row 2 of A, etc. This is computed in TIME $O(\log n)$, GRID $O(1)$ and SPATIALRES $O(n^2)$ as follows.

Let $A' = A$ and $i = n$. We horizontally split A' into a left image A'_L and a right image A'_R. Then A'_L is pointwise multiplied (or masked) by the column image that represents $(10)^i$, in TIME $O(1)$. Similarly A'_R is pointwise multiplied (or masked) by the column image that represents $(01)^i$. The masked images are added. The resulting image has half the number of columns as A' and double the number of rows, and for example: row 1 consists of the first half of the elements of row 1 of A' and row 2 consists of the latter half of the elements of row 1 of A'. We call the result A' and we double the value of i. We repeat the process to give a total of $\log n$ iterations. After these iterations the resulting column image is denoted A_2.

We pointwise multiply A_1 and A_2 to give A_3 in TIME $O(1)$, GRID $O(1)$ and SPATIALRES $O(n^3)$.

To facilitate a straightforward addition we first transpose A_3 in the following way: A_3 is vertically split into a bottom and a top image, the top image is placed to the left of the bottom and the two are scaled to a single image, this splitting and scaling is repeated to give a total of $\log n$ iterations and we call the result A_4. Then to perform the addition, we vertically split A_4 into a bottom and a top image. The top image is pointwise added to the bottom image and the result is thresholded between 0 and 1. This splitting, adding and thresholding is repeated a total of $\log n$ iterations to create A_5. We 'reverse' the transposition that created A_4: image A_5 is horizontally split into a left and a right image, the left image is placed above the right and the two are scaled to a single image, this splitting and scaling is repeated a total of $\log n$ iterations to give A^2. □

The algorithm highlights a few points of interest about the CSM. The CSM has quite a number of space-like resources, and it is possible to have trade-offs between them. For example in the algorithm above, if we allow GRID to increase from $O(1)$ to $O(n)$ then the SPATIALRES can be reduced from $O(n^3)$ to $O(n^2)$. In terms of optical architectures modelled by the CSM this phenomenon could be potentially very useful as certain resources may well be more economically viable than others. The algorithm is used in the proof that that polynomial TIME CSMs (and $\mathcal{C}_2$-CSMs, see below) compute problems that are in the PSPACE class of languages. PSPACE includes the famous NP class. Such computational complexity results are discussed further in Section 9 below.

There are a number of existing CSM algorithms, for these we point the reader to the literature [63–65,93,95,97,98].

8. $\mathcal{C}_2$-CSM

In this section we define the $\mathcal{C}_2$-CSM. One of the motivations for this model is the need to put reasonable upper bounds on the power of reasonable optical computers. As we've shown elsewhere [96], it turns out that CSMs can very quickly use massive amounts of resources, and the $\mathcal{C}_2$-CSM definition is an attempt to define a more reasonable model, especially towards the goal of providing useful upper bounds on its power.

8.1. $\mathcal{C}_2$-CSM

Motivated by a desire to apply standard complexity theory tools to the model, we defined [93,96] the $\mathcal{C}_2$-CSM, a restricted and more realistic class of CSM.

Definition 7 ($\mathcal{C}_2$-CSM) *A $\mathcal{C}_2$-CSM is a CSM whose computation* TIME *is defined for $t \in \{1, 2, \ldots, T(n)\}$ and has the following restrictions:*

- *For all* TIME *t both* AMPLRES *and* PHASERES *have constant value of 2.*
- *For all* TIME *t each of* GRID, SPATIALRES *and* DYRANGE *is $2^{O(t)}$ and* SPACE *is redefined to be the product of all complexity measures except* TIME *and* FREQ.
- *Operations h and v compute the discrete Fourier transform in the horizontal and vertical directions respectively.*
- *Given some* reasonable *binary word representation of the set of addresses $\mathcal{N}$, the address encoding function $\mathfrak{E} : \mathbb{N} \rightarrow \mathcal{N}$ is decidable by a logspace Turing machine.*

Let us discuss these restrictions. The restrictions on AMPLRES and PHASERES imply that $\mathcal{C}_2$-CSM images are of the form $f : [0, 1) \times [0, 1) \rightarrow \{0, \pm\frac{1}{2}, \pm 1, \pm\frac{3}{2}, \ldots\}$. We have replaced the Fourier transform with the discrete Fourier transform [9], this essentially means that FREQ is now solely dependent on SPATIALRES; hence FREQ is not an interesting complexity measure for $\mathcal{C}_2$-CSMs and we do not analyse $\mathcal{C}_2$-CSMs in terms of FREQ complexity [93,96]. Restricting the growth of SPACE is not unique to our model, such restrictions are to be found elsewhere [33,69,73].

In Section 6.1 we stated that the address encoding function $\mathfrak{E}$ should be Turing machine decidable, here we strengthen this condition. At first glance sequential logspace computability may perhaps seem like a strong restriction, but in fact it is quite weak. From an optical implementation point of view it should be the case that $\mathfrak{E}$ is not complicated, otherwise we cannot assume fast addressing. Other (sequential/parallel) models usually have a very restricted 'addressing function': in most cases it is simply the identity function on $\mathbb{N}$. Without an explicit or implicit restriction on the computational complexity of $\mathfrak{E}$, finding non-trivial upper bounds on the power of $\mathcal{C}_2$-CSMs is impossible as $\mathfrak{E}$ could encode an arbitrarily complex Turing machine. As a weaker restriction we could give a specific $\mathfrak{E}$. However, this restricts the generality of the model and prohibits the programmer from developing novel, reasonable, addressing schemes.

9. Optical computing and computational complexity

As we saw in Section 5, there are number of optical algorithms that use the inherent parallelism of optics to provide fast solutions to certain problems. An alternative approach is to ask the following question: How does a given optical model relate to standard sequential and parallel models? Establishing a relationship with computational complexity theory, by describing both upper and lower bounds on the model, gives immediate access to a large collection of useful algorithms and proof techniques.

The parallel computation thesis [31,23,49,87,69] states that parallel time (polynomially) corresponds to sequential space, for reasonable parallel and sequential models. An example would be the fact that the class of problems solvable in polynomial space on a number of parallel models is equivalent to PSPACE, the class of problems solvable on Turing machines that use at most polynomial space [40,8,30,32,22,33,88,85,83,3,84].

Of course the thesis can never be proved, it relates the intuitive notion of reasonable parallelism to the precise notion of a Turing machine. When results of this type were first shown researchers were suitably impressed; their parallel models truly had great power. For example if model M verifies the thesis then M decides PSPACE (including NP) languages in polynomial time. However there is another side to this coin. It is straightforward to verify that given our current best algorithms, M will use at least a superpolynomial amount of some other resource (like space or number of processors) to decide a PSPACE-complete or NP-complete language. Since the composition of polynomials is itself a polynomial, it follows that if we restrict the parallel computer to use at most polynomial time and polynomial other resources, then it can at most solve problems in P.

Nevertheless, asking if M verifies the thesis is an important question. Certain problems, such as those in the class NC, are efficiently parallelisable. NC can be defined as the class of problems that are solvable in polylogarithmic time on a parallel computer that uses a polynomial amount of hardware. So one can think of NC as those problems in P which are solved exponentially faster on parallel computation thesis models than on sequential models. If M verifies the thesis then it may be useful to apply M to these problems. We also know that if M verifies the thesis then there are (P-complete) problems for which it is widely believed that we will not find exponential speed-up using M.

9.1. $\mathcal{C}_2$-CSM *and parallel complexity theory*

Here we summarise some characterisations of the computing power of optical computers. Such characterisations enable the algorithm designer to know what kinds of problems are solvable with resource bounded optical algorithms.

Theorem 2 below gives lower bounds on the computational power of the $\mathcal{C}_2$-CSM by showing that it is at least as powerful as models that verify the parallel computation thesis.

Theorem 2 ([95,97]) $\mathrm{NSPACE}(S(n)) \subseteq \mathcal{C}_2\text{-CSM–TIME}(O(S^2(n)))$

In particular, polynomial TIME $\mathcal{C}_2$-CSMs accept the PSPACE languages. PSPACE is the class of problems solvable by Turing machines that use polynomial space, which includes the famous class NP, and so NP-complete problems can be solved by $\mathcal{C}_2$-CSMs in polynomial TIME. However, any $\mathcal{C}_2$-CSM algorithm that we could presently write to solve PSPACE or NP problems would require exponential SPACE.

Theorem 2 is established by giving a $\mathcal{C}_2$-CSM algorithm that efficiently generates, and squares, the transition matrix of a $S(n) = \Omega(\log n)$ space bounded Turing machine. This transition matrix represents all possible computations of the Turing machine and is of size $O(2^S) \times O(2^S)$. The matrix squaring part was already given as an example (Lemma 1), and the remainder of the algorithm is given in [95]. The algorithm uses SPACE that is cubic in one of the matrix dimensions. In particular the algorithm uses cubic SPATIALRES, $O(2^{3S})$, and all other space-like resources are constant. This theorem improves upon the time overhead of a previous similar result [93,97] that was established via $\mathcal{C}_2$-CSM simulation of the vector machines [72,73] of Pratt, Rabin, and Stockmeyer.

From the resource usage point of view, it is interesting to see that the older of these two algorithms uses GRID, DYRANGE, and SPATIALRES that are each $O(2^S)$, while the newer algorithm shows that if we allow more SPATIALRES we can in fact use only constant GRID and DYRANGE. It would be interesting to find other such resource trade-offs within the model.

Since NP is contained in PSPACE, Theorem 2, and the corresponding earlier results in [93,97], show that this optical model solves NP-complete problems in polynomial TIME. As described in Section 5, this has also been shown experimentally, for example Shaked et al. [80] have recently given a polynomial time, exponential space, optical algorithm to solve the NP-complete travelling salesperson problem. Their optical setup can be implemented on the CSM.

The other of the two inclusions that are necessary in order to verify the parallel computation thesis have also been shown: $\mathcal{C}_2$-CSMs computing in TIME $T(n)$ are no more powerful than $T^{O(1)}(n)$ space bounded deterministic Turing machines. More precisely, we have:

Theorem 3 ([93,94]) $\mathcal{C}_2$-CSM-TIME$(T(n)) \subseteq$ DSPACE$(O(T^2(n)))$

This result gives an upper bound on the power of $\mathcal{C}_2$-CSMs and was established via $\mathcal{C}_2$-CSM simulation by logspace uniform circuits of size and depth polynomial in SPACE and TIME respectively [94].

These computational complexity results for the $\mathcal{C}_2$-CSM have shown that the model is capable of parallel processing in much the same way as models that verify the parallel computation thesis (and models that are known to characterise the parallel class NC). These results strongly depend on their use of non-constant SPATIALRES. The algorithms exploit the ability of optical computers, and the CSM in particular, to operate on large numbers of pixels in parallel. But what happens when we do not have arbitrary numbers of pixels? If allow images to have only a constant number of pixels then we need to find new CSM algorithms. It turns out that such machines also characterise PSPACE in polynomial TIME.

Theorem 4 ([99]) PSPACE *is characterised by* $\mathcal{C}_2$-CSM*s that are restricted to use polynomial* TIME $T = O(n^k)$*,* SPATIALRES $O(1)$*,* GRID $O(1)$*, and generalised to use* AMPLRES $O(2^{2^T})$*,* DYRANGE $O(2^{2^T})$*.*

So by treating images as registers and generating exponentially large, and exponentially small, values we can solve seemingly intractable problems. Of course this kind of CSM is quite unrealistic from the point of view of optical implementations. In particular, accurate multiplication of such values is difficult to implement in optics [99].

To restrict the model we could replace arbitrary multiplication, by multiplication by constants, which can be easily simulated by a constant number of additions. If we disallow multiplication in this way, we characterise P.

Theorem 5 ([99]) $\mathcal{C}_2$-CSM*s without multiplication, that compute in polynomial* TIME*, polynomial* GRID $O(n^k)$*, and* SPATIALRES $O(1)$*, characterise P.*

We can swap the roles of GRID and SPATIALRES, and still obtain a P characterisation:

Theorem 6 ([99]) *CSMs without multiplication, that compute in polynomial* TIME*, polynomial* SPATIALRES $O(n^k)$*, and* GRID $O(1)$*, characterise P.*

Theorems 5 and 6 give conditions under which our optical model essentially looses its parallel abilities and acts like a standard sequential Turing machine.

Via the proofs of Theorems 2 and 3 we can show that $\mathcal{C}_2$-CSMs that simultaneously use polynomial SPACE and polylogarithmic TIME solve exactly those problems in the class NC.

Corollary 7 $\mathcal{C}_2\text{-CSM-SPACE}, \text{TIME}(n^{O(1)}, \log^{O(1)} n) = \text{NC}$

Problems in NC highlight the power of parallelism, as these problems can be solved exponentially faster on a polynomial amount of parallel resources than on polynomial time sequential machines. As further work in this area one could try to find alternate characterisations of NC in terms of the $\mathcal{C}_2$-CSM. In particular, one could try to find further interesting trade-offs between the various space-like resources of the model. In the real world this would correspond to computing over various different optical resources. As discussed in Section 5 it would be interesting for optical algorithm designers to try to design (implementable) optical algorithms for NC problems in an effort to find problems that are well-suited to optical solutions.

References

[1] M. A. G. Abushagur and H. J. Caulfield. Speed and convergence of bimodal optical computers. *Optical Engineering*, 26(1):22–27, Jan. 1987.

[2] L. M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, Nov. 1994.

[3] A. Alhazov and M. de Jesús Pérez-Jiménez. Uniform solution to QSAT using polarizationless active membranes. In J. Durand-Lose and M. Margenstern, editors, *Machines, Computations and Universality (MCU)*, volume 4664 of *LNCS*, pages 122–133, Orléans, France, Sept. 2007. Springer.

[4] H. H. Arsenault and Y. Sheng. *An Introduction to Optics in Computers*, volume TT8 of *Tutorial Texts in Optical Engineering.* SPIE Press, Bellingham, Washington, 1992.

[5] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural complexity, vols I and II.* EATCS Monographs on Theoretical Computer Science. Springer, Berlin, 1988.

[6] R. Barakat and J. H. Reif. Lower bounds on the computational efficiency of optical computing systems. *Applied Optics*, 26(6):1015–1018, Mar. 1987.

[7] F. R. Beyette Jr., P. A. Mitkas, S. A. Feld, and C. W. Wilmsen. Bitonic sorting using an optoelectronic recirculating architecture. *Applied Optics*, 33(35):8164–8172, Dec. 1994.

[8] A. Borodin. On relating time and space to size and depth. *SIAM Journal on Computing*, 6(4):733–744, Dec. 1977.

[9] R. N. Bracewell. *The Fourier transform and its applications.* Electrical and electronic engineering series. McGraw-Hill, second edition, 1978.

[10] K.-H. Brenner, A. Huang, and N. Streibl. Digital optical computing with symbolic substitution. *Applied Optics*, 25(18):3054–3060, Sept. 1986.

[11] D. P. Casasent and G. P. House. Comparison of coherent and noncoherent optical correlators. In *Optical Pattern Recognition V*, Proceedings of SPIE vol. 2237, pages 170–178, Apr. 1994.

[12] D. P. Casasent and D. Psaltis. Position, rotation, and scale invariant optical correlation. *Applied Optics*, 15(7):1795–1799, 1976.

[13] H. J. Caulfield, editor. *Handbook of Optical Holography.* Academic Press, New York, 1979.

[14] H. J. Caulfield. The energetic advantage of analog over digital computing. In *OSA Optical Computing Technical Digest Series*, volume 9, pages 180–183, 1989.

[15] H. J. Caulfield. Space-time complexity in optical computing. In B. Javidi, editor, *Optical information-processing systems and architectures II*, volume 1347, pages 566–572. SPIE, July 1990.

[16] H. J. Caulfield. Space-time complexity in optical computing. *Multidimensional Systems and Signal Processing*, 2(4):373–378, Nov. 1991. Special issue on optical signal processing.

[17] H. J. Caulfield and M. A. G. Abushagur. Hybrid analog-digital algebra processors. In *Optical and Hybrid Computing II*, Proceedings of SPIE vol. 634, pages 86–95, Orlando, Florida, Apr. 1986.

[18] H. J. Caulfield and R. Haimes. Generalized matched filtering. *Applied Optics*, 19(2):181–183, Jan. 1980.

[19] H. J. Caulfield, S. Horvitz, and W. A. V. Winkle. Introduction to the special issue on optical computing. *Proceedings of the IEEE*, 65(1):4–5, Jan. 1977.

[20] H. J. Caulfield, J. M. Kinser, and S. K. Rogers. Optical neural networks. *Proceedings of the IEEE*, 77:1573–1582, 1989.

[21] H. J. Caulfield, W. T. Rhodes, M. J. Foster, and S. Horvitz. Optical implementation of systolic array processing. *Optics Communications*, 40:86–90, 1981.

[22] A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, Jan. 1981.

[23] A. K. Chandra and L. J. Stockmeyer. Alternation. In *17th annual symposium on Foundations of Computer Science*, pages 98–108, Houston, Texas, Oct. 1976. IEEE. (Preliminary Version).

[24] F. S. Chen, J. T. LaMacchia, and D. B. Fraser. Holographic storage in lithium niobate. *Applied Physics Letters*, 13(7):223–225, oct 1968.

[25] A. E. Chiou. Photorefractive phase-conjugate optics for image processing, trapping, and manipulation of microscopic objects. *Proceedings of the IEEE*, 87(12):2074–2085, Dec. 1999.

[26] S. Dolev and H. Fitoussi. The traveling beam: optical solution for bounded NP-complete problems. In P. Crescenzi, G. Prencipe, and G. Pucci, editors, *The fourth international conference on fun with algorithms (FUN)*, pages 120–134, 2007.

[27] J. Durand-Lose. Reversible conservative rational abstract geometrical computation is Turing-universal. In *Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, (CiE)*, volume 3988 of *Lecture Notes in Computer Science*, pages 163–172, Swansea, UK, 2006. Springer.

[28] N. H. Farhat and D. Psaltis. New approach to optical information processing based on the Hopfield model. *Journal of the Optical Society of America A*, 1:1296, 1984.

[29] D. G. Feitelson. *Optical Computing: A survey for computer scientists.* MIT Press, Cambridge, Massachusetts, 1988.

[30] S. Fortune and J. Wyllie. Parallelism in random access machines. In *Proc. 10th Annual ACM Symposium on Theory of Computing*, pages 114–118, 1978.

[31] L. M. Goldschlager. *Synchronous parallel computation.* PhD thesis, University of Toronto, Computer Science Department, Dec. 1977.

[32] L. M. Goldschlager. A unified approach to models of synchronous parallel machines. In *Proc. 10th Annual ACM Symposium on Theory of Computing*, pages 89–94, 1978.

[33] L. M. Goldschlager. A universal interconnection pattern for parallel computers. *Journal of the ACM*, 29(4):1073–1086, Oct. 1982.

[34] J. W. Goodman. Operations achievable with coherent optical information processing systems. *Proceedings of the IEEE*, 65(1):29–38, Jan. 1977.

[35] J. W. Goodman. *Introduction to Fourier Optics.* Roberts & Company, Englewood, Colorado, third edition, 2005.

[36] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo. *Limits to parallel computation: P-completeness theory.* Oxford university Press, Oxford, 1995.

[37] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. 28th Annual ACM Symposium on Theory of Computing*, pages 212–219, May 1996.

[38] P. S. Guilfoyle, J. M. Hessenbruch, and R. V. Stone. Free-space interconnects for high-performance optoelectronic switching. *IEEE Computer*, 31(2):69–75, Feb. 1998.

[39] T. Haist and W. Osten. An optical solution for the travelling salesman problem. *Optics Express*, 15(16):10473–10482, Aug. 2007. Erratum: Vol. 15(10), pp 12627. Aug. 2007.

[40] J. Hartmanis and J. Simon. On the power of multiplication in random access machines. In *Proceedings of the 15th annual symposium on switching and automata theory*, pages 13–23, The University of New Orleans, Oct. 1974. IEEE.

[41] T. Head. Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors. *Bulletin of Mathematical Biology*, 49(6):737–759, 1987.

[42] J. L. Horner, editor. *Optical Signal Processing.* Academic Press, San Diego, 1987.

[43] Y.-N. Hsu and H. H. Arsenault. Optical pattern recognition using circular harmonic expansion. *Applied Optics*, 21(22):4016–4019, Nov. 1982.

[44] A. Huang. Architectural considerations involved in the design of an optical digital computer. *Proceedings of the IEEE*, 72(7):780–786, July 1984.

[45] A. D. Jacobson, T. D. Beard, W. P. Bleha, J. D. Morgerum, and S. Y. Wong. The liquid crystal light value, an optical-to-optical interface device. In *Proceedings of the Conference on Parallel Image Processing, Goddard Space Flight Center*, pages 288–299, Mar. 1972. Document X-711-72-308.

[46] B. Javidi. Nonlinear joint power spectrum based optical correlation. *Applied Optics*, 28(12):2358–2367, June 1989.

[47] B. Javidi and J. Wang. Optimum distortion-invariant filter for detecting a noisy distorted target in nonoverlapping background noise. *Journal of the Optical Society of America A*, 12(12):2604–2614, Dec. 1995.

[48] M. A. Karim and A. A. S. Awwal. *Optical Computing: An Introduction.* Wiley, New York, 1992.

[49] R. M. Karp and V. Ramachandran. Parallel algorithms for shared memory machines. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 17. Elsevier, Amsterdam, 1990.

[50] E. Knill, R. LaFlamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409:46–52, 2001.

[51] J. N. Lee, editor. *Design Issues in Optical Processing.* Cambridge Studies in Modern Optics. Cambridge University Press, Cambridge, Great Britain, 1995.

[52] J. N. Lee, editor. *Design issues in optical processing.* Cambridge studies in modern optics. Cambridge University Press, 1995.

[53] Lenslet Labs. Enlight256. white paper report, Lenslet Ltd., 6 Galgalei Haplada St, Herzelia Pituach, 46733 Israel, Nov. 2003.

[54] R. J. Lipton. Using DNA to solve NP-complete problems. *Science*, 268:542–545, Apr. 1995.

[55] A. Louri and A. Post. Complexity analysis of optical-computing paradigms. *Applied Optics*, 31(26):5568–5583, Sept. 1992.

[56] P. D. MacKenzie and V. Ramachandran. Ercw prams and optical communication. *Theoretical Computer Science*, 196:153–180, 1998.

[57] A. D. McAulay. *Optical Computer Architectures: The Application of Optical Concepts to Next Generation Computers.* Wiley, New York, 1991.

[58] C. Mead. *Analog VLSI and Neural Systems.* Addison-Wesley, Reading, Massachusetts, 1989.

[59] D. A. Miller. Rationale and challenges for optical interconnects to electronic chips. *Proceedings of the IEEE*, 88(6):728–749, June 2000.

[60] C. Moore. Generalized shifts: undecidability and unpredictability in dynamical systems. *Nonlinearity*, 4:199–230, 1991.

[61] C. Moore. Majority-vote cellular automata, Ising dynamics and P-completeness. *Journal of Statistical Physics*, 88(3/4):795–805, 1997.

[62] T. Naughton, Z. Javadpour, J. Keating, M. Klíma, and J. Rott. General-purpose acousto-optic connectionist processor. *Optical Engineering*, 38(7):1170–1177, July 1999.

[63] T. J. Naughton. Continuous-space model of computation is Turing universal. In S. Bains and L. J. Irakliotis, editors, *Critical Technologies for the Future of Computing*, Proceedings of SPIE vol. 4109, pages 121–128, San Diego, California, Aug. 2000.

[64] T. J. Naughton. A model of computation for Fourier optical processors. In R. A. Lessard and T. Galstian, editors, *Optics in Computing 2000*, Proc. SPIE vol. 4089, pages 24–34, Quebec, Canada, June 2000.

[65] T. J. Naughton and D. Woods. On the computational power of a continuous-space optical model of computation. In M. Margenstern and Y. Rogozhin, editors, *Machines, Computations and Universality: Third International Conference (MCU'01)*, volume 2055 of *LNCS*, pages 288–299, Chişinău, Moldova, May 2001. Springer.

[66] M. Oltean. A light-based device for solving the Hamiltonian path problem. In *Fifth International Conference on Unconventional Computation (UC'06)*, volume 4135 of *LNCS*, pages 217–227, York, UK, 2006. Springer.

[67] E. L. O'Neill. Spatial filtering in optics. *IRE Transactions on Information Theory*, 2:56–65, June 1956.

[68] C. H. Papadimitriou. *Computational complexity.* Addison-Wesley, 1995.

[69] I. Parberry. *Parallel complexity theory.* Wiley, 1987.

[70] G. Păun. *Membrane computing: an introduction.* Springer, 2002.

[71] A. Pe'er, D. Wang, A. W. Lohmann, and A. A. Friesem. Optical correlation with totally incoherent light. *Optics Letters*, 24(21):1469–1471, Nov. 1999.

[72] V. R. Pratt, M. O. Rabin, and L. J. Stockmeyer. A characterisation of the power of vector machines. In *Proc. 6th annual ACM symposium on theory of computing*, pages 122–134. ACM press, 1974.

[73] V. R. Pratt and L. J. Stockmeyer. A characterisation of the power of vector machines. *Journal of Computer and Systems Sciences*, 12:198–221, 1976.

[74] A. Pu, R. F. Denkewalter, and D. Psaltis. Real-time vehicle navigation using a holographic memory. *Optical Engineering*, 36(10):2737–2746, Oct. 1997.

[75] J. Reif, D. Tygar, and A. Yoshida. The computability and complexity of optical beam tracing. In *31st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 106–114, St. Louis, MO, Oct. 1990. IEEE.

[76] J. H. Reif and A. Tyagi. Energy complexity of optical computations. In *2nd IEEE Symposium on Parallel and Distributed Processing*, pages 14–21, Dallas, TX, Dec. 1990.

[77] J. H. Reif and A. Tyagi. Efficient parallel algorithms for optical computing with the discrete Fourier transform (DFT) primitive. *Applied Optics*, 36(29):7327–7340, Oct. 1997.

[78] A. A. Sawchuk and T. C. Strand. Digital optical computing. *Proceedings of the IEEE*, 72(7):758–779, July 1984.

[79] N. T. Shaked, S. Messika, S. Dolev, and J. Rosen. Optical solution for bounded NP-complete problems. *Applied Optics*, 46(5):711–724, Feb. 2007.

[80] N. T. Shaked, G. Simon, T. Tabib, S. Mesika, S. Dolev, and J. Rosen. Optical processor for solving the traveling salesman problem (TSP). In B. Javidi, D. Psaltis, and H. J. Caulfield, editors, *Proc. of SPIE, Optical Information Systems IV*, volume 63110G, Aug. 2006.

[81] N. T. Shaked, T. Tabib, G. Simon, S. Messika, S. Dolev, and J. Rosen. Optical binary-matrix synthesis for solving bounded NP-complete combinatorical problems. *Optical Engineering*, 46(10):108201–1–108201–11, Oct. 2007.

[82] P. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. *In Proceedings 35th Annual Symposium on Foundations Computer Science*, pages 124–134, 1994.

[83] P. Sosík. The computational power of cell division in P systems: Beating down parallel computers? *Natural Computing*, 2(3):287–298, 2003.

[84] P. Sosík and A. Rodríguez-Patón. Membrane computing and complexity theory: A characterization of PSPACE. *Journal of Computer and System Sciences*, 73(1):137–152, 2007.

[85] J. Tromp and P. van Emde Boas. Associative storage modification machines. In K. Ambos-Spies, S. Homer, and U. Schöning, editors, *Complexity theory: current research*, pages 291–313. Cambridge University Press, 1993.

[86] G. L. Turin. An introduction to matched filters. *IRE Transactions on Information Theory*, 6(3):311–329, June 1960.

[87] P. van Emde Boas. Machine models and simulations. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 1. Elsevier, Amsterdam, 1990.

[88] J. van Leeuwen and J. Wiedermann. Array processing machines. *BIT*, 27:25–43, 1987.

[89] A. VanderLugt. Signal detection by complex spatial filtering. *IEEE Transactions on Information Theory*, 10(2):139–145, Apr. 1964.

[90] A. VanderLugt. *Optical Signal Processing.* Wiley, New York, 1992.

[91] P.-Y. Wang and M. Saffman. Selecting optical patterns with spatial phase modulation. *Optics Letters*, 24(16):1118–1120, Aug. 1999.

[92] C. S. Weaver and J. W. Goodman. A technique for optically convolving two functions. *Applied Optics*, 5(7):1248–1249, July 1966.

[93] D. Woods. *Computational complexity of an optical model of computation.* PhD thesis, National University of Ireland, Maynooth, 2005.

[94] D. Woods. Upper bounds on the computational power of an optical model of computation. In X. Deng and D. Du, editors, *16th International Symposium on Algorithms and Computation (ISAAC 2005)*, volume 3827 of *LNCS*, pages 777–788, Sanya, China, Dec. 2005. Springer.

[95] D. Woods. Optical computing and computational complexity. In *Fifth International Conference on Unconventional Computation (UC'06)*, volume 4135 of *LNCS*, pages 27–40, York, UK, 2006. Springer. Invited.

[96] D. Woods and J. P. Gibson. Complexity of continuous space machine operations. In S. B. Cooper, B. Löewe, and L. Torenvliet, editors, *New Computational Paradigms, First Conference*

on Computability in Europe (CiE 2005), volume 3526 of *LNCS*, pages 540–551, Amsterdam, June 2005. Springer.

[97] D. Woods and J. P. Gibson. Lower bounds on the computational power of an optical model of computation. In C. S. Calude, M. J. Dinneen, G. Păun, M. J. Pérez-Jiménez, and G. Rozenberg, editors, *Fourth International Conference on Unconventional Computation (UC'05)*, volume 3699 of *LNCS*, pages 237–250, Sevilla, Oct. 2005. Springer.

[98] D. Woods and T. J. Naughton. An optical model of computation. *Theoretical Computer Science*, 334(1-3):227–258, Apr. 2005.

[99] D. Woods and T. J. Naughton. Sequential and parallel optical computing. In *International Workshop on Optical SuperComputing*, 2008. To appear.

[100] T. Yokomori. Molecular computing paradigm – toward freedom from Turing's charm. *Natural computing*, 1(4):333–390, 2002.

[101] F. T. S. Yu, S. Jutamulia, and S. Yin, editors. *Introduction to information optics.* Academic Press, San Diego, 2001.

[102] F. T. S. Yu, T. Lu, X. Yang, and D. A. Gregory. Optical neural network with pocket-sized liquid-crystal televisions. *Optics Letters*, 15(15):863–865, Aug. 1990.

Physically-Relativized Church-Turing Hypotheses: Physical Foundations of Computing and Complexity Theory of Computational Physics

Martin Ziegler*

University of Paderborn, GERMANY

Abstract. We turn the physical Church-Turing Hypothesis from an ambiguous source of sensational speculations into a (collection of) sound and well-defined scientific problem(s):
Examining recent controversies and causes for misunderstanding concerning the state of the Church-Turing Hypothesis (CTH), it is suggested to study the CTH 'sharpened' relative to an arbitrary but specific physical theory—rather than vaguely referring to "nature" in general. For this purpose we combine physical structuralism with computational complexity theory. The benefits of this approach are illustrated by some exemplary results on computability and complexity in computational physics.

* Supported by DFG project `Zi 1009-1/2`. The author would also like to use this opportunity to express his gratitude to JOHN V. TUCKER from Swansea University for putting him on the present (scientific and career) track. BRUNO LOFF and ANDREA REICHENBERGER have kindly provided constructive remarks that helped improving this work.

1 Introduction

In 1937 Alan Turing investigated the capabilities and fundamental limitations of a mathematical abstraction and idealization of a computer. Nowadays, this Turing machine (TM) is considered the most appropriate model of actual digital computers reflecting what a common PC can do or cannot do, and capturing its fundamental capabilities in computability and complexity classes: any computation problem that can be (efficiently) solved in practice by a PC belongs to Δ_1 (to $\mathcal{P}$); and vice versa. In this sense, the TM is widely believed to be universal; and problems $P \notin \mathcal{P}$, or the Halting problem $H \notin \Delta_1$, have to be faced up as principally unsolvable.

1.1 Turing Universality in Computer Science and Mathematics

There are some good reasons for this belief:

- There exists a so-called *universal* Turing machine (UTM), capable of simulating (with at most polynomial slowdown) any other given TM.
- Several other reasonable models of computation have turned out as equivalent to the TM: `WHILE`-programs, λ–calculus etc. Notice that these models correspond to real-world programming languages like `Lisp`.

We qualify those good reasons as *computer scientific*—in contrast to the following evidence based on a purely mathematical notion:

- An integer function f is TM-computable iff it is μ-recursive;
  that is, f belongs to the least class of functions
  - containing the constant function 0,
  - the successor function $x \mapsto x+1$,
  - the projections $(x_1,\ldots,x_n) \mapsto x_i$,
  - and being closed under composition,
  - under primitive recursion,
  - and under so-called μ-recursion.

1.2 Turing Universality in Physics

The[⋆] Church-Turing Hypothesis (CTH) claims *that every function which would naturally be regarded as computable is computable under his* [i.e. Turing's] *definition, i.e. by one of his machines* [Klee52, p.376]. Its *strong* version claims that *efficient* natural computability corresponds to *polynomial-time* Turing computability. Put differently, CTH predicts a negative answer to the following

Question 1. Does nature admit the existence of a system whose computational power strictly exceeds that of a TM?

Notice that the CTH exceeds the realm of computer science; it involves *physics* as the general analysis of *nature*. Hence, in addition to the computer scientific and mathematical dimensions of Turing universality, a third dimension would arise if the answer to Question 1 turned out to be negative (cmp. [Benn95,Svoz05]):

- The class of (efficiently) physically computable functions coincides with the class of (polynomial-time) Turing computable ones.

Insofar a TM can be built at least in principle[⋆⋆] it constitutes a physical system. Conversely, a negative answer to Question 1 means that *every* physical 'computer' can be simulated (maybe even in polynomial time) by a TM. Such an answer is supported by long experience in two ways:

[⋆] To be fair, this is just *one* out of a variety of refined interpretations of a claim which neither ALONZO CHURCH nor ALAN TURING have ever put forth; see, e.g. [Hodg06], [Ord02, SECTION 2.2], [Cope02], or [LoCo08]. In fact I refer to what in in literature is more specifically called the *physical* Church-Turing Hypothesis, but for reasons of conciseness shall in the sequel omit this adjective.

[⋆⋆] it is for instance realized (in good approximation) by any standard PC

- the constant failure to physically solve the Halting problem and
- the success of simulating a plethora of physical systems on a TM, namely in Computational Physics.

So far all attempts have failed to prove the CTH, i.e. have given at best bounds on the *speed* of calculations but not on the general capabilities of computation, based e.g. on the laws of thermodynamics [BeLa85,Fran02] or on the speed of light (special relativity) [Lloy02]. In fact it has been suggested [Loff07, p.39] that the Church-Turing Hypothesis be included into physics as an *axiom* (called the Church-Turing *criterion*): just like the impossibility to extract energy from nothing (or from thermal energy alone) started as a recurring experience and was then postulated as the First (Second) Law of Thermodynamics—and only later 'justified' using Statistical Mechanics. Either way, whether axiomatizing or trying to prove the Church-Turing Thesis, first of all one needs a formalization of Question 1.

Such attempts are, of course, not new. For instance, *Gandy Machines* have been proposed [Gand80,Sieg00] a model of computation based on some specific physical *principles*. Our approach on the other hand is based on arbitrary physical *theories*, providing a more thorough and accurate formal coverage of '*nature*'.

1.3 Summary

The CTH is the subject of a plethora of publications and of many disputes and speculations. The present work aims to put some reason into the ongoing [Galt06] and often sensational [Kie03b,Lloy06] discussion. We are convinced that formalizing Question 1 will come useful. However, it seems unlikely that one single formalization can reach consensus. We notice that most, if not all, disputes about the state of the Church-Turing Hypothesis arise from disagreeing and/or implicit conceptions of how to formalize it [Smi99b]. The best we can hope for is a *class* of formalizations, namely one for each physical theory [BeTu07, SECTION 2(b)]. Let me fortify this:

Manifesto 2. *a) Describing the scientific laws of nature is the purpose and virtue of physics. It does so by means of various physical theories Φ, each of which 'covers' some part of reality (but may deviate on other parts).*

b) Consequently, instead of vaguely referring to 'nature' *(recall Section 1.2) any claim concerning (the state of) the CTH should explicitly mention the specific* physical theory Φ *it considers;*

c) and criticism against such a claim as 'based on unrealistic presumptions' should be regarded as directed towards the underlying physical theory (and stipulate re-investigation subject to another Φ, rather than dismissing the claim itself); cmp. [BeTu07, PRINCIPLE 2.4].

d) Also the input/output encoding better be specified explicitly when referring to some "CTH$^\Phi$": How is the argument $\vec{x}$, of natural or real numbers, fed into the system; i.e. how does its preparation (e.g. in Quantum Mechanics) proceed operationally; and how is the 'result' to be read off (e.g. what 'question' is the system to answer)? [Ship93, SECTION I]

Particularly Item b) is central to the present work and explains for its title!

The suggestion to consider physically-relativized Church-Turing Hypotheses "CTH$^\Phi$" (cmp. also [BeTu07, PRINCIPLE 2.1]) bears the spirit of the related treatment of the famous "$\mathcal{P}$ versus $\mathcal{NP}$" Question in [BGS75]; there it has been shown that, relative to some oracle, both complexity classes coincide whereas relative to another one they do not.

Section 3 below expands on the concept of a physical theory Φ and its analogy to a model of computation in computer science. We turn Manifesto 2b) into a research programme (Section 6) and illustrate its benefit for computational physics. Before, Section 2 reports on previous attempts to disprove the CTH by examples of hypercomputers purportedly capable of solving the Halting problem, and the respective physical theories they exploit. Subsequently, we simplify one such example in order to inspect its source of computational power. Based on this insight, we are in Section 4.2 led to extend the above Manifesto:

Manifesto 2 (continued).

e) The concept of "existence" of a hypercomputational physical system (recall Question 1) must be interpreted from a constructivistic point of view.

2 Physical Computing

Common (necessarily informal) arguments in favor of the Church-Turing Hypothesis usually proceed along the following line: A physical system is mathematically described by an ordinary or partial differential equation which can be solved numerically using time-stepping—as long as the solution remains regular: whereas a singular solution is unphysical or too unstable (or both) in order to be harnessed for physical computing.

On the other hand, in literature there are a variety of suggestions for physical systems of computational power exceeding that of a TM, e.g.

Example 3 *i) General Relativity might admit for space-times such that the clock of a TM M following one world-line seems to reach infinity within finite time according to the clock of an observer O starting at the same event but following another world-line; thus O can decide whether M terminates or not* [EtNe02].

However it is unknown whether such space-times actually exist *in our universe; and if they exist, how to locate them and how far from earth they might be in order to be used for solving the Halting problem. (Notice that the closest known Black Hole, namely next to star* `V4641`*, takes at least* 1600 *years to travel to). Finally, it has been criticized that the TM in question had to run indefinitely—and use corresponding amounts of storage tape and energy.*

ii) While 'standard' quantum computers using a finite number of qubits can be simulated on a TM (although possibly at exponential slowdown), Quantum Mechanics (QM) supports operators on in*finite superpositions which could be exploited to solve the Halting problem* [CDS00,Kie03a,ACP04,Zieg05].

On the other hand there are already considerable doubts that finite[***] *quantum parallelism is in considerable is practical due to issues of* decoherence*, i.e. susceptibility to external, classical noise (a kind of instability if you like); hence how much more unrealistic be infinite one!*

iii) Already in their mathematical formulation, certain theories of Quantum Gravitation involve combinatorial conditions which are known undecidable to a TM [GeHa86].

These, however, are still mere (and preliminary) theories...

iv) A light ray passing through a finite system of mirrors corresponds to the computation of a Turing machine; and by detecting whether it finally arrives at a certain position, one can solve the Halting problem [RTY94].

The catch is that the ray must adhere to Geometric Optics, i.e. have infinitely small diameter, be devoid of dispersion, and propagate instantaneously; also the mirrors have to be perfect.

v) The above claim that singular solutions can be ruled out is put into question by the discovery of non-collision singularities in Newtonian many-body systems [Yao03,Smi06a,Svoz07].

On the other hand, the construction of these singularities heavily relies on the moving particles being ideal *points obeying Newton's Law (with the singularity at 0) up to* arbitrary *small distances.*

vi) Even Classical Mechanics has been suggested to allow for physical objects which can be probed in finite time to answer queries "$n \in X$" for any fixed set $X \subseteq \mathbb{N}$ (and in particular for the Halting problem) [BeTu04].

However creating this classical device seems to require solving X in the first place.

vii) Already 1D heat conduction has been shown [PERi81] capable of evolving a computable initial condition $x \mapsto u(0,x)$ into an uncomputable one $x \mapsto u(1,x)$. Here the caveat is that, while the initial condition is computable and continuously differentiable, its derivative is not, and hence again implicitly contains the superrecursive power as input [WeZh02].

Notice that each approach is based on and exploits sometimes beyond recognition some (more or less specific) physical theory. Also, the indicated reproaches against each approach to hypercomputation aim at the physical theory it is based on.

[***] The present world-record seems to provide calculations on only 28 qubits; and even that is rather questionable [Pont07]

3 Physical Theories

have been devised as the scientific means for objectively describing and predicting the behavior of nature for thousands of years. Nowadays, we may feel inclined to patronize e.g. ARISTOTLE's eight books, but his concept of *Elements* (air, fire, earth, water) constitutes an important first step towards putting some structure into the many phenomena experienced[†]

Since Aristotle, a plethora of physical theories of space-time has evolved (cf. e.g. [Duhe85]), associated with famous names like GALILEO GALILEI, PTOLEMY, NICOLAUS COPERNICUS, JOHANNES KEPLER, SIR ISAAC NEWTON, HENDRIK LORENTZ, and ALBERT EINSTEIN. Moreover, theories of electricity and magnetism have sprung and later became unified (JAMES CLERK MAXWELL) with GAUSSian Optics. And there are various[‡] quantum mechanical and field theories. The unification process continued: Electricity and Magnetism, been merged into Electrodynamics, were joined by Quantum Mechanics to make up Quantumelectrodynamics (QED), and then with Weak Interaction formed Electroweak Interaction; moreover Gravitation and Special Relativity became General Relativity.

Remark 4 (Analogy between a Physical Theory and a Model of Computation). *Each such theory has arisen, or rather been devised in order to describe* some *part of nature with sufficient accuracy—while necessarily neglecting others. (Quantum Mechanics for instance is aimed at describing elementary particles moving considerably slower than light; whereas Relativity Theory focuses on very fast yet macroscopic objects.) We point out the analogy of a physical theory to a model of computation in computer science: Here, the aim is to reflect* some *aspects of actual computing devises as well, possibly at the expense of other aspects. (A Turing machine has unbounded working tape; hence it can decide whether a 4GB-memory bounded PC algorithm terminates. Whereas the canonical model for computing devises with finite memory, a DFA is unable to decide the correct placement of brackets.)*

But what exactly *is* a physical theory? In addition to a means for clearing up misunderstandings as indicated in Footnote[‡], agreement on this issue is a crucial prerequisite for treating important further questions like:

> *Are Newton's Laws an extension of Kepler's?* [Duhe54]
> *Does Quantum Mechanics imply Classical Mechanics—and if so, in what sense exactly?*

To us, such intertheory relations [Batt07,Stoe95] are in turn relevant in view of the above Manifesto 2 with questions as the following one:

> *Do the computational capabilities of Quantum Mechanics include those of Classical Mechanics?*

3.1 Structuralism in Physics

Just like a physical theory is regularly obtained by trying to infer a simple description of a family of empirical data points obtained from experimental measurements, a *meta*-theory of physics takes the variety of existing physical theories as empirical data points and tries to identify their common underlying structure. The philosophy of science knows several meta-theories of physics, i.e., conceptions of what a physical theory is [Schm08]:

- SNEED focuses on their mathematical aspects [Snee71]; STEGMÜLLER suggests to formalize physical theories in analogy to the Bourbaki Programme in mathematics [Steg79,Steg86].
- C.F. V. WEIZSÄCKER envisions the success of unifying previously distinct theories (recall above) to continue and ultimately lead into a "Theory of Everything" [Weiz85,Sch97b]. From this point of view any other physical theory (like e.g. Newton Mechanics) is merely a tentative draft [Wein94].

[†] Even more, closer observation reveals that an argument like "*A rock flung up will fall down,* because *it is a rock's nature to rest on earth.*" is no less circular than the following two more contemporary ones: "*A rock flung up will fall down,* because *there is a force pulling it towards the earth.*" and *Electrons in an atom occupy different orbits,* because *they are Fermions.*

[‡] Remember how scientists regularly get into a fight when starting to talk about (their conception of) Quantum Mechanics

- MITTELSTAEDT emphasizes pluralism in physical theories, i.e., various theories equally appropriate to describe the same range of phenomena [Mitt72, SECTION 4]. Also [Hage82] points out (among many other things) that any physical theory or *model* is a mere approximation and idealization of reality.
- For our purpose, LUDWIG [Ludw90] and, building thereon, SCHRÖTER [Schr96] propose the most appropriate and elaborated formalization, based on the following (meta-)

Definition 5 (Sketch). *A physical theory Φ consists of*

- *a description of a part of nature it applies to (WB)*
- *a mathematical theory as the language to describe it (MT)*
- *and mapping between physical and mathematical objects (AP).*

In this setting, each physical theory has a specific and limited range of applicability (WB): a quite pragmatic approach, compared to the almost eschatological hope of von Weizsäcker and Weinberg for a ToE. From the point of view of Ludwig and Schröter on the other hand, each new theory adds to and extends with its WB (=images of MT under AP) that part of nature 'covered' by physics: just like a mathematical manifold being covered and described by the images of Euclidean subsets under charts [Miln97]; see also [Svoz06].

Remark 6. Some major physical theories have already been formalized in the sense of Definition 5; we mention for instance two[‡] Quantum Mechanics [Ludw85,Haet96] and space-time theories [Meis90,Schr88,ScSc92,Sche92,Sch97a]. Fortunately, our approach suffices with a rather low level of detail compared to the depth of formalization of each of the three components (WB,MT,AP) provided by the exhaustive [Schr96].

3.2 On the Reality of Physical Theories

The purpose of a physical theory Φ is to describe a part of nature. Hence, when some better description Φ′ is found, a 'revolution' may occur and Φ gets disposed of [Kuhn62]. However a full rejection of Φ has never happened (and is one source of criticism against Kuhn): more commonly, the new theory Φ′ is applied to those parts of nature which the old one would not describe (sufficiently well) while keeping Φ for applications where it long has turned as appropriate.

Example 7 *a) Classical/Continuum Mechanics (CM) for instance is often heard of as 'wrong'—because matter is in fact composed from atoms circled by electrons on stable orbits—yet it still constitutes the theory which most mechanical engineering is based on.*

b) Similarly, audio systems are successfully designed using Ohm's Law for (complex) electrical resistance: in spite of Maxwell's Equations being a more accurate description of alternating currents, not to mention QED.

Ludwig in [Ludw85] has proven that QM, often considered as 'better' (in the sense of more fundamental and realistic) a theory than CM, does *not* include or imply CM; see also [Boku08]. (Nevertheless such claims regularly re-emerge especially in popular science.) Moreover, even QM itself is merely[§] an approximation to parts of nature, unrealistic e.g. at high velocities or in the presence of large masses.

These observations urge us to enhance Manifesto 2a+c) with an ontological commitment to scientific realism:

Manifesto 8. *A physical theory Φ (like, e.g. CM) constitutes an element of reality: It* exists[¶] *no less than "points" or "atoms" do. In particular, it is advisable to investigate the computational power of, and within such a Φ: Dismissing a theory for being 'unrealistic' in one extend or another would in consequence deprive us of* all *of them!*

[§] In particular we disagree with the, seemingly prevalent, opinion that Quantum Theory is somehow salient or even universal in some sense [HaHa83,Holl96]

[¶] Note that this refers to existence on the *meta*-level of physical theories, as opposed to the existence of a physical object *within* a physical theory according to Manifesto 2e). Our distinction between these two levels is analogous to GÖDEL's (meta-)proof of the existence (!), for any [...] mathematical theory, of an arithmetical statement which cannot, *within* this theory, be proven nor refuted.

Again, we stress the analogy to theoretical computer science (Remark 4) studying the computational power of models of computation M (e.g. finite automata, nondeterministic pushdown automata, linear-bounded nondeterministic Turing machines: the famous Chomsky Hierarchy *of formal languages)* although *each such M is unrealistic in* some *respects.*

4 Hypercomputation in Classical Mechanics?

Let us exemplify Example 3vi) with an alternative 'hypercomputer' similar to the one presented in [BeTu04] yet stripped down to purely exhibit the core idea and make it accessible for further study.

Example 9 *Consider a solid body, a cuboid into which has been carved a 'comb' with infinitely many teeth of decreasing width and distance, cf. Figure 1. Moreover, having broken off tooth no.n iff* $n \notin H$*, we arrive at an encoding of the Halting problem into a physical object in CM.*

This very object (together with some simple mechanical control) is a hypercomputer: It can be read off and used to decide for each $n \in \mathbb{N}$ *the question "*$n \in H$*?" by probing with a wedge the presence of the corresponding tooth.*

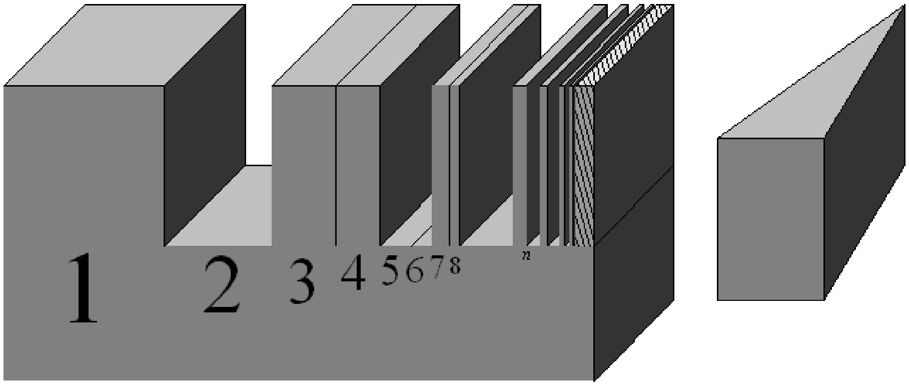


Fig. 1. Infinite comb with a wedge to probe its teeth

A first reproach against Example 9 might object that the described system, although capable of solving the Halting problem, is no hyper*computer*: because it cannot do anything else, e.g. simulate other Turing machines. But this is easy to mend: just attach the system to a universal TM, realized in CM [FrTo82].

The second deficiency of Example 9 is more serious: the concept of a solid body in CM is merely an idealization of actual matter composed from a very large but still finite number of atoms—bad news for an infinite comb. However, as pointed out in Section 3.2, we are to take a physical theory for serious and study the computational power of, and within CM.

But there remains another important

Observation 10 (Third issue about Example 9) *Even within CM, i.e. granting the existence of ideal solids and infinite combs: How are we to get hold of one encoding H? Obviously, one cannot* construct *it from a blank without solving the Halting problem in the first place. Our only chance is simply to* find *one (e.g. left behind by some aliens* [Clar68,StSt71]*) without knowing how to create one ourselves.*

4.1 Existence in Physics

In order to formalize the Church-Turing Hypothesis (a prerequisite for attempting to settle it), we notice an ambiguity about the word "*exist*" in Question 1 pointed out already in [Zieg05, REMARK 1.4]: For a physical object, "existence" (within a physical theory) means that

A) one has to actually *construct* it?
B) its *non*-existence leads to a contradiction?
C) or that its *existence* does *not* lead to a contradiction (i.e. is consistent)?

These three opinions correspond in mathematics to the points of view taken by a *constructivist*, a *classical* mathematician (working e.g. in the Zermelo-Fraenkel framework) and one 'believing' in the Axiom of Choice, respectively. The last standpoint (C) is well known to lead to counter-intuitive consequences when taken in the physical realm of CM:

Example 11 (Banach-Tarski Paradoxon) *For a solid ball (say of gold) of unit size in 3-space, there exists a partition into finitely many (although necessarily not Lebesgue-measurable) pieces that, if put together appropriately (i.e. after applying certain Euclidean isometries), then form* two *solid balls of unit size.*

This example (see also [Svoz97]) is in no danger of causing inflation: first, because actual material gold is not infinitely divisible (cmp. the second deficiency of Example 9); secondly, even *within* CM, because the partition of the ball 'exists' merely in the above Sense C).

Hence, in order to avoid both 'obviously' unnatural (counter-) Examples 9 and 11, Manifesto 8 leads us to transfer and adapt the constructivist standpoint from mathematics to physics and for physics as well.

4.2 Constructivism into Physical Theories!

As explained above, the "existence" of some physical object within a theory Φ has to be interpreted constructively; cmp. e.g. PROBLEMS 6.2 and 6.3 in [BeTu04]. To this end and as in [CDCG95], let us distinguish two ways of introducing constructivism into a physical theory $\Phi = (\mathrm{MT}, \mathrm{AP}, \mathrm{WB})$:

α) By interpreting the mathematical theory MT constructively; compare [BiBr85,Kush84,BrSv00,Flet02] and [DSKS95, SECTION III].
β) By imposing constructivism onto the side of physical objects WB.

It seems that Method α), although meritable of its own, does not quite meet our goal of making a *physical* theory constructive:

Example 12 *Consider the condition for a function $f : X \to Y$ between normed spaces to be* open*; or even simpler: that of the image $f[B(0,1)] \subseteq Y$ of the unit ball in X to be an open subset of Y.*

$$\forall u \in f[B(0,1)] \;\; \exists \varepsilon > 0 \;\; \forall y \in B(u,\varepsilon) \;\; \exists x \in B(0,1) : \quad f(x) = y \quad . \tag{1}$$

A constructivist would insist that both *existential quantifiers have to be interpreted constructively; whereas in a setting of computation on real numbers by rational approximation, applications suffice that only ε is computable from u, while the existence of x depending on y need not: compare* [Zieg06].

4.2.1 Constructing Physical Objects

Underlying β) is the conception that every object in nature (or more precisely: that part of nature described by WB) be

- either a primitive one (e.g. a tree, modeled in Φ as a homogeneous cylinder of density $\rho = 0.7\mathrm{g/cm}^3$; or some ore, modeled as $\mathtt{Cu\,Fe\,S_2}$)
- or the result of some technological process applied to such primitive objects.

The latter may, e.g., include crafting a tree into a wheel or even a wooden gear; or smelting ore to produce bronze.

Notice how such a process—the sequence of operations from cutting the tree, cleaning, sawing, carving; or of melting, reducing, and alloying copper—constitutes an algorithm (and crucial cultural knowledge passed on from carpenters or redsmiths[‖] to their apprentices). More modern and advanced science, too,

[‖] An attentative referee has pointed out the inconsequence to admit human interaction for creating a hypercomputer, but not during the computation itself. Permitting the latter leads to notions considered, e.g., in [Mins68] or [GoWe08].

knows and teaches 'algorithms' for constructing physical objects: e.g., in mechanical engineering (say, designing a gear); or in QM (using a furnace with boiling silver and some magnets to create a beam of spin-$\frac{1}{2}$ particles as in the famous STERN and GERLACH Experiment: This is an example of operationally *constructing* a physical object, namely corresponding via AP to a certain wave function ψ as a mathematical object in MT).

Thus, we are led to extend Definition 5:

Definition 13 (Meta-). *The WB of a physical theory Φ consists of*

- *a specific collection of* primitive *objects (PrimOb)*
- *and all so-called* constructible *objects,*

i.e., that can be obtained from primitive ones by a sequence (o_i) of preparatory operations.

- *The latter are elements o from a specified collection PrepOp.*
- *Moreover, the sequence (o_i) must be 'computable'.*

The first two items of Definition 13 are analogous to a mathematical theory MT consisting of *axioms* (i.e. claims which are true by definition) and *theorems*: claims which follow from the axioms by a sequence of *arguments*. The last requirement in Definition 13 is to prevent the body in Example 9 from being 'constructed' by repeated** "breaking off a tooth" as preparatory operations. On the other hand, we seem to be heading for a circular notion: trying to capture the *computational* contents of a physical theory Φ has required to restrict to 'constructible' objects which, in turn, are defined as the result of a *computable* sequence of preparatory operations. That circle is avoided as follows:

Definition 13 (continued).*'Computability' means relative to a pre-theory φ to, and to be specified with, Φ.*

4.2.2 Pre-Theories: Ancestry among Physical Theories

Recall the above example from metallurgy of redoxing an ore: this may described by the phlogiston theory (an early form of theoretical chemistry, basically extending Aristotle's concept of four Elements by a fifth resembling what nowadays would be considered oxygen). Such a 'chemical' theory φ of its own is required to formulate (yet does not imply) metallurgy Φ and, in particular, the algorithm therein that yields to bronze: φ is a pre-theory to Φ.

We give some further and more advanced examples of pre-theories:

Example 14 *a) The classical Hall Effect relies on Ohm's law of electrical direct current as well as on Lorentz' force law.*

b) The Stern-Gerlach experiment, and the quantum theory of spin Φ it spurred, is based on
  - *a classical, mechanical theory of a spinning top and precession;*
  - *some basic theory of (inhomogeneous) magnetism and in particular of Lorentz force onto a dipole*
  - *an atomic theory of matter (to explain e.g. the particle beam)*
  - *and even a theory of vacuum (*TORRICELLI, VON GUERICKE*).*

c) In fact, any quantum theory of microsystems requires [Ludw85] some macroscopic pre-theory in order to describe the devices (furnaces, scintillators, amplifiers, counters) for preparing and measuring the microscopic ensembles under consideration.

d) BARDEEN, COOPER, *and* SCHRIEFFER*'s Nobel prize-winning BCS-Theory of superconductivity is essentially based on QM*

e) whereas superconducting magnets, in turn, are essential to many particle accelerators used for exploring elementary particles.

The reader is referred to [Schr96, DEFINITION 4.0.8] for a more thorough and formal account of this concept.

** The reader may be tempted to admit only finite sequences of preparatory operations. However, this would exclude woodturning a handrail out of a wooden cylinder by letting the carving knife follow a curve, i.e. a continuous sequence.

Observation 15 *Technological progress can be thought as a directed acyclic graph: a node u corresponds to a physical theory Φ and may be based on (one or more) predecessor nodes, pre-theories φ to Φ. Put differently, physical theories form* nets *or logical* hierarchies*; cmp.* [Schr96, VERMUTUNG 14.1.2] *and* [Stoe95].

It would be an interesting endeavor to trace the net of (sub-)theories which the Large Hadron Collider (LHC) at CERN is based on.

5 Computational Physics

Over the last few decades, computer simulations of physical systems have become an important new discipline of physics of its own (in addition to experimental, applied, and theoretical physics). It has, however, received very little support on behalf of Theoretical Computer Science. Specifically scientists working in this area—typically highly-skilled programmers with an extensive education in physics and excellent intuition for it—are highly interested in

Question 16. For a specific (class of) physical systems Φ:

1. Why are they so hard to simulate (in the sense of computing resources like CPU-time) ?
2. How about computational predictions of their long-term behavior? (Compare [Ship93, SECTION I]. . .)
3. Are our respective algorithms optimal, and in what sense?

And, more generally: Which are the principal limits of computer simulation?

Answers to such questions for various Φ (namely theories in the sense of Section 3) are highly appreciated in Computational Physics: answers given of course in the language of Computational Complexity Theory [Papa94] and using the methods thereof, namely locating Φ in some complexity (or recursion theoretic) class *and* proving that it is complete for that class.

We observe that there are rather few serious and rigorous answers to such questions to-date [FLS05,Wolf85,Krei87,Moor90,Ship9

5.1 Complexity in Computational Physics ↔ Physically-Relativized Church-Turing Hypotheses

CTH and Question 1 are usually considered with the purpose of exploiting nature's computing capabilities. In view of our physically-relativized approach this requires, for a (hyper-)computational system Φ,

- the ability to operationally *initialize* it ('preparation') with a given argument x;
- to read-off the resulting value $f(x)$ calculated by Φ;
- to detect if (or, preferably, even predict when) Φ has completed its 'computation'.

These properties are clearly violated by most items in Example 3; hence, they are (even more) unlikely to give rise to a hypercomputer. However, in the case of computer-simulating a physical system Φ, the above issues vanish: Scientists regularly set up initial conditions for two inspiraling black holes, say, and then follow their evolution numerically [MTB*07].

Observation 17 *A system Φ which turns out to violate CTH has important consequences for computational physics, even though it might not be harnessable as a hypercomputer: It follows that Φ is intractable to simulations on a Turing machine (and thus neither on any nowadays digital computer).*

Indeed, a variant, the (strong) *Simulation Church-Turing Hypothesis* claims (cf. e.g. [Loff07, PART I]):

> *Every function which simulates a finite physical system can be computed by a Turing machine (in polynomial time).*

Manifesto 2 applies also to this variant of the CTH and yields a natural approach towards resolving Question 16(1) as well as Questions 16(2-4): Fix some well-specified physical theory Φ and explore both (Turing-) computability and computational complexity of simulating and predicting the behavior of systems in Φ. Here, one is particularly interested in *completeness* results: In the extreme case, the physical theory under consideration is rich enough to allow for universal computation (Turing-completeness) and consequently incorporates the Halting problem; this means that such a Φ is (at least in the worst-case sense) intractable to Computational Physics; recall e.g. [GeHa86]. But even systems which do comply with CTH might violate its *strong* version, i.e., admit superpolynomial lower time complexity bounds; or, more realistically, they can be proven complete for some classical complexity class like $\mathcal{NP}$ or $\mathcal{PSPACE}$:

Observation 18 *If (any part of) nature is capable of universal computation, this entails some sort of computational completeness, that is a lower complexity bound.*

Remark 19. (Not just) Physicists may tend to avert such negative results: they want solutions, not infeasibility. However, trying to (devise algorithms to) solve a proven (e.g. $\mathcal{NP}$, $\mathcal{PSPACE}$, or even Turing-) hard problem is of course a waste of effort, time, and money. Hence, seemingly 'negative' results can in fact be used positively:

- as a kind of 'radar' guiding around pitfalls of intractability;
- or as an incentive to slightly modify the problem under consideration (e.g. from exact to approximate, recall `Knapsack`) in order to make it tractable;
- or as a proof that some algorithm already employed is in fact optimal (and attempts for further improvement thus in vain).

Compare also DAVID HILBERT's emphasis in his famous 1900 presentation in Paris on the high problem solving effectiveness of negative results in mathematical research.

5.2 Real Number Computation

As opposed to classical problems considered in computational complexity, those problems arising in Computational Physics typically involve real numbers [PERi89,WeZh02,WeZh06]. In order to study their complexities three major approaches are at hand:

a) the bit model: using Turing machines to calculate (numerator and denominator of) rational or dyadic approximations q_n to the desired real x up to some prescribed error [Turi36,Grze57]. Here, complexity is inherently connected to classical, discrete classes like $\mathcal{P}$, $\mathcal{NP}$, and $\mathcal{PSPACE}$ [Frie84,Ko91].
b) the algebraic model: replacing the Boolean $(\{0,1\},\vee,\wedge,\neg)$ and/or integer ring $(\mathbb{Z},+,-,\times,<)$ with that of reals $(\mathbb{R},+,-,\times,<)$, that is considering real numbers as primitives and arithmetic operations thereon as exact, one arrives at the BSS Machine [BCSS98] or real-RAM [PrSh85, SECTION 1.4]; see also [TuZu00]. So one arrives at complexity classes corresponding to the classical ones, too [BSS89,MeMi97].
c) Restrict to rational inputs!

Approach b) models fixed precision (e.g. `double`) floating point arithmetic whereas approach a) is more appropriate for calculations using adaptive accuracy. Note that in both cases, uncomputability/hardness easily occurs with*out* completeness [Wolf85,Moor90,Smi06a,MeZi06].

Approach c) does not fully count as *real* number computation. Also, carelessly restricting the domain of a problem might deprive it of some important features; recall e.g. [Xia92,Smi06a]. Finally, if the output is not rational again, closure under composition fails already semantically.

6 Vision of a Research Programme

We encourage a systematic exploration of the computational power (i.e. completeness) of a large variety of physical theories [BeTu07, PRINCIPLE 2.2]. The first goal is a general picture of physically-relativized Church-Turing Hypotheses, i.e., on the boundary between decidability and Turing-completeness [BeTu07, PRINCIPLE 2.3]. In the next step one may turn to lower complexity classes like $\mathcal{EXP}$, $\mathcal{PSPACE}$, $\mathcal{NP}$, $\mathcal{P}$, and $\mathcal{NC}$. The focus be on a thorough investigation, starting from simplest, decidable theories and slowly proceeding towards more complex ones (not necessarily in historical order) rich enough to admit a Turing-complete system therein. In particular, it seems advisable to begin with rather modest physics:

6.1 Celestial Mechanics

Recall the historical progress of describing and predicting the movement of planets and stars observed in sky from Eudoxus/Aristotle via Ptolemy, Copernicus, and Kepler to Newton and Einstein. These descriptions constitute physical theories! (They are not necessarily comparable in the sense of reduction.) The present subsection exemplifies our proposed approach by investigating and reporting on the computational complexities of two of them.

6.1.1 Newton

Particularly with regard to computability and complexity, Newtonian Mechanics is far from a simple theory. For this reason we are going to study certain simple subtheories of it [BeTu07, SECTION 2(c)]. Specifically, let us consider a physical theory Φ of N points moving in Euclidean 3-space under mutual attracting force proportional to distance^{-2} (inverse-square law). This is the case for Electrostatics (Coulomb) as well as for Classical Gravitation (Newton). Some concretions of Question 16 may ask:

1. Does point #1 reach within one second the unit ball B centered at the origin?
2. Does some point eventually escape to infinity?
3. Do two points (within 1sec or ever) collide?

It has been argued that Question 3 makes not much sense, because a 'collision' of *ideal* points (recall Manifesto 8) can be analytically continued just to pass through each other. Note that Question 1 is not 'well-posed' in case that the point just touches the boundary of B; it is therefore usually accompanied by the *promise* that point #1 either meets the interior of B within one second or avoids the blown-up ball $2B$ for two seconds; and shown $\mathcal{PSPACE}$-hard in this case [ReTa93]. Concerning Question 2, recently it has been revealed that a point may actually escape to infinity within *finite* time [Xia92]; the question of whether this happens has been shown undecidable [Smi06a]—although for input configurations described by (possibly transcendental) *real* numbers given as *in*finite sequences of rational approximations: for such encodings, mere discontinuity is known to imply uncomputability trivially and without completeness [Grze57].

6.1.2 Planar Eudoxus/Aristotle

An early theory of celestial mechanics originates from ancient Greece. An important purpose of it (and also of its successors, see Section 6.1.4 below) was to describe and predict the movement of planets and stars and in particular their conjunctions. Let us capture these aims in the following

Question 20. 1. Will certain planets ever attain perfect conjunction?
2. or within a given time interval?
3. or reach an approximate conjunction, i.e. meet up to some prescribed angular distance ε?

According to ARISTOTLE (Book Λ of *Metaphysics*) and EUDOXUS OF CNIDUS, earth resides in the center of the universe (recall the beginning of Section 3) and is circled by *celestial spheres* moving the celestial bodies.

Definition 21. *Let Φ denote the physical theory (which we refrain from fully formalizing in the sense of Definition 5 or even* [Schr96]*) parameterized by the initial positions u_i of planets $i = 1,\dots,N$, and their constant directions $\vec{d_i}$ and velocities v_i of rotation.*

By Φ', we mean a two-dimensionally restricted version: planets rotate on circles perpendicular to one common direction; compare Figure 2. Moreover, initial positions and angular velocities are presumed 'commensurable[††]*', that is, rational (multiples of π).*

Recall that $\mathcal{NC} \subseteq \mathcal{P}$ is the class of problems solvable in polylogarithmic parallel time on polynomially many processors; whereas $\mathcal{P}$–hard problems (say w.r.t. logspace-reductions) presumably do not admit such a beneficial parallelization. The greatest common divisor $\gcd(a,b)$ of two given (say, n-bit) integers

[††] We don't want anybody to get drowned like, allegedly, HIPPASUS OF METAPONTUM. Also, since rational numbers are computable, we thus avoid the issues from Section 4.2.

Fig. 2. Celestial orbs as drawn in PETER APIAN's *Cosmographia* (Antwerp, 1539)

can be determined[‡‡] in polynomial time; however it is unknown whether it belongs to $\mathcal{NC}$ or is $\mathcal{P}$-hard; the same holds for the calculation of an extended Euclidean representation "$a \cdot y + b \cdot z = \gcd(a,b)$", i.e. of $(y,z) = \mathrm{gcdex}(a,b)$ [GHR95, B.5.1].

After these preliminaries, we are able to state the computational complexity of the above theory Φ'; more precisely the complexity of the decision problems raised in Question 20 in terms of Φ's parameters:

Theorem 22. *Let $k \leq n \in \mathbb{N}$ and $u_1,\ldots,u_n,v_1,\ldots,v_n \in \mathbb{Q}$ be given initial positions and angular velocities (measured in multiples of 2π) of planets $\#1,\ldots,\#n$ in Φ'.*

a) Planets #1 and #2 will eventually appear in perfect conjunction iff $v_1 \neq v_2 \vee u_1 = u_2$.

b) Planets #1 and #2 appear closer than $\varepsilon > 0$ to each other within time interval (a,b) iff it holds, in interval notation:

$$\emptyset \;\neq\; \mathbb{Z} \cap \big((a,b)\cdot(v_1 - v_2) + u_1 - u_2 + (-\varepsilon,+\varepsilon)\big) \ .$$

This can be decided within $\mathcal{NC}^1$.

c) The question of whether all *planets $\#1,\ldots,\#n$ will ever attain a perfect conjunction, can be decided in $\mathcal{NC}^{\mathrm{gcd}}$;*

d) and if so, the next time t for this to happen can be calculated in $\mathcal{NC}^{\mathrm{gcdex}}$.

e) Whether there exist k (among the n) planets that ever attain a perfect conjunction, is $\mathcal{NP}$–complete a problem.

6.1.3 Proofs

The major ingredient is the following tool concerning the computational complexity of problems about rational arithmetic progressions:

Definition 23. *For $u,v \in \mathbb{Q}$, let $u \div v := (a \div b)/q$ and $\gcd(u,v) := \gcd(a,b)/q$ where $a,b,q \in \mathbb{Z}$ are such that $u = a/q$ and $v = b/q$ and $1 = \gcd(a,b,c)$; similarly for $u \operatorname{rem} v$ and $\operatorname{lcm}(u,v)$.*

For $a,\alpha \in \mathbb{Q}$, write $P_{a,\alpha} := \{\alpha + a \cdot z : z \in \mathbb{Z}\}$.

[‡‡] The attentative reader will connive our relaxed attitude concerning decision versus function problems

Lemma 24. *a) Given $a,\alpha\in\mathbb{Q}$, the unique $0\le\alpha'<a$ with $P_{a,\alpha}=P_{a,\alpha'}$ can be calculated as $\alpha':=\alpha\operatorname{rem}a$ within complexity class $\mathcal{NC}^1$.*

b) Given a,α and b,β, the question whether $P_{a,\alpha}\cap P_{b,\beta}=\emptyset$ can be decided in $\mathcal{NC}^{\text{gcd}}$

c) and, if so, (c,γ) with $P_{a,\alpha}\cap P_{b,\beta}=P_{c,\gamma}$ can be calculated in $\mathcal{NC}^{\text{gcdex}}$.

d) Items b) and c) extend from the intersection of two given arithmetic progressions to that of k.

e) Given n,k and $a_1,\alpha_1,\ldots,a_n,\alpha_n$, deciding whether some k among the arithmetic progressions $P^{(i_1)}:=P_{a_{i_1},\alpha_{i_1}},\ldots,P^{(i_k)}:=P_{a_{i_k},\alpha_{i_k}}$ have a non-empty common intersection, is $\mathcal{NP}$–complete.

A result similar to the last item has been obtained in [MaHa94]...

Proof. a) Notice that $P_{a,\alpha}=P_{a,\alpha'}\Leftrightarrow\alpha-\alpha'\in P_{a,0}$. Hence there exists exactly one such α' in $[0,a)$, namely $\alpha'=\alpha\operatorname{rem}a$. Moreover, integer division belongs to $\mathcal{NC}$ [BCH86,CDL01].

b) Observe that $P_{a,\alpha}\cap P_{b,\beta}\neq\emptyset$ holds iff $\gcd(a,b)$ divides $\alpha-\beta$. Indeed, the extended Euclidean algorithm then yields $z_1',z_2'\in\mathbb{Z}$ with $\gcd(a,b)=-a\cdot z_1'+b\cdot z_2'$; then $\alpha-\beta=-a\cdot z_1+b\cdot z_2$ yields $P_{a,\alpha}\ni\alpha+a\cdot z_1=\beta+b\cdot z_2\in P_{b,\beta}$. Conversely $\alpha+a\cdot z_1=\beta+b\cdot z_2\in P_{a,\alpha}\cap P_{b,\beta}$ implies that $\alpha-\beta=-a\cdot z_1+b\cdot z_2$ is a multiple of any (and in particular the greatest) common divisor of a and b.

c) Notice that $c=\operatorname{lcm}(a,b)=a\cdot b/\gcd(a,b)$; according to the proof of b), $\gamma:=\alpha+a\cdot z_1$ will do, where $z_1,z_2\in\mathbb{Z}$ with $\alpha-\beta=-a\cdot z_1+b\cdot z_2$ result from the extended Euclidean algorithm applied to (a,b).

d) Notice that

$$x\in P_{a_1,\alpha_1}\cap\cdots\cap P_{a_k,\alpha_k}\quad\Leftrightarrow\quad x\equiv\alpha_i\pmod{a_i},\quad i=1,\ldots,k\ .\tag{2}$$

According to the Chinese Remainder Theorem, the latter system of congruences in turn admits such a solution x iff $\gcd(a_i,a_j)$ divides $\alpha_i-\alpha_j$ for all pairs (i,j).

In order to calculate such an x, notice that a straight-forward iterative $P_{a_{1..k-1},\alpha_{1..k-1}}\cap P_{a_k,\alpha_k}$ fails as it does not parallelize well, and also the numbers calculated according to c) may double in length in each of the k steps. Instead, combine the P_{a_i,α_i} in a binary way first two tuples $P_{a_{2j,2j+1},\alpha_{2j,2j+1}}$ of adjacent ones, then on to quadruples and so on. At logarithmic depth (=parallel time), this yields the desired result $x=:\alpha_0$ and $a_0:=\operatorname{lcm}(a_1,\ldots,a_k)$ satisfying $P_{a_0,\alpha_0}=\bigcap_{i=1}^k P_{a_i,\alpha_i}$.

e) It is easy to guess $i_1,\ldots,i_k$ and, based on d), verify in polynomial time that $P^{(i_1)}\cap\ldots\cap P^{(i_k)}\neq\emptyset$.

We establish $\mathcal{NP}$–hardness by reduction from Clique [GaJo79]: Given a graph $G=([n],E)$, choose $n\cdot(n-1)/2$ pairwise coprime integers $q_{i,\ell}=q_{\ell,i}\ge 2$, $1\le i<\ell\le n$; for instance $q_{i,\ell}:=p_{i+n\cdot(\ell-1)}$ will do, where p_m denotes the m-th prime number, found in time polynomial in $n\le|\langle G\rangle|$ (though *not* in $|\langle p_m\rangle|\approx\log m+\log\log m$) by simple exhaustive search. Then calculate $a_i:=\prod_{\ell\neq i}q_{i,\ell}$ and observe that $\gcd(a_i,a_j)=q_{i,j}$ for $i\neq j$. Now start with $\alpha_1:=0$ and iteratively for $\ell=2,3,\ldots,n$ determine α_ℓ by solving the following system of simultaneous congruences:

$$\alpha_\ell\equiv\begin{cases}\alpha_i & \pmod{q_{i,\ell}}\ \text{ for }\ (i,\ell)\in E\\ 1+\alpha_i & \pmod{q_{i,\ell}}\ \text{ for }\ (i,\ell)\notin E\end{cases}\quad,\qquad 1\le i<\ell\tag{3}$$

Indeed, as the $q_{i,\ell}$ are pairwise coprime, the Chinese Remainder Theorem asserts the existence of a solution—computable in time polynomial in n, regarding that α_ℓ can be bounded by $\prod_{i,j}q_{i,j}$ having a polynomial number of bits). The thus constructed vector $(\alpha_i)_i$ satisfies:

$$\alpha_i\equiv\alpha_j\pmod{\underbrace{\gcd(a_i,a_j)}_{=q_{i,j}}}\quad\Leftrightarrow\quad(i,j)\in E$$

because, for $(i,j)\notin E$, Equation (3) implies $\alpha_i\equiv\alpha_j+\mathbf{1}\pmod{q_{i,j}}$.

We claim that this mapping $G\mapsto(a_i,\alpha_i:1\le i\le n)$ constitutes the desired reduction: Indeed, according to Equation (2), any sub-collection $P^{(i_1)},\ldots,P^{(i_k)}$ has non-empty intersection (i.e. a common element x) iff $\alpha_{i_\ell}\equiv\alpha_{i_j}\pmod{\gcd(a_{i_\ell},a_{i_j})}$, i.e., by our construction, iff $(i_\ell,i_j)\in E$; hence, cliques of G are in one-to-one correspondence with subcollections of intersecting arithmetic progressions. □

Proof (Theorem 22). At time t, planet #i appears at angular position $u_i+t\cdot v_i\mod 1$; and an exact conjunction between #i and #j occurs whenever $u_i+t\cdot v_i=u_j+t\cdot v_j+z$ for some $z\in\mathbb{Z}$, that is iff

$$t\ \in\ \left\{\frac{u_j-u_i}{v_i-v_j}+z\cdot\frac{1}{v_i-v_j}\right\}\ =\ P^{(i,j)}:=P_{a_{i,j},\alpha_{i,j}}\quad\text{where }a_{i,j}:=\frac{1}{v_i-v_j},\alpha_{i,j}:=\frac{u_j-u_i}{v_i-v_j}\ .\tag{4}$$

Therefore, planets #1,...,#n attain a conjunction at some time t iff $t \in \bigcap_{i=1}^{n} P^{(1,i)}$. The existence of such t thus amounts to the non-emptiness of the joint intersection of arithmetic progressions and can be decided in the claimed complexity according to Lemma 24b+d). Moreover, Lemma 24a+c+d) shows how to calculate the smallest t.

Concerning $\mathcal{NP}$–hardness claimed in Item f), we reduce from Lemma 24e): Given n arithmetic progressions $P^{(i)} = P_{a_i,\alpha_i}$, let $u_i := -\alpha_i \cdot a_i$, $v_i := 1/a_i$, and $u_0 := 0 =: v_0$. Then conjunctions between #0 and #i occur exactly at times $t \in P^{(i)}$; and $P^{(i_1)},\ldots,P^{(i_k)}$ meet iff (and when/where) #0,#$i_1,\ldots,$#i_k do.

Approximate conjunction up to ε in time interval (u,v) means:

$$\exists t \in (u,v)\ \exists z \in \mathbb{Z}:\ (v_2 - v_1)t + u_2 - u_1 + z \in (-\varepsilon, +\varepsilon)$$

which is equivalent to Claim b). The boundaries of the interval $(a,b)\cdot(v_1 - v_2) + u_1 - u_2 + (-\varepsilon,+\varepsilon)$ can be calculated in $\mathcal{NC}^1$. □

6.1.4 General Eudoxus/Aristotle; Ptolemy, Copernicus, and Kepler

Proceeding from the restricted 2D theory Φ' to Eudoxus/Aristotle's full Φ obviously complicates the computational complexity of the above predictions; and it seems desirable to make that precise, e.g. with the help of [ACG93,GaOv95,BrKi98,CPPY06]. Moreover also Φ in turn had been refined: PTOLEMY introduced additional so-called *epicycles* and *deferents* located and rotating on the originally earth-centered spheres. That extension (and its additional free parameters) allowed him to describe the observed planetary motions more accurately. Copernicus relocated the spheres (and sub-spheres thereon) to be centered around the sun rather than earth. And Kepler replaced them with ellipses in space. Again, the respective increase in complexity is worth-while investigating:

From an observer's perspective located on earth, the planets' 3D movements project down to curves. So, we are led to study the complexity of deciding whether and when such given curves, parameterized by time, intersect. A first step is to decide (mathematically) whether these curves are of an algebraic or transcendental nature, in order to choose the appropriate model of computation (Section 5.2). We point out that even for problems which naturally involve transcendental numbers restricting the input to rationals (and handling them in the original Turing model), may turn out to admit surprising algorithmic treatment; see, e.g., [CCK*04] or [Zieg06, PROPOSITION 30]!

6.2 Opticks

There is an abundance of (physical theories giving) explanations for optical phenomena; cmp. e.g. *The Book of Optics* by IBN AL-HAYTHAM (1021) or NEWTON's book providing the title of this section. We are specifically interested in the progression from geometric via Gaussian optics (taking into account dispersion) over HUYGENS and FOURIER (i.e. diffractive, wave) optics to Maxwell's theory of electromagnetism and even to quantum and quantum field theories (describing e.g. Raman/Rayleigh and other kinds of scattering). Note that this sequence of optical theories Φ_i reflects their historical succession, but *not* a logical one in the sense that Φ_{i+1} 'implies' (and hence is computationally at least as hard as) Φ_i.

Our purpose is to explore more thoroughly the computational complexities of these theories. Their computational relations may happen to be similar, unrelated, or just opposite to their historical ones! Consider for example geometric optics versus Electrodynamics:

6.2.1 Geometric Optics considers light rays as ideal geometric objects, i.e., of infinitesimal section proceeding instantaneously and straightly until hitting a, say, mirror. Now depending on the kind of mirrors (straight or curved, with rational or algebraic parameters) and the availability of further optical devices (lenses, beam splitters), [RTY94] has developed a fairly exhaustive (although not entirely sound) taxonomy of the induced computational complexities of ray tracing ranging from $\mathcal{PSPACE}$ to *un*decidable!

6.2.2 Electrodynamics on the other hand treats light as a vector-valued wave obeying a system of linear partial differential equations named after JAMES CLERK MAXWELL. From given initial conditions, their solution is computable even over real numbers [WeZh99]!

6.3 Quantum Mechanics

The advent of Quantum Computing is based on the hope to surpass the strong CTH, i.e. to build computers *not* polynomial-time equivalent to a Turing machine. It originates back to RICHARD P. FEYNMAN's famous Lectures on Computation [FLS05] and has, in connection with the work of PETER SHOR's, become a fashionable yet speculative [Kie03a] research topic lacking a general picture [Myrv95,Smi99a,Breu02,Zieg05,WeZh06,Smi06b].

Speaking in complexity theoretic terms, the (as usual highly ambiguous) question raised by the strong CTH (recall Section 1.2) asks to locate the computational power of QM somewhere among or between $\mathcal{P}$, $\mathcal{P}^{\text{IntegerFactorization}}$, $\mathcal{NP}$, and Δ_2. Further, it is worth-while to explore how the answer depends on the underlying Hamiltonians being un-/bounded as indicated in [PERi89, CHAPTER 3]?

For a sound and more definite investigation, our approach suggests to start exploring well-specified sub- and pre-theories of QM. These may, e.g., be the BOHR-SOMMERFELD theory of classical electron orbits with integral action-angle conditions.

Another promising direction considers computational capabilities and complexity of Quantum Logic:

6.3.1 Quantum Logic arises as an abstraction of the purely algebraic structure exhibited by the collection of *effect* operators (i.e. projections onto closed subspaces) introduced by G. LUDWIG on a Hilbert space, that are certain quantum mechanical observables; cf. e.g. [Birk67,Svoz98]. This discipline has flourished from the comparison with (i.e. systematic and thorough investigation of similarities and differences to) Boolean logic:

For a finite-dimensional Hilbert space $\mathcal{H} \in \{\mathbb{R}^N, \mathbb{C}^N\}$, variables $\mathbf{a}, \mathbf{b}, \mathbf{c}$ denote any linear subspace of $\mathcal{H}$, including $\mathbf{0} := \{0\}$ and $\mathbf{1} := \mathcal{H}$ itself; moreover let $\mathbf{a} \wedge \mathbf{b} := \mathbf{a} \cap \mathbf{b}$, $\neg \mathbf{a} := \mathbf{a}^{\perp}$, $\mathbf{a} \leq \mathbf{b} :\Leftrightarrow \mathbf{a} \subseteq \mathbf{b}$, and $\mathbf{a} \vee \mathbf{b} := \text{linspan}(\mathbf{a} \cup \mathbf{b})$. These operations indeed extend the classical Boolean connectives in that the latter are recovered by restricting to $\mathbf{0}$ and $\mathbf{1}$, i.e. to $N = 1$. The collection of linear subspaces of $\mathcal{H}$, equipped with $(\vee, \wedge, \neg, \leq)$, forms a lattice; which however violates distributivity: Formula

$$(\mathbf{a} \vee \mathbf{b}) \wedge (\mathbf{a} \vee \mathbf{b}^{\perp}) \wedge (\mathbf{a}^{\perp} \vee \mathbf{b}) \wedge (\mathbf{a}^{\perp} \vee \mathbf{b}^{\perp}) = \mathbf{1} \tag{5}$$

is not classically satisfiable but does, for $N \geq 2$ admit satisfying assignment $\mathbf{a} := \text{linspan}\binom{1}{0}$ and $\mathbf{b} := \text{linspan}\binom{1}{1}$. Nevertheless it has recently been proved that the satisfiability problem for such quantum logic formula is decidable [DHMW05, SECTION 3] and raised the question on the algorithmic complexity of this problem. We make the following

Proposition 25. *For $N \in \mathbb{N}$, let*

$$\text{QL}_N := \big\{ \langle \varphi \rangle : \textit{formula } \varphi \textit{ in } m \textit{ variables} \\ \textit{admits subspaces } \mathbf{a}_1, \ldots, \mathbf{a}_m \subseteq \mathbb{C}^N \textit{ such that } \varphi(\mathbf{a}_1, \ldots, \mathbf{a}_m) \neq \mathbf{0} \big\} . \tag{6}$$

a) QL_1 *is $\mathcal{NP}$–complete,*
b) *For each $N \leq M$, $\text{QL}_N \preccurlyeq \text{QL}_M$. In particular, QL_M is $\mathcal{NP}$–hard.*
c) *For fixed N (or with N given unary), $\text{QL}_N \in \mathcal{NP}_{\mathbb{C}}$.*

That is, QL_N can be decided by a nondeterministic BSS machine in time polynomial in N and m; recall Section 5.2 (and cmp. also [CuGr97] *for BSS computation on binary inputs).*

Proof. Item a) is of course the famous Cook-Levin Theorem. Item b) follows from $\text{QL}_N \subseteq \text{QL}_M$ [DHMW05, LEMMA 5]. Concerning Item c), a BSS machine may first 'guess' dimensions $0 \leq d_1, \ldots, d_m \leq N$ of, and orthogonal bases for, subspaces $\mathbf{a}_1, \ldots, \mathbf{a}_m$; then use linear algebra to obtain an orthogonal basis for the value $\varphi(\mathbf{a}_1, \ldots, \mathbf{a}_m)$. □

More generally, we envision a theory of computational complexity similar to that of Boolean circuits [Papa94, SECTIONS 4.3 and 11.4] with classical gates replaced by such quantum *logic* ones (cmp. [Ying05] for quantum logic finite automata). Note that this is different from (well-studied) 'standard' quantum complexity theory and the quantum circuit model it is based on:

- Quantum gates map $\mathcal{H}$ to $\mathcal{H}$ whereas operations "$\vee$", "$\wedge$", and "$\neg$" map (pairs of) subspaces of $\mathcal{H}$ to subspaces.
- Closed subspaces correspond to quantum 0/1-observables, i.e. to measurements; whereas quantum circuits are generally required to be reversible, forcing any measurement to be delayed to the end of the computation.
- A quantum circuit maps pure n-qubit states (i.e. in $N = 2^n$-dimensional Hilbert space) to mixed states; whereas the result of a measurement is always a pure state.
- Quantum parallelism usually refers to the simultaneous operation on (any linear combination of) all qubit states; whereas parallelism in a quantum logic circuit enters as in the Boolean case with notions like 'width' and 'depth'.
- It seems that *in*equality of two quantum logic expressions "$X(\mathbf{a},\mathbf{b},\mathbf{c}) \neq Y(\mathbf{a},\mathbf{b},\mathbf{c})$" cannot be expressed using equality and quantum logic connectives alone; cmp. also Question 26b) below. That is, additional Boolean negation (and/or quantifiers) may be needed here.

We find that the last item makes a further exploration of the expressive power of quantum logic worthwhile; and the first items suggest quantum logic circuits as an alternative (and hopefully more tractable) model for complexity theoretic explorations in quantum mechanics. To begin with, we ask

Question 26. a) Which m-variate functions on the lattice of linear subspaces of $\mathbb{C}^N$ can be realized as quantum logic expressions?
b) Does Proposition 25b) also hold with "$\neq \mathbf{0}$" in Equation (6) replaced by "$= \mathbf{1}$"?
c) If a quantum logic formula admits a satisfying assignment in $\mathbb{C}^N$, does it also admit one in $\mathbb{Q}^N$?

Concerning a) recall that, in the Boolean case $N = 1$, the answer is of course: all $F(m,1) = 2^{2^m}$ of them. In the two-variate non-Boolean case $N > 1$ on the other hand it holds $F(2,N) = 96$: the free orthomodular lattice on two generators a,b is known to have 96 elements $x(a,b)$ [Kalm83] which can be achieved to differ $x(\mathbf{a},\mathbf{b}) \neq y(\mathbf{a},\mathbf{b})$ for appropriate linear subspaces $\mathbf{a},\mathbf{b}$ of $\mathbb{Q}^2$.

7 Conclusion

Although gradually increasing in speed according to Moore's Law, all digital computers built since the last 60 years remain basically still instances of (and in particular polynomial-time equivalent to) the very same machine model that Alan Turing had introduced already in 1937.

> *Is this due to our lack of ideas for fundamentally new approaches to computing? Or have we already reached the ultimate computing capabilities of nature? Why does nature admit (i.e. is rich enough to provide us with the physical means to realize) universal computation, anyway?*

In the present work, we have proposed an approach towards answering these questions: Start off modestly by considering the CTH relative to some simple physical theories Φ; then gradually work the way up to more complex ones. This approach permits to systematically explore the boundary between computable and incomputable physical theories; and similarly for the boundaries of completeness for various complexity classes.

References

[ACG93] M.J. Atallah, P. Callahan, M.T. Goodrich: "$\mathcal{P}$–Complete Geometric Problems", pp.443–462 in *Int. J. Comput. Geometry Appl.* vol.**3:4** (1993).

[ACP04] V.A. Adamyan, C.S. Calude, B.S. Pavlov: "Transcending the limits of Turing computability", pp.119–137 in *Quantum Information Complexity. Proc. Meijo Winter School 2003* (T. Hida, K. Saito, S. Si Editors), World Scientific Singapore (2004).

[Batt07] R. Batterman: "Intertheory Relations in Physics", in *Stanford Encyclopedia of Philosophy*, `http://plato.stanford.edu/entries/physics-interrelate` (2007).

[BCH86] P. Beame, S. Cook, H. Hoover: "Log Depth Circuits for Division and Related Problems", pp.994–1003 in *SIAM Journal of Computing* vol.**15** (1986).

[BeLa85] C.H. Bennett, R. Landauer: "The Fundamental Physical Limits of Computation", pp.48–56 in *Scientific American*, vol.**253(1)** (1985).

[Benn95] C.H. Bennet: "Universal Computation and Physical Dynamics", pp.268–273 in *Physica D* vol.**86** (1995).

[BeTu04] E.J. Beggs, J.V. Tucker: "Computations via Experiments with Kinematic Systems", *Technical Report* 5–2004, Department of Computer Science, University of Wales Swansea.

[BeTu07] E.J. Beggs, J.V. Tucker: "Experimental Computation of Real Numbers by Newtonian Machines", pp.1541–1561 in *Proc. Royal Society* vol.**463** (2007).

[BGS75] T. Baker, J. Gill, R. Solovay: "Relativizations of the $\mathcal{P} = ?\mathcal{NP}$ Question", pp.431–442 in *SIAM Journal on Computing* vol.**4:4** (1975).

[BiBr85] E. Bishop, D.S. Bridges: "*Constructive Analysis*", Springer (1985).

[Birk67] G. Birkhoff: "Lattice Theory", vol.**XXV** in *Amer. Math. Soc. Colloq. Publ.*, Providence (1967).

[Boku08] A. Bokulich: "*Reexamining the Quantum-Classical Relation*", Cambridge (2008).

[Breu02] T. Breuer: "*Quantenmechanik: Ein Fall für Gödel?*", Spektrum (2002).

[BrKi98] H. Breu, D.G. Kirkpatrick: "Unit Disk Graph Recognition is $\mathcal{NP}$-hard", pp.3–24 in *Computational Geometry Theory and Applications* vol.**9** (1998).

[BrSv00] D. Bridges, K. Svozil: "Constructive Mathematics and Quantum Physics", pp.503–515 in *International Journal of Theoretical Physics* vol.**39:3** (2000).

[BCSS98] L. Blum, F. Cucker, M. Shub, S. Smale: "*Complexity and Real Computation*", Springer (1998).

[BSS89] 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).

[CCK*04] E.-C. Chang, S.W. Choi, D.Y. Kwon, H.J. Park, C.K. Yap: "Shortest Path amidst Disc Obstacles is Computable", pp.567–590 in *Int. J. Comput. Geometry Appl* vol.**16** (2006).

[CPPY06] S.W. Choi, S.I. Pae, H.J. Park, C.K. Yap: "Decidability of Collision between a Helical Motion and an Algebraic Motion", pp.69–82 in (G. Hanrot and P. Zimmermann Eds.) *Proc. 7th Conference on Real Numbers and Computers* (RNC7), LORIA, Nancy, France (2006).

[CDL01] A. Chiu, G.I. Davida, B.E. Litow: "Division in logspace-uniform $\mathcal{NC}^1$", pp.259–275 in *Informatique Théoretique et Applications* vol.**35:3** (2001).

[CDS00] C.S. Calude, M.J. Dinneen, K. Svozil: "Reflections on Quantum Computing", pp.35–37 in *Complexity* vol.**6:1** (2000).

[CDCG95] G. Cattaneo, M.L. Dalla Chiara, R. Giuntini: "Constructivism and Operationalism in the Foundations of Quantum Mechanics", pp.21–31 in [DSKS95].

[Clar68] A.C. Clarke: "*2001: A Space Odyssey*", (1968).

[Cope02] J. Copeland: "Hypercomputation", pp.461–502 in *Minds and Machines* vol.**12**, Kluwer (2002).

[CuGr97] F. Cucker, D. Grogoriev: "On the Power of Real Turing Machines over Binary Inputs", pp.243–254 in *SIAM J. Comput* vol.**26:1** (1997).

[Davi01] E.B. Davies: "Building Infinite Machines", pp.671–682 in *Brit. J. Phil. Sci.* vol.**52** (2001).

[Deut85] D. Deutsch: "Quantum theory, the Church-Turing principle and the universal quantum computer", pp.97–117 in *Proc. Royal Society of London A* vol.**400** (1985).

[DSKS95] W. DePauli-Schimanovich, E. Köhler, F. Stadler (Eds.): "The Foundational Debate: Complexity and Constructivity in Mathematics and Physics", *Vienna Circle Institute Yearbook*, Kluwer (1995).

[Duhe85] P. Duhem: "*Le système dy monde. Histoire des doctrines cosmologiques de Platon á Copernic*", Paris (1913); partial English translation by R. Ariew: "*Medieval Cosmology: Theories of Infinity, Place, Time, Voide, and the Plurality of Worlds*", University of Chicago Press (1985).

[Duhe54] P. Duhem: "*La théorie physique, son objet, sa structure*", Paris: Cevalier & Rivière (1906); English translation by P.P. Wiener: "*The Aim and Structure of Physical Theory*", Princeton University Press (1954).

[DHMW05] J.M. Dunn, T.J. Hagge, L.S. Moss, Z. Wang: "Quantum Logic as Motivated by Quantum Computing", pp.353–359 in *Journal of Symbolic Logic* vol.**70:2** (2005).

[EtNe02] G. Etesi, I. Németi: "Non-Turing Computations Via Malament-Hogarth Space-Times", pp.341–370 in *International Journal of Theoretical Physics* vol.**41:2** (2002).

[Flet02] P. Fletcher: "A Constructivist Perspective on Physics", pp.26–42 in *Philosophia Mathematica* vol.**10** (2002).

[FLS05] R.P. Feynman, R.B. Leighton, M. Sands: "*The Feynman Lectures on Computation including Feynman's Tips on Physics*" (2nd Edition), Addison Wesley (2005).

[FLT82] E. Fredkin, R. Landauer, T. Toffoli (Edts): *Proc. Conf. Physics of Computation*:
"Part I: Physics of Computation", *International Journal of Theoretical Physics* vol.**21** Nos.3+4 (1982)
"Part II: Computational Models of Physics", *Int. Journal of Theoretical Physics* vol.**21** Nos.6+7 (1982)
"Part III: Physical Models of Computation", *Int. Journal of Theoretical Physics* vol.**21** No.12 (1982).

[Fran02] M.P. Frank: "The Physical Limits of Computing", pp.16–26 in *IEEE Computing in Science and Engineering* vol.**4:3** (2002).

[Frie84] H. Friedman: "The Computational Complexity of Maximization and Integration", pp.80–98 in *Advances in Mathematics* vol.**53** (1984).

[FrTo82] E. Fredkin, T. Toffoli: "Conservative Logic", pp.219–253 in *International Journal of Theoretical Physics* vol.**21** (1982).

[Galt06] A. Galton: "The Church-Turing thesis: Still valid after all these years?", pp.93–102 in *Applied Mathematics and Computation* vol.**178** (2006).

[Gand80] R. Gandy: "Church's Thesis and Principles for Mechanisms", pp.123–148 in *The Kleene Symposium* (J. Barwise, H.J. Keisler, and K. Kunen, editors), North-Holland (1980).

[GaOv95] A. Gajentaan, M.H. Overmars: "On a Class of $O(n^2)$ Problems in Computational Geometry", pp.165–185 in *Comput. Geom.* vol.**5** (1995).

[GaJo79] M.R. Garey, D.S. Johnson: "*Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$–Completeness*", Freeman (1979).

[GeHa86] R. Geroch, J.B. Hartle: "Computability and Physical Theories", pp.533–550 in *Foundations of Physics* vol.**16(6)** (1986).

[GHR95] R. Greenslaw, H.J. Hoover, W.L. Ruzzo: "*Limits to Parallel Computation*", Oxford University Press (1995).

[GoWe08] D. Goldin, P. Wegner: "The Interactive Nature of Computing", pp.17–38 in *Minds and Machines* vol.**18:1** (2008).

[Grze57] A. Grzegorczyk: "On the Definitions of Computable Real Continuous Functions", pp.61–77 in *Fundamenta Mathematicae* **44** (1957).

[HaHa83] S.W. Hawking, J.B. Hartle: "The Wave Function of the Universe", pp.2960–2975 in *Physical Review D* vol.**28** (1983).

[Haet96] F. Hättich: "Zur Axiomatik der Quantenmechanik", master's thesis, University of Paderborn, Dept. of Theoretical Physics (1996);
`http://www.upb.de/cs/ag-madh/WWW/ziegler/frank.diplom.pdf`

[Hage82] N. Hager: "Modelle in der Physik", vol.**278** in *Wissenschaftliche Taschenbücher*, Akademie-Verlag Berlin (1982).

[Hagg07] Hagge: "$QL(\mathbb{C}^n)$ Determines n", pp.1194–1196 in *Journal of Symbolic Logic* vol.**72:4** (2007).

[Hodg06] A. Hodges: "Did Church and Turing Have a Thesis about Machines?", pp.242–252 in (A. Olszewski, J. Wolenski, and R. Janusz, editors) *ChurchŠs Thesis after 70 Years*, Ontos Verlag (2006).

[Holl96] P.R. Holland: "Is Quantum Mechanics Universal?", pp.99–110 in (J.T. Cushing, A. Fine, S. Goldstein, Eds) *Bohmian Mechanics and Quantum Theory: An Appraisal*, Kluwer (1996).

[Kalm83] G. Kalmbach: "*Orthomodular Lattices*", Academic Press (1983).

[Kie03a] T. Kieu: "Quantum Algorithm for Hilbert's Tenth Problem", pp.1461–1478 in *International Journal of Theoretical Physics* **42**, Springer (2003).

[Kie03b] T.D. Kieu: "Computing the non-computable", pp.51–71 in *Contemporary Physics* vol.**44(1)** (2003).

[Klee52] S. Kleene: "*Introduction to Metamathematics*", Van Nostrand (1952).

[Ko91] K.-I. Ko: "*Computational Complexity of Real Functions*", Birkhäuser (1991).

[Krei87] G. Kreisel: "Church's Thesis and the Ideal of Informal Rigour", pp.499–519 in *Notre Dame Journal of Formal Logic* vol.**28:4** (1987).

[Kuhn62] T.S. Kuhn: "*The Structure of Scientific Revolutions*", University of Chicago Press (1962).

[Kush84] B.A. Kushner: "Lectures on Constructive Mathematical Analysis", vol.**60** in *Translations of Mathematical Monographcs*, AMS (1984);
translated from Russian (Moscow 1973) and edited by E. Mendelson and L.J. Leifman.

[Lloy02] S. Lloyd: "Ultimate Physical Limits to Computation", pp.1047–1054 in *Nature* vol.**406** (Aug.2000).

[Lloy06] S. LLOYD: "*Programming the Universe*", Random House (2006).

[LoCo08] B. LOFF, J.F. COSTA: "Five Views of Hypercomputation", to appear in *International Journal of Unconventional Computing*.

[Loff07] B. LOFF: "*Physics, Computation and Definability*", Master's Thesis, Mathematics Department of the Instituto Superior Técnico, Lisbon, Portugal (Nov.2007).

[Ludw85] G. LUDWIG: "*An Axiomatic Basis for Quantum Mechanics*", Springer (1985).

[Ludw90] G. LUDWIG: "*Die Grundstrukturen einer physikalischen Theorie*", Springer (1990).

[MaHa94] B.S. MAJEWSKI, G. HAVAS: "The Complexity of Greatest Common Divisor Computations", pp.184–193 in *Proc. 1st International Symposium on Algorithmic Number Theory*, Springer LNCS vol.**877** (1994).

[Matz92] D. MATZKE (Chairman): "*IEEE Proc. Workshop on Physics and Computation*" (Texas 1992), `http://ieeexplore.ieee.org/xpl/RecentCon.jsp?punumber=4863`;
see also `http://ieeexplore.ieee.org/xpl/RecentCon.jsp?punumber=2948` (1994).

[Meis90] R. MEISTER: "A Structural Analysis of the Ehlers-Pirani-Schild Space-Time Theory", Master's Thesis, University of Paderborn, Dept. of Theoretical Physics (1990).

[MeMi97] K. MEER, C. MICHAUX: "A Survey on Real Structural Complexity Theory", pp.113–148 in *Bull. Belg. Math. Soc.* vol.**4** (1997).

[MeZi06] K. MEER, M. ZIEGLER: "Uncomputability Below the Real Halting Problem", pp.368–377 in *Proc. 2nd Conference on Computability in Europe* (CiE'06), Springer LNCS vol.**3988**.

[Miln97] J. MILNOR: "*Topology from the Differentiable Viewpoint*", Princeton University Press (1997).

[Mins68] M. MINSKY: "Matter, Mind and Models", pp.425–432 in *Semantic Information Processing*, MIT Press (1968).

[Mitt72] P. MITTELSTAEDT: "*Die Sprache der Physik*", B.I. Wissenschaftsverlag (1972).

[Moor90] C. MOORE: "Unpredictability and undecidability in dynamical systems", pp.2354–2357 in *Phys. Rev. Lett.* vol.**64** (1990).

[MTB*07] P. MARRONETTI, W. TICHY, B. BRÜGMANN, J.A. GONZALEZ, M. HANNAM, S. HUSA, U. SPERHAKE: "Binary black holes on a budget: Simulations using workstations", pp.43–58 in *Class.Quant.Grav.* vol.**24** (2007).

[Myrv95] W.C. MYRVOLD: "Computability in Quantum Mechanics", pp.33–46 in [DSKS95].

[Papa94] C.H. PAPADIMITRIOU: "*Computational Complexity*", Addison-Wesley (1994).

[PERi81] M.B. POUR-EL, J.I. RICHARDS: "The Wave Equation with Computable Initial Data such that its Unique Solution is not Computable", pp.215–239 in *Advances in Mathematics* vol.**39:4** (1981).

[PERi89] M.B. POUR-EL, J.I. RICHARDS: "*Computability in Analysis and Physics*", Springer (1989).

[PIM06] A. PERCUS, G. ISTRATE, C. MOORE: "*Computational Complexity and Statistical Physics*", Oxford University Press (2006).

[Ord02] T. ORD: "Hypercomputation: computing more than the Turing machine", Honour's Thesis, University of Melbourne (2002); `http://arxiv.org/math.LO/0209332`

[Pont07] J. PONTIN: "A Giant Leap Forward in Computing? Maybe Not", *The New York Times*, April 8 (2007).

[PrSh85] F.P. PREPARATA, M.I. SHAMOS: "*Computational Geometry: An Introduction*", Springer Monographs in Computer Science (1985).

[ReTa93] J. REIF, S.R. TATE: "The Complexity of N-Body Simulations", pp.162–176 in *Proc. 20th International Conference on Automata, Languages, and Programming* (ICALP'93), Springer LNCS vol.**700**.

[RTY94] J.H. REIF, J.D. TYGAR, A. YOSHIDA: "Computability and Complexity of Ray Tracing", pp.265–276 in *Discrete and Computational Geometry* vol.**11** (1994).

[Sche92] U. SCHELB: "An axiomatic Basis of space-time theory, Part III", pp.297–309 in *Reports on Mathematical Physics* vol.**31** (1992).

[Sch97a] U. SCHELB: "*Zur physikalischen Begründung der Raum-Zeit-Geometrie*", Habilitation thesis, University of Paderborn (1997);
`http://www.upb.de/cs/ag-madh/WWW/ziegler/SCHELB/hab1.ps.gz`

[Sch97b] E. SCHEIBE: "*Die Reduktion physikalischer Theorien*", Springer (1997).

[Schm08] H.-J. SCHMIDT: "Structuralism in Physics", in *Stanford Encyclopedia of Philosophy*, `http://plato.stanford.edu/entries/physics-structuralism/` (2008).

[Schr88] J. SCHRÖTER: "An axiomatic Basis of space-time theory, Part I", pp.303–333 in *Reports on Mathematical Physics* vol.**26** (1988).

[Schr96] J. SCHRÖTER: "*Zur Meta-Theorie der Physik*", de Gruyter (1996).

[ScSc92] J. SCHRÖTER, U. SCHELB: "An axiomatic Basis of space-time theory, Part II", pp.5–27 in *Reports on Mathematical Physics* vol.**31** (1992).

[Ship93] J. SHIPMAN: "Aspects of Computability in Physics", pp.299–314 in *Workshop on Physics and omputation* (IEEE Computer Society, Los Alamitos, NM, 1993).

[Sieg00] W. SIEG: "Calculations by Man and Machine" (2000).

[Smi99a] W.D. SMITH: “Church’s Thesis meets Quantum Mechanics”, pre-print (1999); `http://www.math.temple.edu/ wds/homepage/churchq.ps`

[Smi99b] W.D. SMITH: “History of «Church’s Theses» and a Manifesto on Converting Physics into a Rigorous Algorithmic Discipline”, pre-print (1999); `http://www.math.temple.edu/ wds/homepage/churchhist.ps`

[Smi06a] W.D. SMITH: “Church’s Thesis meets the *N*-Body Problem”, pp.154–183 in *Applied Mathematics and Computation* vol.**178** (2006).

[Smi06b] W.D. SMITH: “Three counterexamples refuting Kieu’s plan for «quantum adiabatic hypercomputation» and some uncomputable quantum mechanical tasks”, pp.184–193 in *J. Applied Mathematics and Computation* vol.**178** (2006).

[Snee71] J.D. SNEED: “*The Logical Structure of Mathematical Physics*”, Reidel Dordrecht (1971).

[Steg79] W. STEGMÜLLER: “*The Structuralist View of Theories*”, Springer (1979).

[Steg86] W. STEGMÜLLER: “Theorie und Erfahrung”, vol.**II** part.**F** in *Probleme und Resultate der Wissenschaftstheorie und Analytischen Philosophie*, Springer Studienausgabe (1986).

[Stoe95] M. STÖLZNER: “Levels of Physical Theories”, pp.47–64 in [DSKS95].

[StSt71] A. STRUGATSKY, B. STRUGATSKY: “*Roadside Picnic*” (1971).

[Svoz93] K. SVOZIL: “*Randomness and Undecidability in Physics*”, World Scientific (1993).

[Svoz95] K. SVOZIL: “A Constructivist Manifesto for the Physical Sciences – Constructive Re-Interpretation of Physical Undecidability”, pp.65–88 in [DSKS95].

[Svoz97] K. SVOZIL: “Linear Chaos Via Paradoxical Set Decompositions”, pp.785–793 in *Chaos, Solitons, and Fractals* vol.**7:5** (1996).

[Svoz98] K. SVOZIL: “*Quantum Logic*”, Springer (1998).

[Svoz05] K. SVOZIL: “Computational universes”, pp.845–859 in *Chaos, Solitons and Fractals* vol.**25:4** (2005).

[Svoz06] K. SVOZIL: “Physics and metaphysics look at computation”, pp.491–517 in *Church’s Thesis after 70 Years* (A. Olszewski, J. Wolenski, R. Janusz Edts.), Ontos (2006).

[Svoz07] K. SVOZIL: “Omega and the Time Evolution of the *n*-Body Problem”, pp.231–236 in *Randomness and Complexity, from Leibniz to Chaitin* (C.S. Calude Edt.), World Scientific (2007).

[Turi36] A.M. TURING: “On Computable Numbers, with an Application to the Entscheidungsproblem”, pp.230–265 in *Proc. London Math. Soc.* vol.**42(2)** (1936).

[TuZu00] J.V. TUCKER, J.I. ZUCKER: “Computable functions and semicomputable sets on many-sorted algebras”, pp.317–523 in *Handbook of Logic in Computer Science* vol.**5** (S. Abramsky, D.M. Gabbay, T.S.E. Maybaum Edts), Oxford Science Publications (2000).

[Yao03] 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).

[Wein94] S. WEINBERG: “*Dreams of a Final Theory: The Scientist’s Search for the Ultimate Laws of Nature*”, Vintage (1994).

[Weiz85] C.F. VON WEIZSÄCKER: “Das Gefüge der Theorien”, Kapitel **6** in *Aufbau der Physik*, Carl Hanser (1985).

[WeZh99] K. WEIHRAUCH, N. ZHONG: “The Wave Propagator Is Turing Computable”, pp.697–707 in *Proc. 26th International Colloquium on Automata, Languages and Programming* (ICALP 1999), Springer LNCS vol.**1644**.

[WeZh02] K. WEIHRAUCH, N. ZHONG: “Is wave propagation computable or can wave computers beat the turing machine?”, pp.312–332 in *Proc. London Mathematical Society* vol.**85:2** (2002).

[WeZh06] K. WEIHRAUCH, N. ZHONG: “Computing Schrödinger propagators on Type-2 Turing machines”, pp.918–935 in *Journal of Complexity* vol.**22:6** (2006).

[Wolf85] S. WOLFRAM: “Undecidability and Intractability in Theoretical Physics”, pp.735–738 in *Physical Review Letters* vol.**54:8** (1985).

[Xia92] Z. XIA: “The Existence of Noncollision Singularities in the *N*-Body Problem”, pp.411–468 in *Ann. Math.* vol.**135:3** (1992).

[Ying05] M. YING: “A Theory of Computation Based on Quantum Logic (I)”, pp.134–207 in *Theoretical Computer Science* vol.**344** (2005).

[Zieg05] M. ZIEGLER: “Computational Power of Infinite Quantum Parallelism”, pp.2057–2071 in *International Journal of Theoretical Physics* vol.**44:11** (2005).

[Zieg06] M. ZIEGLER: “Effectively Open Real Functions”, pp.827–849 in *Journal of Complexity* vol.**22** (2006).

[Zieg08] M. ZIEGLER: “(A Meta-Theory of) Physics and Computation”, p.145 in *Verhandlungen der Deutschen Physikalischen Gesellschaft* (DPG), Feb.2008.

[Zuse06] “Is the Universe a Computer? From Konrad Zuse’s Invention of the Computer to his «Calculating Space» to Quantum Computing”, Symposium under the patronage of Dr. Annette Schavan, Federal Minister of Education and Research (Nov. 2006). `http://www.gi-ev.de/regionalgruppen/berlin/download/veranstaltungen/ist_das_universum_flyer.pdf`

Author Index

Adamatzki, Andrew
Andréka, Hajnal
Beggs, Edwin
Boker, Udi
Bournez, Olivier
Brukner, Časlav
Buescu, Jorge
Campagnolo, Manuel Lameiras
Chassaing, Philippe
Cohen, Johanne
Cooper, S. Barry
da Costa, Newton C. A.
Delvenne, Jean-Charles
Dershowitz, Nachum
Dória, Francisco António
Durand-Lose, Jérôme
Gerin, Lucas
Gorecka, J. N.
Górecki, Jerzy
Graça, Daniel
Hogarth, Mark
Igarashi, Y.
Koegler, Xavier
Naughton, Thomas J.
Németi, István
Németi, Péter
Stannett, Mike
Svozil, Karl
Tkadlec, Josef
Thompson, B. C.
Tucker, John V.
Velupillai, K. Vela
Woods, Damien
Ziegler, Martin
Zucker, Jeffery