PROCEEDINGS OF THE SYMPOSIUM ON NATURAL/UNCONVENTIONAL COMPUTING AND IT'S PHILOSOPHICAL SIGNIFICANCE @ AISB/IACAP 2nd - 6th July 2012

Conference Paper · July 2012

CITATIONS
0

READS
50

2 authors, including:

Gordana Dodig Crnkovic
Chalmers University of Technology
124 PUBLICATIONS **768** CITATIONS

SEE PROFILE

AISB/IACAP World Congress 2012

Birmingham, UK, 2-6 July 2012

Natural Computing/ Unconventional Computing and its Philosophical Significance

Gordana Dodig-Crnkovic and Raffaela Giovagnoli

(Editors)

Part of

Published by
The Society for the Study of
Artificial Intelligence and
Simulation of Behaviour

http://www.aisb.org.uk

ISBN 978-1-908187-24-6

Foreword from the Congress Chairs

For the Turing year 2012, AISB (The Society for the Study of Artificial Intelligence and Simulation of Behaviour) and IACAP (The International Association for Computing and Philosophy) merged their annual symposia/conferences to form the AISB/IACAP World Congress. The congress took place 2–6 July 2012 at the University of Birmingham, UK.

The Congress was inspired by a desire to honour Alan Turing, and by the broad and deep significance of Turing's work to AI, the philosophical ramifications of computing, and philosophy and computing more generally. The Congress was one of the events forming the Alan Turing Year.

The Congress consisted mainly of a number of collocated Symposia on specific research areas, together with six invited Plenary Talks. All papers other than the Plenaries were given within Symposia. This format is perfect for encouraging new dialogue and collaboration both within and between research areas.

This volume forms the proceedings of one of the component symposia. We are most grateful to the organizers of the Symposium for their hard work in creating it, attracting papers, doing the necessary reviewing, defining an exciting programme for the symposium, and compiling this volume. We also thank them for their flexibility and patience concerning the complex matter of fitting all the symposia and other events into the Congress week.

John Barnden (Computer Science, University of Birmingham)
Programme Co-Chair and AISB Vice-Chair
Anthony Beavers (University of Evansville, Indiana, USA)
Programme Co-Chair and IACAP President
Manfred Kerber (Computer Science, University of Birmingham)
Local Arrangements Chair

Foreword from the Symposium Chairs

Even though Turing is best known for the Turing machine and the Turing test, his contribution is significantly wider. He was among the first to pursue what Denning [1] calls "computing as natural science", and thus Hodges [2] describes Turing as natural philosopher:

> "He thought and lived a generation ahead of his time, and yet the features of his thought that burst the boundaries of the 1940s are better described by the antique words: natural philosophy."

Rozenberg [3] makes similar observation about Turings genuine interest in nature seen through the prism of natural computing, as visible from his work on morphogenesis [4] and work on architecture of neural networks. Teuscher [5] illustrates by the example of Turing letter to Ross Ashby, where Turing explains the connection between the biological brain and its computational model:

> "The ACE is in fact, analogous to the "universal machine" described in my paper on computable numbers. This theoretical possibility is attainable in practice, in all reasonable cases, at worst at the expense of operating slightly slower than a machine specially designed for the purpose in question. Thus, although the brain may in fact operate by changing its neuron circuits by the growth of axons and dendrites, we could nevertheless make a model, within the ACE, in which this possibility is allowed for, but in which the actual construction of ACE did not alter, but only the remembered data, describing the mode of behaviour applicable at any time."
> `http://www.rossashby.info/letters/turing.html`

Today we are both learning about the structures and behaviours of natural systems by modelling them as information processing networks, as well as learning about possible ways of computation by constructing new computation models and machinery based on natural processes understood as computation. [7]

According to the Handbook of Natural Computing [6] natural computing is the field of research that investigates both human-designed computing inspired

by nature and computing taking place in nature. In particular, natural computing includes:

- Computational models inspired by natural systems such as neural computation, evolutionary computation, cellular automata, swarm intelligence, artificial immune systems, artificial life systems, membrane computing and amorphous computing.
- Computation performed by natural materials such as bioware in molecular computing or quantum-mechanical systems in case of quantum computing.
- Study of computational nature of processes taking place in (living) nature, such as: self-assembly, developmental processes, biochemical reactions, brain processes, bionetworks and cellular processes.

Variety of arguments for natural/unconventional computation, ranging from technical, logical, scientific-theoretical and philosophical are proposed by Cooper [12], MacLennan [13], Stepney [8], Burgin [14], Paun [15] and others. There are conferences and journals on natural computing, unconventional computing, organic computing, and similar related new fields of computing. A good overview on non-classical computation may be found in [**?**] and [9].

Hypercomputation or super-Turing computation denotes models of computation that go beyond Turing machines, presenting methods for computation of non-Turing-computable functions. The term "super-Turing computation" appeared in a 1995 paper by Siegelmann [10]. The term "hypercomputation" was coined in 1999 by Copeland and Proudfoot, [11].

The notion of representation is at the basis of a lively debate that crosses philosophy and artificial intelligence. This is because the comparison starts from the analysis of mental representations. Traditionally, philosophers use the notion of intentionality to describe the representational nature of mental states namely intentional states are those that represent something, because mind is directed toward objects.

The challenge for AI is therefore to approximate to human representations i.e. to the semantic content of human mental states. The task to consider the similarity between human and artificial representation could involve the risk of skepticism

about the possibility of computing this mental capacity. If we consider computationalism as defined in purely abstract syntactic terms then we are tempted to abandon it because human representation involves real world constrains.

But, a new view of computationalism could be introduced that takes into consideration the limits of the classical notion and aims at providing a concrete, embodied, interactive and intentional foundation for a more realistic theory of mind.

The symposium on Natural/unconventional computing and its philosophical significance connects the natural computation with hypercomputation and new understanding of representation which are compatible with each other, and show that there is mutual support between all those approaches.

References

[1] Denning P. J. (2007) Computing is a natural science, Commun. ACM, 50(7), pp. 13-18.

[2] Hodges, A. (1997) Turing. A Natural philosopher. Phoenix

[3] Rozenberg, G. (2008) Computer science, informatics and natural computing - personal reflections. In New Computational Paradigms: Changing Conceptions of What Is Computable, Springer, pp. 373-379.

[4] Turing, A.M. (1952) The chemical basis of morphogenesis, Philosophical Transactions of the Royal Society of London, B 237, pp. 37-72.

[5] Teuscher, C. (2002) Turing's Connectionism: An Investigation of Neural Networks Architectures, Springer Verlag, London.

[6] Rozenberg G., Bäck T., Kok J. N., eds. (2011) Handbook of Natural Computing, volume II. Springer.

[7] Kari, L. and Rozenberg, G. (2008) The many facets of natural computing, Commun. ACM, 51(10), pp. 72-83.

[8] Stepney S. et al. (2005). Journeys in non-classical computation I: A grand challenge for computing research. Int. J. Parallel, Emergent and Distributed Systems, 20(1):5-19.

[9] Stepney S. et al. (2006) Journeys in Non-Classical Computation II: Initial journeys and waypoints. Int. J. Parallel, Emergent and Distributed Systems. 21(2):97-125.

[10] Siegelmann H. (1999) Neural Networks and Analog Computation: Beyond the Turing Limit. Boston: Birkhöuser.

[11] Copeland, B. J. and Proudfoot, D. (1999) Alan Turing's forgotten ideas in computer science. Scientific American 280, 76-81.

[12] Cooper, S. B., Löwe, B., Sorbi, A. (2008) New Computational Paradigms. Changing Conceptions of What is Computable. Springer Mathematics of Computing series, XIII. (S. B. Cooper, B. Löwe, A. Sorbi, Eds.). Springer.

[13] MacLennan, B. (2011) Artificial Morphogenesis as an Example of Embodied Computation, International Journal of Unconventional Computing 7, 12 , pp. 3-23.

[14] Burgin, M. (2005) Super-Recursive Algorithms, Springer Monographs in Computer Science.

[15] Paun G. (2005) Bio-inspired Computing Paradigms (Natural Computing) in Unconventional Programming Paradigms. Lecture Notes in Computer Science, 2005, Volume 3566.

The accepted papers of the symposium address the following two main topics:

I) NATURAL/UNCONVENTIONAL COMPUTING which covers the emerging paradigm of natural computing, and its philosophical consequences with different aspects including theoretical and philosophical view of natural/ unconventional computing with its philosophical significance; characterizing the differences between conventional and unconventional computing; digital vs. analog and discrete vs. continuous computing; recent advances in natural computation (as computation found in nature, including organic computing; computation performed by natural materials; and computation inspired by nature); computation and its interpretation in a broader context of possible frameworks for modelling and implementing computation, and

II) REPRESENTATION AND COMPUTATIONALISM that highlights the relevance of the relationship between human representation and machine representation to analyse the main issues concerning the contrast between symbolic representations/processing on the one hand and nature-inspired, non-symbolic forms of computation on the other hand–with a special focus on connectionism, including work on hybrids of symbolic and non-symbolic representations. Particular developments addressed are the 'Embedded, Embodied, Enactive' approach to cognitive science (Varela et al); the 'Dynamic Systems' approach (by, say, Port and Van Gelder); Process/procedural representations (e.g. by O'Regan) and other representational possibilities that are clearly available: no representations or minimal representations;

All submitted papers have undergone a rigorous peer review process by our programme committee. We thank the reviewers for their constructive and thorough efforts. Finally we would like to thank the AISB/IACAP World Congress organizers for the support as well as our symposium participants for good collaboration.

Gordana Dodig-Crnkovic (Mälardalen University, Sweden)
Raffaela Giovagnoli (Pontifical Lateran University, Italy)

Symposium Organization

Program Committee

Barry S Cooper

Mark Burgin

Bruce MacLennan

Hector Zenil

Koichiro Matsuno

Vincent C. Müller

William A Phillips

Andree Ehresmann

Leslie Smith

Christopher D. Fiorillo

Plamen Simeonov

Marcin Schroeder

Brian Josephson

Shuichi Kato

Walter Riofrio

Craig A. Lindley

Jordi Vallverd

Angela Ales Bello

Gerard Jagers op Akkerhuis

Harold Boley

Cristophe Menant

Rossella Fabbrichesi

Giulio Chiribella

Jennifer Hudin

Philip Goyal

Klaus Mainzer

Chairs

Gordana Dodig-Crnkovic (Mälardalen University, Sweden)
Raffaela Giovagnoli (Pontifical Lateran University, Italy)

Contents

Unconventional Computer Programming

Susan Stepney[1]

Abstract. Classical computing has well-established formalisms for specifying, refining, composing, proving, and otherwise reasoning about computations. These formalisms have matured over the past 70 years or so.

Unconventional Computing includes the use novel kinds of substrates – from black holes and quantum effects, through to chemicals, biomolecules, even slime moulds – to perform computations that do not conform to the classical model. Although many of these substrates can be "tortured" to perform classical computation, this is not how they "naturally" compute.

Our ability to exploit unconventional computing is partly hampered by a lack of corresponding programming formalisms: we need models for building, composing, and reasoning about programs that execute in these substrates. What might, say, a slime mould programming language look like?

Here I outline some of the issues and properties of these unconventional substrates that need to be addressed to find "natural" approaches to programming them. Important concepts include embodied real values, processes and dynamical systems, generative systems and their meta-dynamics, and embodied self-reference.

1 Introduction

Let's look at the genesis of conventional computing. Turing formalised the behaviour of real world "computers" (human clerks carrying out calculations [11]) following a finite sequence of discrete, well-defined rules. This formalisation led to an abstract model: the Turing Machine (TM) [46].

Turning to the real world, there are many processes we might want to describe, understand, or exploit computationally: termites building complex nests following (relatively) simple rules; slime moulds growing in the topology of road networks; chemical oscillations set up to perform boolean operations. What are the relevant abstractions? Are they just the discrete TM again?

At this stage in the development of unconventional (or non-Turing) computation, I think that this is the wrong question. First, we should investigate these processes to discover what computations they perform "naturally". I would not trust a slime mould computer to spell-check my writing, or calculate my tax return. But certain forms of computers, for example analogue devices, can perform their computations much more "naturally" (for example, much more power-efficiently [16, p83]) than a digital version. Let's start from this point, discover what kinds of computation are natural to a range of systems, and then abstract from there.

We should not worry that our unconventional computers are ridiculously primitive compared to our smartphones: classical computation has seventy years of an exponentially growing lead on us.

[1] York Centre for Complex Systems Analysis, University of York, UK, email: susan.stepney@york.ac.uk

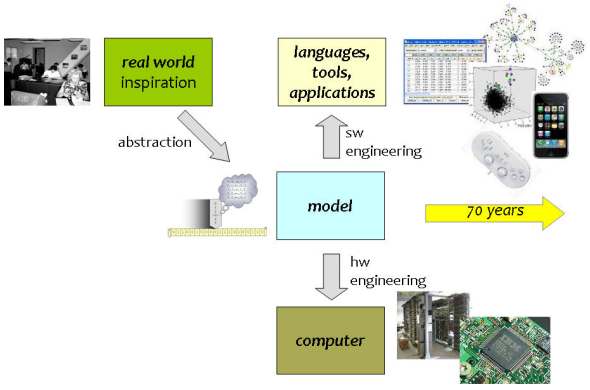


Figure 1. Classical computation: the real world inspiration of human computers led to an abstract model. This was realised in hardware and exploited in software, and developed for 70 years, into a form unrecognisable to its early developers.

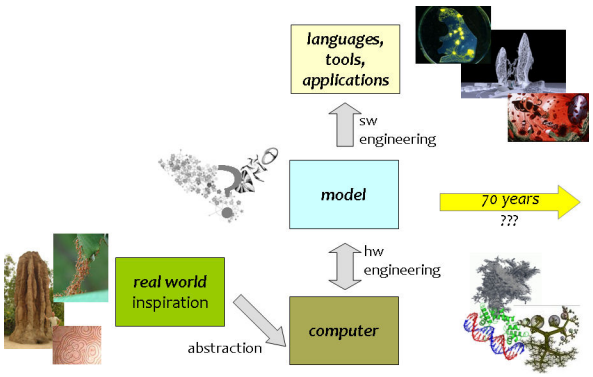


Figure 2. Unconventional computation: the real world inspiration of biological and other systems is leading to novel hardware. This must be abstracted into a computation model, and augmented with appropriate programming languages an tools. 70 years from now, the technology will be unrecognisable from today's ideas.

2 Classical history and unconventional futures

In a sense, classical computation got things backwards: theory before hardware and applications (figure 1). Unconventional computing seems to be taking a different route: the real world inspiration is leading to novel hardware (in some cases, wetware) devices, rather than directly to a model. Our job as computer scientists is to work out good underlying computational models and appropriate languages

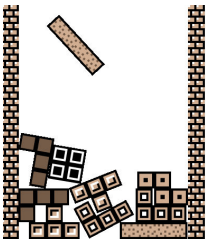


Figure 3. The wrong model (screenshot partway through a game of Not Tetris, http://stabyourself.net/nottetris2)

that naturally fit with the hardware, and also to engineer more efficient and flexible hardware (figure 2).

Getting a good abstract model is important. The wrong model (figure 3), an unnatural model, will mean that our ability to exploit the unconventional substrates will be compromised.

3 Computational models as abstractions of physics

We know that the classical model of computation captures too little of reality: its underlying workings formalise an essentially Newtonian view of physics. Quantum physics allows multiple symbols to be *superposed* in a single tape position [15], and *entangled* between positions. General relativity allows the machine's frame and the observer's frame to experience different proper times; in particular a Malament-Hogarth spacetime allows an observer to experience finite proper time whilst the machine that they are observing experiences infinite proper time [24]. And these two branches of physics are themselves a century old. What of quantum gravity computers, or string-theoretic computers? The Turing model is *unphysical*.

However, some unconventional computational models capture too much: like TMs they are unphysical, but in a different way. Analogue computers usually use *continuous* physical quantities as analogues of the value being computed. These continuous physical quantities are *modelled* as real numbers. A single real number can encode a countably infinite amount of information. But this does *not* mean that the physical quantity that it models can encode an infinite amount of information. This has nothing to do with quantum limits to continuity. Well before such limits, even the most accurately measured fundamental physical constants are not measured to more than 10 or 12 decimal places [35]. The most accurately measured physical quantity, the rubidium hyperfine frequency, is known to an accuracy of 2.5×10^{-15} [36]. The value of the mathematical constant π to 39 digits can give the volume of the observable universe to the nearest atom [4, p17]. To measure π to more precision than this, we would need a measuring device bigger than the size of the universe. Despite this, π has been calculated to 10 trillion decimal places [47]: an impressive computation, but a completely physically unmeasurable value. Computational models need to be based on real-world physics: not only the laws, but also the practical constraints.

What models of computation are suitable for natural physical computers? This includes not only exotic physics, but also biological systems. We need good abstractions, that not only do not impose unphysical requirements, but that naturally fit the implementations. So, for example, if a system is naturally noisy and non-deterministic, it is better to find a model that can exploit this, rather than engineer the substrate away from its natural state to one that better matches some unnatural deterministic model.

4 Inspired by biological modelling

Let's look at how biology is being conceptualised and modelled, in order to get some pointers to requirements for computational models of biological computers. We start with a pair of quotations, about organism-centric biology.

> Organic life exists only so far as it evolves in time. It is not a thing but a process—a never-resting continuous stream of events — Cassirer [10, p49, ch.IV]

> It must be a biology that asserts the primacy of processes over events, of relationships over entities, and of development over structure. — Ingold [25]

A process-centric description is arguably also needed in the context of emergence [43]. To summarise these ideas: "Life is a verb, not a noun." [19, p203, ch.X].

So, the emphasis from these authors is of process, dynamics, development (which, despite themselves being nouns, describe verb-like properties), rather than of entities, states, events. Let's look at these three features, how well they are captured by current formalisms, and what more is needed.

4.1 Process

"Process" might seem like an easy starting point, as we have process algebras and process calculi galore [5, 9, 22, 23, 30–33] in computer science. These describe the interactions between concurrently-executing processes, and (one of) the semantics of a process is its *trace*: a ("never-resting") stream of events.

Process algebras, with their non-terminating processes, can have their semantics modelled in non-well-founded set theory [2, 40]. NWF set theory replaces the usual axiom of foundation with the anti-foundation axiom (AFA); many of the well-known operations of set theory (such as union, intersection, membership) carry across. The crucial difference is that, unlike in the better known well-founded set theory, in NWF set theory we can have perfectly well-defined systems with infinite chains of membership that do not bottom-out, $\ldots \in X_3 \in X_2 \in X_1 \in X_0$, and cycles of membership, such as $X \in Y \in X$ and even $X \in X$.

Using NWF set theory gives a very different view of the world. With well-founded sets, we can start at the bottom (that is what well-foundedness guarantees exists), with the relevant "atoms", and construct sets from these atoms, then bigger sets from these sets, inductively. This seems like the natural, maybe the only, way to construct things. But non-well-foundedness is not like this. There are perfectly good non-well-founded sets that just cannot be built this way: there are sets with no "bottom" or "beginning": it can be "turtles all the way down" [21, p1]. NWF set theory allows sets that are intrinsically circular, or self-referential, too. It might be true that "the axiom of foundation has played almost no role in mathematics outside of set theory itself" [7, p58], but set theory has had an enormous impact on the way many scientists model the world. Might it be that the whole concept of reductionism relies on the (mathematically unnecessary) axiom of foundation? Process algebras, with their NWF basis, might well offer a new view on how things can be constructed.

But it is not all good news. Well-founded set theory is taught to school children; NWF set theory, coalgebra, and coinduction, are currently found only in quite densely mathematical books and papers. We need a *Coalgebra for Dummies*. One of the most accessible introductions currently available is Bart Jacobs' "two-thirds of a book in preparation" [26].

More importantly for programming unconventional computers, most process algebras cannot exploit endogenous novelty. Processes and communication channels are predefined; no new kinds of processes or channels can emerge and then be exploited by the formal system. This may require a reflective [27] process algebra. Reflection may be a pre-requisite for describing any system displaying open-ended novelty [42]. PiLar [12,13] is a reflective process-algebraic architecture description language, developed to define software architectures in terms of patterns of change; reflection allows it to change itself: to change the patterns of change. The mathematical underpinnings need to incorporate such features; NWF set theory, with its allowance of circular definitions, is suitable for modelling reflective systems that can model themselves.

4.2 Dynamics

For a formalism underpinning "dynamics", we could consider dynamical systems theory [3,8,44]. This is a very general formalism: a dynamical system is defined by its state space, and a rule determining its motion through that state space. In a continuous physical dynamical system, that rule is given by the relevant physical laws. Classical computation can be described in discrete dynamical systems terms [41], where the relevant rule is defined by the computer program. Hence it seems that dynamical systems approach can provide a route to an unconventional computational view of physical embodied systems exploiting the natural dynamics of their material substrates.

Dynamical systems can be understood at a generic level in terms of the structure of their state space: their attractors, trajectories, parameterised bifurcations, and the like [3,8,44]. Trajectories may correspond to computations and attractors may correspond to computational results [41]; new attractors arising from bifurcations may correspond to emergent properties [20,43].

A dynamical systems view allows us to unify the concepts of process and particle (of verb and noun). Everything is process (motion on a trajectory, from transient behaviour to motion on an attractor), but if viewed on a long enough timescale, its motion on an attractor blurs into a particle. "An attractor functions as a symbol when it is observed *... by a slow observer*" [1]. On this longer timescale the detailed motion is lost, and a stable pattern emerges as an entity in its own right. This entity can then have a dynamics in a state space of its own, and so on, allowing multiple levels of emergence.

However, the mathematical underpinnings support none of these exciting and intuitive descriptions. Classical dynamical systems theory deals with closed systems (no inputs or outputs, no coupling to the environment) in a static, pre-defined state space.

The closest the state space itself comes to being dynamic is by being parameterised, where a change in the parameter value can lead to a change in the attractor structure, including bifurcations. Here the parameter links a family of dynamical systems. If the parameter can be linked to a feature of the computational system, then it can be used to control the shape of the dynamics.

Ideally, the control of the parameter should be internal to the system, so that the computation can have some control its own dynamics. Current dynamical systems theory does not have this reflective component: the parameter is external to the system. A full computational dynamical systems theory would need to include metadynamics, the dynamics of the state space change. Currently metadynamics is handled in an *ad hoc* fashion, by separating it out into a slower timescale change [6,34].

4.3 Development

The requirement for "development", allowing (the state space of) systems to "grow", happens naturally in most classical programming languages: for example, statements such as malloc(n) or new Obj(p) allocate new memory for the computation to use, thereby increasing the dimensionality of the computational state space. However, this everyday usage is rarely cast in explicit developmental terms.

Explicit development is captured by generative grammars such as L-systems [38], and by rewriting systems such as P-systems [37] and other membrane computing systems. These discrete systems can be cast as special cases of "dynamical systems with dynamical structure" within the MGS language [17,18,29], based on local transformations of topological collections.

These formalisms capture mainly the growth of discrete spaces. There is still the interesting question of growing continuous spaces: how does a new continuous dimension appear in continuous time? How does a hybrid system containing both discrete and continuous dimensions grow?

If we are thinking of systems that can exhibit perpetual novelty and emergence, then we also need a system where the growth rules can grow. The growing space (new dimensions, new kinds of dimensions) should open up new possibilities of behaviour. One way to do this is to embed the rules in the space itself, so that as the space grows, the rules governing how the space grows themselves grow. This approach can be used to program self-replicating spaces [45]. Now the computation is not a trajectory though a static state space, it is the trajectory of the growing space itself.

4.4 Self-reference

Although "self-reference" is not one of the features identified from the biological modelling inspiration, it has come up in the discussions around each individual feature, and is a crucial feature of classical computation and biological self-reproduction.

The biologist Robert Rosen claims that there is a sense in which self-definition is an essential feature of life that cannot be replicated in a computer [39]. He defines organisms to be "closed to efficient causation": Aristotle's "efficient cause" is the cause that brings something about; life is self-causing, self-defining, autopoietic [28]. Rosen claims that "mechanisms", including computer programs (and hence simulations of life) cannot be so closed, because they require something outside the system to define them: they have an arbitrary non-grounded semantics. That is, there is an arbitrary separation of the semantics of the program (a virtual machine) and the implementation (the physical machine); life however has only the one, physical, semantics.

However, it is not as straightforward as that. Organic life also has an arbitrary semantics. As Danchin points out [14, p110], there is a level of indirection in the way organisms represent their functionality: the mapping from DNA codons to the amino acids they code for is essentially arbitrary. So life too may be embodied in a virtual machine with arbitrary semantics.

What is common to biological and computational self-reference is that the "data" and "program" are the "same kind of stuff", so that programs can modify data that can be interpreted as new programs. In biology this stuff comprises chemicals: a chemical may be passive data (uninterpreted DNA that codes for certain proteins); it may be an executing "program" (some active molecular machinery, possibly manipulating DNA).

So self-referential, self-modifying code is crucial in biology. It

is achievable in classical computation through reflective interpreted programs. It is plausible that this capability is also crucial for unconventional computation executing on the natural embodied dynamics of physical substrates.

5 Conclusions

Unconventional computers, particularly those embodied in biological-like substrates, may require novel programming paradigms. By looking to biology, we see that these paradigms should include as first class properties the concepts of: process, dynamics, development, and self-reference.

Some existing formalisms may suggest appropriate starting points, but much yet remains to be done. This should not be surprising: classical computation has matured tremendously over the last seventy years, while unconventional computing is still in its infancy. If over the next seventy years unconventional computing can make even a fraction of the advances that classical computing has made in that time, that new world of computation will be unrecognisably different from today.

REFERENCES

[1] Ralph H. Abraham, 'Dynamics and self-organization', in *Self-organizing Systems: The Emergence of Order*, ed., F. Eugene Yates, 599–613, Plenum, (1987).

[2] Peter Aczel, *Non-well-founded sets*, CSLI, 1988.

[3] Kathleen T. Alligood, Tim D. Sauer, and James A. Yorke, *Chaos : an introduction to dynamical systems*, Springer, 1996.

[4] Jörg Arndt and Christoph Haenel, π *Unleashed*, Springer, 2001.

[5] J. C. M. Baeten and W. P. Weijland, *Process Algebra*, Cambridge University Press, 1990.

[6] M. Baguelin, J. LeFevre, and J. P. Richard, 'A formalism for models with a metadynamically varying structure', in *Proc. European Control Conference, Cambridge, UK*, (2003).

[7] Jon Barwise and John Etchemendy, *The Liar: an essay on truth and circularity*, Oxford University Press, 1987.

[8] Randall D. Beer, 'A dynamical systems perspective on agent-environment interaction', *Artificial Intelligence*, **72**(1-2), 173–215, (1995).

[9] Luca Cardelli and Andrew D. Gordon, 'Mobile ambients', *Theor. Comput. Sci.*, **240**(1), 177–213, (2000).

[10] Ernst Cassirer, *An Essay on Man*, Yale University Press, 1944.

[11] B. Jack Copeland, 'The modern history of computing', in *The Stanford Encyclopedia of Philosophy*, ed., Edward N. Zalta, fall 2008 edn., (2008).

[12] Carlos Cuesta, Pablo de la Fuente, Manuel Barrio-Solórzano, and Encarnacin Beato, 'Coordination in a reflective architecture description language', in *Coordination Models and Languages*, eds., Farhad Arbab and Carolyn Talcott, volume 2315 of *LNCS*, 479–486, Springer, (2002).

[13] Carlos Cuesta, M. Romay, Pablo de la Fuente, and Manuel Barrio-Solórzano, 'Reflection-based, aspect-oriented software architecture', in *Software Architecture*, eds., Flavio Oquendo, Brian Warboys, and Ron Morrison, volume 3047 of *LNCS*, 43–56, Springer, (2004).

[14] Antoine Danchin, *The Delphic Boat: what genomes tell us*, Harvard University Press, 2002.

[15] David Deutsch, 'Quantum theory, the Church-Turing principle and the universal quantum computer', *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, **400**(1818), 97–117, (1985).

[16] C. C. Enz and E. A. Vittoz, 'CMOS low-power analog circuit design', in *Emerging Technologies, Tutorial for 1996 International Symposium on Circuits and Systems*, 79–133, IEEE Service Center, (1996).

[17] Jean-Louis Giavitto and Olivier Michel, 'Data structure as topological spaces', in *Proceedings of the 3nd International Conference on Unconventional Models of Computation UMC02*, volume 2509 of *LNCS*, pp. 137–150. Springer, (2002).

[18] Jean-Louis Giavitto and Antoine Spicher, 'Topological rewriting and the geometrization of programming', *Physica D*, **237**(9), 1302–1314, (2008).

[19] Charlotte Perkins Gilman, *Human Work*, McClure, Philips and Co, 1904.

[20] Jeffrey Goldstein, 'Emergence as a construct: History and issues', *Emergence*, **1**(1), 49–72, (1999).

[21] Stephen W. Hawking, *A Brief History of Time*, Bantam Dell, 1988.

[22] Jane Hillston, *A Compositional Approach to Performance Modelling*, Cambridge University Press, 1996.

[23] C. A. R. Hoare, *Communicating Sequential Processes*, Prentice Hall, 1985.

[24] Mark Hogarth, 'Does general relativity allow an observer to view an eternity in a finite time?', *Foundations of Physics Letters*, **5**(2), 173–181, (1992).

[25] Tim Ingold, 'An anthropologist looks at biology', *Man*, **25**, 208–229, (1990).

[26] Bart Jacobs, *Introduction to Coalgebra. Towards Mathematics of States and Observations*, 2005. draft available from http://www.cs.ru.nl/B.Jacobs/PAPERS/.

[27] Pattie Maes, 'Concepts and experiments in computational reflection', in *OOPSLA'87*, pp. 147–155. ACM Press, (1987).

[28] Humberto R. Maturana and Francisco J. Varela, *Autopoeisis and Cognition: the realization of the living*, D. Reidel, 1980.

[29] Olivier Michel, Jean-Pierre Banâtre, Pascal Fradet, and Jean-Louis Giavitto, 'Challenging questions for the rationale of non-classical programming languages', *International Journal of Unconventional Computing*, **2**(4), 337–347, (2006).

[30] Robin Milner, *A Calculus of Communicating Systems*, Springer, 1980.

[31] Robin Milner, *Communication and Concurrency*, Prentice Hall, 1989.

[32] Robin Milner, *Communicating and Mobile Systems: the* π*-Calculus*, Cambridge University Press, 1999.

[33] Robin Milner, *The Space and Motion of Communicating Agents*, Cambridge University Press, 2009.

[34] E. Moulay and M. Baguelin, 'Meta-dynamical adaptive systems and their application to a fractal algorithm and a biological model', *Physica D*, **207**, 7990, (2005).

[35] NIST. The NIST reference on constants, units, and uncertainty, 2011. http://physics.nist.gov/cuu/Constants/.

[36] NPL. What is the most accurate measurement known?, 2010. http://www.npl.co.uk/reference/faqs/what-is-the-most-accurate-measurement-known-(faq-quantum).

[37] Gheorghe Păun, 'Computing with membranes', *Journal of Computer and System Sciences*, **61**(1), 108–143, (2000).

[38] Przemyslaw Prusinkiewicz and Aristid Lindenmayer, *The Algorithmic Beauty of Plants*, Springer, 1990.

[39] Robert Rosen, *Life Itself: a comprehensive enquiry into the nature, origin, and fabrication of life*, Columbia University Press, 1991.

[40] Davide Sangiorgi, 'On the origins of bisimulation and coinduction', *ACM Transactions on Programming Languages and Systems*, **31**(4), 15:1–15:41, (2009).

[41] Susan Stepney, 'Nonclassical computation: a dynamical systems perspective', in *Handbook of Natural Computing, volume II*, eds., Grzegorz Rozenberg, Thomas Bäck, and Joost N. Kok, chapter 52, Springer, (2011).

[42] Susan Stepney and Tim Hoverd, 'Reflecting on open-ended evolution', in *ECAL 2011, Paris, France, August 2011*, pp. 781–788. MIT Press, (2011).

[43] Susan Stepney, Fiona Polack, and Heather Turner, 'Engineering emergence', in *ICECCS 2006: 11th IEEE International Conference on Engineering of Complex Computer Systems, Stanford, CA, USA, August 2006*, pp. 89–97. IEEE, (2006).

[44] Steven H. Strogatz, *Nonlinear Dynamics and Chaos*, Westview Press, 1994.

[45] Kohji Tomita, Satoshi Murata, and Haruhisa Kurokawa, 'Self-description for construction and computation on graph-rewriting automata', *Artificial Life*, **13**(4), 383–396, (2007).

[46] Alan M. Turing, 'On computable numbers, with an application to the entscheidungsproblem', *Proceedings of the London Mathematical Society*, **s2-42**(1), 230–265, (1937).

[47] Alexander J. Yee and Shigeru Kondo. Round 2 . . . 10 trillion digits of pi, 2011. http://www.numberworld.org/misc_runs/pi-10t/details.html.

The coordination of probabilistic inference in neural systems

William A Phillips[1]

[1]**Abstract.** Life, thought of as adaptively organised complexity, depends upon information and inference, which is nearly always inductive, because the world, though lawful, is far from being wholly predictable. There are several influential theories of probabilistic inference in neural systems, but here I focus on the theory of Coherent Infomax, and its relation to the theory of free energy reduction. Coherent Infomax shows, in principle, how life can be preserved and improved by coordinating many concurrent inferences. It argues that neural systems combine local reliability with flexible, holistic, context-sensitivity. What this perspective contributes to our understanding of neuronal inference is briefly outlined by relating it to cognitive and neurophysiological studies of context-sensitivity and gain-control, psychotic disorganization, theories of the Bayesian brain, and predictive coding. Limitations of the theory and unresolved issues are noted, emphasizing those that may be of interest to philosophers, and including the possibility of major transitions in the evolution of inferential capabilities.

1 INTRODUCTION

Many forms of organised complexity have arisen during nature's long journey from uniformity to maximal disorder, despite the ever present forces of noise and disorder. Biological systems are able to create and preserve organised complexity, by, in effect, making good predictions about the likely consequences of the choices available to them. This adaptively organised complexity occurs in open, holistic, far-from-equilibrium, non-linear systems with feedback. Though usually implicit, probabilistic inference is crucial, and useful inference is only possible because the laws of physics are sufficiently reliable. The endless variety of individual circumstances and the prevalence of deterministic chaos and quantal indeterminacy make many things uncertain, however; so, to thrive, biological systems must combine reliability with flexibility.

It is in neural systems that the crucial role of probabilistic inference is most obvious. Helmholtz correctly emphasized the centrality of unconscious inference to perception, and many examples of its use for contextual disambiguation can be given [1]. Furthermore, it has also now been explicitly shown how such unconscious inference may also be central to reinforcement learning, motor control, and many other biological processes [2].

Better formalisation of these issues is clearly needed, so Section 3 outlines an elementary neurocomputational perspective that uses information theory measures to shed light on them, i.e. the theory of Coherent Infomax [3, 4, 5]. A major advantage of that theory is that, in addition to being formally specified and simulated in large artificial neural networks, it has wide-ranging empirical roots, being related, often in detail, to much empirical data from neuroanatomy, cellular and synaptic physiology, cognitive psychology, and psychopathology. Section 4 briefly discusses relations between this theory and that of free energy reduction [2], to which it has deep connections, and which has been applied to an even wider range of phenomena than has Coherent Infomax. Finally, in Section 5, difficulties of the theory and unresolved issues of possible philosophical interest are briefly discussed. First, the following section outlines some of the difficult conceptual problems to be solved by theories of neuronal inference.

2 THEORIES OF NEURONAL INFERENCE AND DIFFICULT PROBLEMS THAT THEY MUST SOLVE

The preceding arguments suggest several issues on which we need to make progress. What is organised complexity? What are the capabilities and constraints of various forms of inductive inference, e.g. classical versus Bayesian [6], conscious versus unconscious [7]? How is reliability combined with flexibility, i.e. how is information about reliable generalities combined with information about individual particularities? How is localism combined with holism? What forms of learning and processing does neural inference require, and how are they implemented at the synaptic, local circuit, and systems levels? Do biological capabilities for probabilistic inference evolve towards forms of inference with greater accuracy, generality, or abstraction? Information theory measures such as Shannon entropy and free-energy have been applied to these issues, but how can they be tested and what do they contribute to our understanding?

Several theories of probabilistic inference in neural systems have been proposed, including the Bayesian brain [8, 9], predictive coding [10], efficient coding and Infomax [11, 12, 13], and sensorimotor integration [14]. It has been argued that all can be unified via the principle of least variational free energy [2, 15, 16]. The free energy principle is formulated at the level of the interaction of the system with its environment – and emphasizes Bayes optimal inference using hierarchical architectures with backward as well as forward connections. As free energy theory offers a broad synoptic view of neuronal inference the Coherent Infomax theory will be compared to that.

The theory of Coherent Infomax stresses the necessity of avoiding information overload by selecting only the information that is needed. This necessity arises not only from requirements of computational tractability, but also from an unavoidable property of noisy high-dimensional spaces. As dimensionality increases the number of possible locations in that space increases

[1] Dept. of Psychology, Univ. of Stirling, FK9 4LA, UK.
And Frankfurt Institute of Advanced Studies
Email: wap1@stir.ac.uk.

exponentially, with the consequence that nearly all events occur at novel locations. Probabilistic inference based on prior events then becomes impossible. This problem is well-known within the machine learning community, where it is referred to as the 'curse-of-dimensionality'. It may be avoided by selecting only the information that is 'relevant'; but how? Coherent Infomax suggests a solution: select information that reveals latent statistical structure in the available data. Useful combinations of datasets between which to seek predictive relations may be found by genetic search prescribing gross system architectures combined with the learning algorithms of Coherent Infomax, as outlined in the following section.

3 THE THEORY OF COHERENT INFOMAX: A BRIEF OUTLINE

An unavoidable consequence of the curse-of-dimensionality is that large amounts of data must be divided into subsets that are small enough to make learning feasible. If they were processed independently, however, then relations between the subsets would be unobservable. Success in finding useful relations would then be completely dependent upon the original division into subsets, but that is unlikely to be adequate unless the crucial relations were already known. Coherent Infomax responds to this dilemma by dividing data at each level of an interpretive hierarchy into many small subsets, and searching for variables defined on them that are predictably related across subsets. This strategy allows for endlessly many ways in which the data can be divided into subsets and linked by modulatory coordinating interactions between them.

These considerations suggest minimal requirements for local neural processors performing such inference. They must have a subset of inputs within which latent variables may be discovered and compressed into fewer dimensions. These are referred to as driving, or receptive field (RF), inputs. They must also receive inputs conveying information about the activity of other processors with which they are to seek predictive relations. These are referred to as contextual field (CF) inputs. They control the gain of response to the driving RF inputs but cannot by themselves drive processor activity, because, if they did, that would contradict the strategy for avoiding the curse-of-dimensionality. Given this constraint, each local processor can have a rich array of contextual inputs, far richer than its array of driving inputs. It is the contextual input that enables the local processor to discover relevant variables in its driving input. For example, the inclusion of reward signals in the context will enable it discover those driving variables that are predictably related to that kind of reward. Reward signals are not necessary for this kind of learning, however, as it discovers latent statistical structure between any inputs to the local processors.

The theory of Coherent Infomax has grown from combining such considerations with much empirical data from several relevant disciplines [3, 4, 5]. Only a brief outline is given here. For full formal presentations see the original publications. The theory uses three-way mutual information and conditional mutual information to show how it is possible, in principle, for contextual inputs to have large effects on the transmission of information about the primary driving inputs, while transmitting little or no information about themselves, thus influencing the transmission of cognitive content, but without becoming confounded with it. Guided by neuroanatomy, the gross system architecture assumed is that of at most a few tens of hierarchical layers of processing, with very many specialized but interactive local processors at each stage. Feed forward connections between layers are driving, whereas a larger number of lateral and feedback connections provide coordinating gain-control as shown in Figure 1. Minimally, the function of local processors is

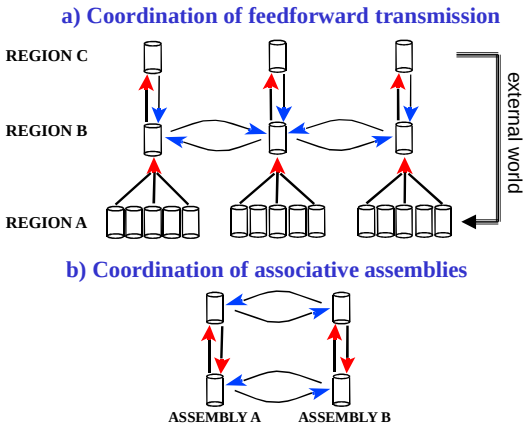


Figure 1. Examples of system architectures that could be built from the local neural processors of Coherent Infomax, shown here as small cylindrical columns. Though only a few are shown in each region, in useful applications, as in mammalian cerebral cortex, there would be very many in each region. Receptive field connections, shown by thick lines, provide the input from which information is to be selected and compressed. Coordinating contextual field connections, shown by thin lines, control the gain of response, and provide the inputs with which predictive relations are to be sought.

The Objective of Coherent Infomax is:
Max I(X;R) so that I(X;R;C) > I(X; R|C) & Min I(X; C|R)

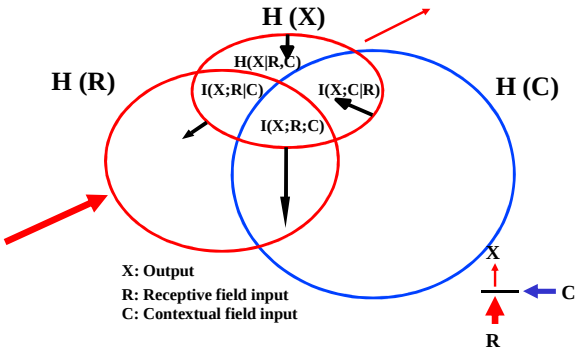


Figure 2. The objective of local processors in Coherent Infomax. The ovals show the Shannon entropy in each of three probability distributions. Information flow through the local processor is shown in the small icon, bottom right. Contextual entropy can be greater than the other two because it is not to be transmitted in the output. Thus, it enables narrowly focussed receptive field processing to operate within a broad context.

to select and compress that information in their driving receptive field (RF) input that is relevant to the current task and situation, as indicated by the contextual field (CF) input that modulates transmission of RF information. This is formalized as an objective function describing the signal processing work to be done, as shown in Figure 2 by arrows associated with each of the four components of the output H(X). Outward pointing arrows show components that should be increased, with priority being shown by arrow length. Inward pointing arrows show components that should be decreased. In short, the objective is to maximise the information transmitted about the receptive field input subject to the constraints of substantial data reduction while emphasizing the mutual information between receptive field input and contextual field input and minimizing any information transmitted specifically about the context.

To show how that objective could be met in neural systems, a biologically plausible activation function for idealized local neural processors was formulated to include the required gain-control, and a learning rule for modifying the synaptic strengths of the connections between these local processors was derived analytically from the objective function. What most impressed us about the consequent learning rule is that, although it was deduced formally from the objective function, assuming none of the physiological evidence concerning the dependence of synaptic plasticity on current and prior activity, it is broadly in agreement with that evidence. The theory of Coherent Infomax thus shows how it is possible for neural systems to perform probabilistic inference in a way that combines reliability with flexibility, and localism with holism, while making useful inference feasible by selecting only information that is relevant, and thus avoiding the curse-of-dimensionality. It has guided studies of common neurobiological foundations for cortical computation [17], dynamic coordination in the brain [18], cognitive impairments in schizophrenia [19], and of relations between probability theory, organised complexity and brain function [20].

4 RELATIONS TO THE THEORY OF FREE ENERGY REDUCTION

The current growth of interest in inference and prediction as possible keys to a fundamental understanding of neuronal systems is seen in the many groups working on 'predictive coding' and the 'Bayesian brain' as cited in Section 2. Those theories do not usually make use of gain-control or context to select the relevant information to be coded and used, however, and rarely show explicitly how the curse-of-dimensionality can be avoided. One theory that may be able to do so, however, is that proposing a unifying brain theory based on ideas from statistical physics and machine learning [2]. This has already received deep philosophical examination, and been found to have considerable interest from that perspective [21], even though it still needs further development. It interprets many aspects of neural structure and function as having evolved to reduce Helmholtz free-energy using a form of predictive coding in which ascending activities predicted by feedback descending from higher levels in the hierarchy are suppressed. In contrast to this, Coherent Infomax proposes that activities predicted by contextual input can be amplified. Thus, the form of predictive coding used in free energy theory seems to imply effects of context that are in opposition to those of Coherent Infomax. Furthermore, the theory of free energy reduction is formulated at the level of an agent in an environment with distal causes and parameters that are hidden from the agent; Coherent Infomax is formulated at the level of local neural processors operating within a large population of other such processors, with which they can communicate either directly or indirectly.

There are at least three grounds for thinking that these two theories are not in essence opposed, however. First, both theories imply that the fundamental objective of neuronal dynamics is to reduce any differences between predicted and observed probability distributions. Indeed, it may even be possible to unify the two perspectives by formulating the objective of Coherent Infomax as the maximisation of predictive success and of free energy reduction as the minimisation of prediction failure (Phillips and Friston, in preparation). Such a common goal could be described as maximising the transmission of information that is relevant to the context, or alternatively as reducing uncertainty about sensory inputs given the contextual constraints. Second, the two theories may be complementary, rather than opposed, because Coherent Infomax emphasizes lateral connections between streams of processing dealing with distinct datasets, while also including some downward connectivity, whereas the theory of free energy reduction emphasizes downward connections, while also including some lateral connectivity. Third, it has been argued that predictive coding theories can be made formally equivalent to theories based on evidence for amplifying effects of top-down attentional inputs [22]. This was done by reorganising the computations required for predictive coding, and assuming that suppressive effects of prediction operate on intra-regional signals, rather than on inter-regional signals. Furthermore, a detailed model of that form of predictive coding argues that it is compatible with much of the neurobiological evidence [23]. These studies therefore suggest that some form of predictive coding may be compatible with both Coherent Infomax and the theory of free energy reduction. Deeper examination of relations between those two theories is therefore a major task for the future.

5. UNRESOLVED ISSUES AND DIFFICULTIES OF THE THEORY

The conceptual depth and empirical scope of the free energy and Coherent Infomax theories raises many unresolved and controversial issues, some of which may have philosophical significance. There is time here to mention only a few, and each in no more than speculative and flimsy outline.

First, is any unified theory of brain function possible? As a recent philosophical examination of the free energy theory shows this is an issue of lasting debate, with the 'neats' saying 'Yes', and the 'scruffies' saying 'No' [21]. As the issue cannot be resolved by failing to find any unifying theory, it can only be resolved by finding one. Some are happy to leave that search to others, on the assumption that Darwinian evolution is the only unifying idea in biology. Even if true that need not deter the search for unifying principles of brain function, however, because it can be argued that free energy theory both formally specifies what adaptive fitness requires and shows how neural systems can meet those requirements (Friston, personal communication).

Second, another crucial issue concerns the possibility of major transitions in the evolution of inferential capabilities.

Seven major transitions in the evolution of life have been identified [24], such as the transition from asexual to sexual reproduction. Only one of those concerned cognition, i.e. the transition to language. Major transitions in the evolution of inferential capabilities prior to language are also possible, however, and it is crucial to determine whether this is so because empirical studies of inferential capabilities will be misinterpreted if they are assumed to reflect a single strategy, when instead they reflect a mixture of strategies, either across or within species. One way in which work of the sort discussed here could contribute to this issue is by proposing various possible inferential strategies. They could range from those with requirements that are easier to meet but with severely limited capacities, through intermediate stages of development, to those having more demanding requirements but with enhanced capabilities. Some possible transitions are as follows: from predictions only of things that are directly observable to estimates of things not directly observable; from generative models averaged over various contexts to those that are context specific; from hypotheses determined by input data to those that are somehow more internally generated; from probabilistic inference to syntactic structure, and, finally, from hypothesis testing to pure hypothesizing freed from testing. Within stages marked by such transitions there would still be much to be done by gradual evolutionary processes. For example, context-sensitive computations can make astronomical demands on computational resources, so they will be useful only if appropriate constraints are placed on the sources and size of contextual input, as already shown for its use in natural language processing [25]. Thus, even given the ability to use contextual information, the search for useful sources of contextual input could still be a lengthy process, even on an evolutionary timescale, and produce much diversity.

Third, how can apparently simple objectives, such as specified by Coherent Infomax and free energy theory, help us understand the overwhelming evidence for wide individual differences in cognitive style and capabilities? To some extent answers to this question are already available as it has been shown that within human cognition there are wide variations in context-sensitivity across sex and occupation [26], culture [27], schizotypy [28], and developmental stage [29]. The use of these theories to help us understand the diversity of cognitive capacities both within and between species is as yet only in its infancy, however.

Fourth, why are there several different neurobiological mechanisms for gain-control? Earlier work done from the Coherent Infomax perspective, both in relation to normal and psychotic cognition [19], emphasized only NMDA synaptic receptors for the predominant excitatory neurotransmitter glutamate, but we now realize that several other gain-control mechanisms are also important, particularly at the level of micro-circuitry involving inhibitory inter-neurons. The various uses, capabilities and limitations of these different mechanisms for gain-control remain to be determined.

Fifth, as Coherent Infomax is formulated at the level of local neural processors that operate only within a population of other such processors, are they not doomed to imprisonment in such a 'Chinese room', with no hint of a world beyond? As Fiorillo argues, neuroscience must be able to 'take the neuron's perspective' [30], but how can that be done without thereby losing contact with the distal world beyond? Coherent Infomax suggests an answer to this dilemma, first, by being formulated explicitly at the level of the local neuronal processor, and, second, by searching for predictable relations between diverse datasets. Discovery of such interdependencies implies the existence of distal causes that produce them. The more diverse the datasets the more distal their common origins are likely to be. This can be seen as a neurocomputational version of Dr Johnson's refutation of idealism when he kicked a stone and said "I refute it thus". A distal reality is implied both by the agreement between what is seen and what is felt, and by the successful prediction of the outcomes of action. Though this argument seems plausible to me, I am not a philosopher, so it may be in need of closer philosophical examination.

Sixth, coherence, as conceived within this theory, depends upon the long-term statistics of the co-occurrence of events defined at the highly specialized level of receptive fields, which convey information only about fragments of the current state as a whole, so how can episodic capabilities that deal with unique events, such as working memory and episodic memory, be included within such a conception? My working assumption is that these capabilities are closely related to the syntactic grammars of language and schematic structures. Though syntactic and statistical conceptions of cognition have long been contrasted, there is no fundamental conflict between them because, as many studies have shown, grammars can be acquired by statistical inference. The use of such grammars to create novel but relevant patterns of activity seems to be, in essence, close to what the theory of Coherent Infomax has to offer, but I know of no attempt to explore that possibility.

Seventh, how can attention and consciousness be included within these theories? Within Coherent Infomax, attention is assumed to operate via the contextual inputs, which are purely modulatory as required. One psychophysical study of texture perception by humans used the formal information theoretic measures of the theory, and, indeed, in that case attention had the complex set of properties predicted [30]. That one study has not been followed-up, however, and though it has promise, far more needs to be done. The theory of free energy reduction has also been related in detail to attention [31], but in that case also far more is needed.

Eighth, can the dynamics of biological systems be described as maximising a formally specified objective without implying that they have a long-term objective? This question is distinct from the much debated issue contrasting descriptive and prescriptive formulations. Instead, it concerns the temporal course of evolution. Is it progressive or not? Evolutionary biologists are divided on this issue, but Coherent Infomax implies that it can be progressive, provides a conceptual measure of the progress, i.e. as increasing organised complexity, and suggests ways in which neuronal systems contribute to that progress [20]. We can then think of life at the ecological and species levels, not as 'evolved to reproduce', but as 'reproducing to evolve'; i.e. in the direction of the formally specified objective. From that perspective we can think of our own individual efforts as directed, not merely towards survival, but as directed towards whatever organised complexities we choose to create.

Acknowledgements: I thank Andy Clark, Karl Friston, and Jim Kay for insightful discussions of issues addressed in this paper.

REFERENCES

[1] Phillips, W. A., von der Malsburg, C. & Singer, W. Dynamic coordination in brain and mind. In: *Dynamic coordination in the brain: from neurons to mind*. Strüngmann forum report: Vol. 5. Von der Malsburg, C.; Phillips, W. A.; Singer, W., Eds., MIT Press, Cambridge, MA, USA, Chapter 1, pp 1-24, (2010)

[2] Friston, K. J. The free-energy principle: a unified brain theory? *Nat. Rev. Neurosci. 11*, 127–138, (2010).

[3] Phillips, W. A.; Kay, J.; Smyth, D. The discovery of structure by multi-stream networks of local processors with contextual guidance. *Network-Comp. Neural Systems*, *6*, 225–246, (1995).

[4] Kay, J.; Floreano, D.; Phillips, W. A. Contextually guided unsupervised learning using local multivariate binary processors. *Neural Networks 11*, 117–140, (1998).

[5] Kay, J.; Phillips, W. A. Coherent infomax as a computational goal for neural systems. *B. Math. Biol.* 73, 344-372. DOI 10.1007/s11583-010-9564-x, (2011).

[6] Jaynes, E. T. *Probability Theory: The Logic of Science*, Edited by G. Larry Bretthorst Cambridge University Press. Cambridge, UK, (2003).

[7] Engel, C.; Singer, W. *Better than Conscious?* Strüngmann forum report: Vol. 1. MIT Press, Cambridge, MA, USA, (2008).

[8] Yuille, A., & Kersten, D. Vision as Bayesian inference: analysis by synthesis? *Trends Cogn Sci. , 10* (7), 301-8,(2006).

[9] Knill, D. C.; Pouget, A. The Bayesian Brain: the role of uncertainty in neural coding and computation. *Trends Neurosci*., *27*, 712-719.

[10] Rao, R. P., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nat Neurosci. , 2* (1), 79-87, (2004).

[11] Barlow, H. B. Inductive inference, coding, perception, and language. *Perception , 3*, 123-34, (1974).

[12] Barlow, H. B. Possible principles underlying the transformations of sensory messages. In W. Rosenblith (Ed.), *Sensory Communication* (pp. 217-34). Cambridge, MA: MIT Press, (1961).

[13] Linsker, R. Perceptual neural organization: some approaches based on network models and information theory. *Annu Rev Neurosci. , 13*, 257-81, (1990).

[14] Wolpert, D. M., Diedrichsen, J., & Flanagan, J. R. Principles of sensorimotor learning. *Nat Rev Neurosci.* , in press.

[15] Friston, K., Kilner, J., & Harrison, L. A free energy principle for the brain. *J Physiol Paris. , 100* (1-3), 70-87, (2006).

[16] Friston, K. J.; Stephan, K.E., Free-energy and the brain. *Synthese*, 159, 417-458, (2007).

[17] Phillips, W. A.; Singer, W. In search of common foundations for cortical computation. *Behav. Brain Sci.*, *20*, 657–722, (1997).

[18] von der Malsburg, C., Phillips, W. A., & Singer, W. Eds. *Dynamic coordination in the brain: from neurons to mind.* Strüngmann forum report: Vol. 5. MIT Press, Cambridge, MA, USA, (2010).

[19] Phillips, W. A.; Silverstein, S. M. Convergence of biological and psychological perspectives on cognitive coordination in schizophrenia. *Behav. Brain Sci.*, *26*, 65–138, (2003).

[20] Phillips, W. A. Self-organized complexity and Coherent Infomax from the viewpoint of Jaynes's probability theory. *Information*, 3(1), 1-15, DOI 10.3390/info3010001, (2012).

[21] A. Clark. Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behav. Brain Sci*. (in press).

[22] Spratling, M. W. Predictive-coding as a model of biased competition in visual attention. *Vis. Res.* 2008, *48*, 1391-1408, (2008).

[23] C. Wacongne, J-P Changeaux and S. Deheane. A neuronal model of predictive coding accounting for the mismatch negativity. *J. Neurosci*. 32: 3665-3678 (2012).

[24] Szmatháry, E.; Maynard Smith, J. The major evolutionary transitions. *Nature*, *374*, 227-232, (1995).

[25] Ginter, F.; Boberg, J.; Jarvinen, J.; Salakoski, T. New techniques for disambiguation in natural language and their application to biological text. *J. Mach. Learn. Res., 5*, 605-621, (2004).

[26] Phillips, W.A., Chapman, K.L.S., & Berry, P.D. Size perception is less context-sensitive in males. *Perception*, 33, 79–86, (2004).

[27] Doherty, M.J., Tsuji, H., & Phillips, W.A. The context-sensitivity of visual size perception varies across cultures. *Perception*, 37, 1426–1433, (2008).

[28] Uhlhaas, P. J., Phillips, W. A., Mitchell, G., & Silverstein, S. M. Perceptual grouping in disorganized schizophrenia. *Psychiatry Research*, 145, 105-117, (2006).

[29] Doherty, M. J., Campbell, N. M., Tsuji, H., & Phillips, W. A. (2009) The Ebbinghaus illusion deceives adults but not young children, *Developmental Science*, 1-8; DOI 10.1111/j.1467-7687.2009.00931.x

[30] C. D. Fiorillo. On the need for a unified and Jaynesian definition of probability and information within neuroscience. *Information*, 3:175-203 (2012).

[31] W. A. Phillips and B. J. Craven. Interactions between coincident and orthogonal cues to texture boundaries. *Percep and Psychophys* 62:.1019-1038.

[32] H. Feldman, H., and K. J. Friston. Attention, uncertainty, and free-energy. *Frontiers in Human Neuroscience , 4*, 215 (2010).

Neurobiological Computation and Synthetic Intelligence

Craig A. Lindley[1]

Abstract. Cognitivist approaches to the development of engineered systems having properties of autonomy and intelligence are limited in their lack of grounding and emphasis upon linguistically derived models of the nature of intelligence. An alternative approach to the synthesis of intelligence is to take inspiration more directly from biological nervous systems. Such an approach, however, must go far beyond twentieth century models of artificial neural networks (ANNs), which greatly oversimplify brain and neural functions. The synthesis of intelligence based upon biological foundations must draw upon and become part of the ongoing rapid expansion of the science of biological intelligence. This includes an exploration of broader conceptions of information processing, including different modalities of information processing in neural and glial substrates. The medium of designed intelligence must also expand to include biological, organic and inorganic molecular systems capable of realising asynchronous, analog and self-* architectures that digital computers can only ever simulate.

1 INTRODUCTION

Alan Turing [23] provided the definitive challenge for research in artificial intelligence (AI) of creating a computer program that could not be distinguished in communication via a remote interface from a human operator. This challenge has had the great advantage of providing a constrained and measurable problem for artificial intelligence, which more generally suffers from being highly unconstrained [9]. That is, AI seeks to make machines more intelligent, which immediately raises questions of what intelligence is and how it might be detected or measured. The focus of the Turing test on textual discursive capability has resulted in a symbolic AI paradigm that emphasizes the definition of formalised linguistic representations and the logic of high level cognitive operations that are involved in verbal and textual discourse. The Turing test meshes well with Newell and Simon's *physical symbol system hypothesis* [14] that: "A physical symbol system has the necessary and sufficient means for general intelligent action." In this conception, the foundations of discursive intelligence become the foundations of general intelligence.

While this approach may lead to systems that can pass the Turing test within limited contexts, as a general paradigm of intelligence it are severe limitations, as summarised by Lindley [12]. Indeed, following the argument of [12], not only is the Turing test limited to symbolic discourse, the foundation of Turing's challenge, *computing machinery*, is a narrow and historically situated understanding of machines. In the age of nanotechnology and biotechnology, the distinction between machines and biological organisms breaks down. This suggests that the realisation of intelligence by design must shift foundations towards the design of self-replicating, self-assembling and self-organizing biomolecular elements or analogs capable of generating cognizing systems as larger scale assemblies, analogous to the neurobiological substrate of human cognition. That is, the paradigm of biomolecular engineering implies the construction of human level intelligence (HLI), not from the top-down by the manipulation of symbols, but from the bottom-up by the synthesis of neural architectures starting at the level of molecular engineering.

Here this bottom-up approach will be referred to as *synthetic intelligence* (SI). SI is in contrast to symbolic AI, the latter being largely based upon top-down analysis of higher level cognitive functions with the aim of deriving an abstract symbolic machinery that can be implemented at a logical level independently of mechanisms by which representation and logical inference may be automated.

While SI can be motivated by reflection upon the limited success of symbolic AI, it finds a strong and obvious demonstration of its principles in human neural systems, of which only limited abstract functions are captured by symbolic AI. However, the *achievement* of SI as a product of engineering is, of course, yet to be demonstrated. Moreover, the pursuit of SI immediately raises the question of how well we understand neurobiology. One of the issues to consider in this is the degree to which our understanding of neurobiology is conditioned by historically situated metaphors and current technologies, just as [12] describes the metaphorical construction of AI in terms of intelligence as computation, and of robotics as the reinvention of the (typically) human form in the media of twentieth century electromechanical engineering technologies.

This paper considers the more specific question of the nature and role of information processing in understanding the functions of neurobiology, especially those functions that may appear most relevant to the realisation of HLI, and the implications of this for how SI may be achieved. A *metaphor* can be defined as "a figure of speech in which a word or phrase is applied to an object or action that it does not literally denote in order to imply a resemblance" (http://www.collinsdictionary.com/dictionary/english/metaphor, accessed 11 January 2012). In our discourse on neurobiology it is often very unclear when terms of description are metaphorical and when they are literal. For example, if it is claimed that the brain is a computer, it is clearly not being suggested that we have a hard disk and a motherboard in our heads. Rather, we use the metaphor of computation to highlight an aspect of how the brain might function. However, the degree to which this is a literal or a metaphorical description depends upon the degree to which specific models of computation capture more or less

[1] School of Computing Science, Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden. craig.lindley@bth.se.

fundamental aspects of the operation of the brain associated with the realisation of HLI. For SI, which seeks to *realise* designed intelligence, the distinction between the literal and the metaphorical, and what falls within the gap, can be critical to success or failure.

2 NEUROBIOLOGY: SYSTEMS, SIGNALS AND PROCESSES

Understanding the brain requires understanding at many different spatial scales, including those of ion channels (at a scale around 1 pm), signalling pathways (1 nm), synapses (1 μm), dendritic subunits (10 μm), neurons (100 μm), microcircuits (1 mm), neural networks (1 cm), subsystems (10 cm) and the whole nervous system (1 m) [22]. For understanding intelligence, a key question is: at what levels of the hierarchy are information processes critical to intelligence carried out? Behind this question is that of what aspects of the physiology, structure and operation of the brain that are *not* captured by the concept of information processing may nevertheless be critical to achieving HLI? Or turning the last question around, *which* concept(s) of information processing are critical to achieving HLI?

Symbolic AI has focussed upon behaviour, what may be inferred from behaviour regarding functional capacities, and the derivation from language constructs of more formalised models (e.g. taxonomies, propositions, rules, etc) of linguistic forms. *Subsymbolic* AI has focussed upon simplified models of neurons and neural networks characterised by different learning rules and topologies. Since HLI has yet been approached by any kind of AI system, it must be asked if these approaches are adequate, or whether comparison with biological brains and nervous systems can reveal structures, functions, processes or principles that have not yet been used in AI that may nevertheless be critical for the achievement of artificial or synthetic HLI. Since SI is concerned with the bottom-up creation of intelligence, the discussion here will focus upon the lower layers of the hierarchy, i.e. the levels of simple neural interconnections and below (ignoring larger scale circuits, network topologies and subsystems).

The most ubiquitous image of the brain is that of a kind of wet computer with circuitry consisting of a vast network of interconnected neurons transmitting electrical signals among themselves, with each neuron summing weighted inputs and issuing an output signal if the sum of inputs exceeds a threshold (the *integrate-and-fire* model of neurons). This view is the *neuronal doctrine* [24], which places neurons and their synaptic interconnections at the centre of brain functionality.

A single bipolar neuron cell (Figure 1, top) consists of a cell body from which there extend dendritic trees surrounding the cell body, and an elongated axon that also leads to a branching end structure. Dendrites accept inputs from other neurons in the form of neurotransmitters, via synapses that often occur on small projections referred to as *dendritic spines*. Any given input can have an additive or subtractive effect upon the summation of inputs at the neuron body, with different inputs having different strengths, as modelled at the bottom of Figure 1. When a sufficient balance of additive over subtractive inputs is received, the cell body accumulates enough of a potential difference between the inner and outer surfaces of its surrounding membrane for ion channels embedded in the membrane to open, leading to the movement of ions between the inside and the outside of the cell. This movement of ions cascades along the cell axon as an action potential, a voltage spike providing a signal that is transmitted via the terminal axonal branches to the dendrites of other neurons. After the passage of such an electrochemical pulse, the ionic balance across the neuron cell membrane returns to the rest potential. The details of this process are covered in many neuroscience texts (e.g. [2], [21], [11]). However, it is relevant to note that the speed of transmission of an action potential in the fastest, myelinated, cells of the peripheral nervous system is about 150 ms^{-1}, a speed that is two million times slower than the transmission of an electric signal along a wire or a pulse of light along an optical fibre. The metaphor of neural transmission as an electrical signal is highly misleading in this respect; an action potential is a measure and propagator of a cascading flow of charged particles, a process of electrochemical diffusion.

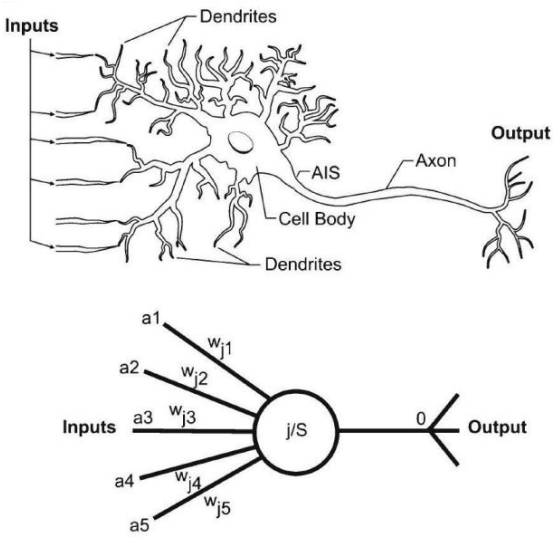


Figure 1. A simplified representation of an "integrate-and-fire" neuron cell and its mathematical structure, from Hameroff (2009).

The primary signal transmission connections between neurons occur at *synapses*. A synapse consists of a synaptic terminal on the presynaptic side, a synaptic cleft, which is a gap of ~20 nm width, and a postsynaptic membrane on the receiving side. A selection from a wide range of possible neurotransmitters are issued from synaptic terminals of the activated neuron. The neurotransmitters move across the synaptic cleft to receptors on dendrites on the post-synaptic side. Hence action potentials as such are not directly transmitted from one neuron to another (in most cases), the inter-neuron connection being mediated by neurotransmitters passing across the synaptic cleft and into receptor proteins in the post-synaptic membrane. Hence chemical synapses actually provide electrical and physical isolation between interconnected neurons, another departure from the metaphor of an electrical signal processing network. However, some neurons *do* have electrical synaptic interconnections, *gap junctions* that are channels that allow the

passage of ions for direct propagation of electrical signals, but also allowing the passage of larger molecules, thereby creating metabolic coupling in addition to electrochemical coupling between neurons [24]. There are also *anterograde* connections from post-synaptic neurons to presynaptic neurons, typically realised by gaseous neurotransmitters such as nitric oxide.

Neurotransmitters, as well as hormones secreted by neurons, are not limited to local effects, but can diffuse more widely through extracellular space, thereby bypassing the dendritic/axonal network. These latter, broad diffusion processes can be referred to as 'Volume Transmission' (VT) processes. Processing within the dendritic/axonal network can be referred to as 'wiring transmission' (WT) [24]. WT is rapid (from several microseconds to a few seconds), highly localised, signals pass between two cells, and the effects are phasic (i.e. event-related). VT is slow (from seconds to minutes/hours), global, has one-to-many signals, and the effects are tonic (extended over numerous events). VT may be the consequence of synapse leakage, open synapses, ectopic release (i.e. neurotransmitters released from the surface away from synapses), etc..

WT networks formed by neurons and their interconnections are the main information processing structure of the brain posited by the neuronal doctrine. Hence it is this interconnected structure of weighted links, integrators and action potential generators that has been regarded as the primary system of information processing and computation in the brain. This is also the model that has been adopted by simple artificial neural network models derived from the neuronal doctrine during the latter half of the twentieth century. The neuronal doctrine accommodates increasing sophistication in the understanding of how biological neurons and their networks function. Synapses are highly plastic, their plasticity being a major mechanism of learning, with synapses between co-active neurons being strengthened while synapses between neurons that are rarely active at the same time deteriorate. Most of the adult human brain does not undergo any significant new neuron creation (with the notable exception of the olfactory epithelium), but dendrites and dendritic connections, ion channels, dendritic spines and synapses undergo continuous ongoing changes. By these mechanisms, cortical neuron populations of a sufficient size appear to be capable of learning any specifiable function of an arbitrary number of dimensions (e.g. [5]).

Information processing by action potentials, their propagation through neuron networks, and the integrate-and-fire operation of individual neurons represents a one level of computation in the brain. Processing of information within the dendritic trees of single neurons has recently been proposed to represent local forms of computation *within* the dendritic structure, together with back propagation of spikes from the soma via dendrites, spike generation in dendritic spines and shafts, and bistable dynamics [13]. Computations conducted within dendral structures may include simple arithmetic, from simple to complex logic functions, filtering, and even integrate and fire functions within dendral substructures, creating a two-layered 'neuron' model (similar to simple classic ANNs) within a single neuron. The extended structure of dendritic trees means that different spatiotemporal input patterns can have different effects on neuron firing, allowing for computation of directional selectivity, in retinal and audio processing [13]. Moreover, Rall and Shepherd [16] proposed that two neuronal populations (excitatory and inhibitory) could communicate via direct synapses between their dendrites, without involving axonal propagation (see also [17]). While these forms of computation may be observed within dendritic structures, as London and Hausser [13] note, the key question is the extent to which the brain takes advantage of these building blocks to perform computations. For SI the question is that of the extent to which these mechanisms may contribute to HLI. Hameroff [10] even suggests that dendritic cross-connections provide the foundations of consciousness.

The changes in ion channels, dendritic and axonal tree growth and interconnections, and synaptic processes in response to network activity can be seen as *another* level of computation, and one more fundamentally associated with neural plasticity. It is also possible to model the *internal* processes of cells in terms of information processing and/or computation (e.g. [22]). This includes the effects of neurotransmitter reception, continuous metabolic processes, and interactions between these two. Hence computation/information processing occurs at the *intra*-cellular level, as well as at the WT and VT levels.

This picture of the neuronal doctrine makes information processing, or computation, within the brain complex and multi-levelled. In general there about 500 different types of human neurons. There are about 100 billion neurons in a single brain, each of which is connected to 1,000-10,000 others with over 200,000 km of axons [15]. Hence the WT network is highly complex, even without considering detailed mechanisms of intercellular communication, dendritic processing, synaptic processing, plasticity and intra-cellular processes.

However, neurons only constitute about 10% of brain cells. The rest consist of *glial* cells [24], of which 80% are *astrocytes*, the subtype of concern in this paper. For most of the time since the discovery of glial cells in the late 19th century they have been regarded as secondary support cells for neurons, e.g. providing nutrients and mopping up excess neurotransmitters. However, research over the last couple of decades has radically revised this understanding. It is now known that astroglia are the stems cells from which neurons differentiate. Those that remain as astrocytes form networks connected via gap junction bridges that provide intercellular communication, providing transfer paths for ions, metabolic factors and second messengers throughout the central nervous system (CNS). Astrocytes also engage in long distance communication by calcium wave propagation initiated by stimulation of neurotransmitter receptors in the astroglial cell membrane [24]. Astroglia appear to express all known forms of neurotransmitters, which can influence neuron activity, and they possess numerous ion channels that can be activated by extracellular and intracellular activity, such as the activity of neighbouring neurons [24]. Hence neurons and astroglia appear to form parallel and intercommunicating systems of signal transmission and processing. Glial cells also determine the differentiation, microarchitecture, synaptogenesis, and death of neurons and neural structures. Verkhratsky and Butt [24] hypothesise that neuronal networks are specialised for fast communication (i.e. metaphorically, they provide a kind of *internet* within the CNS), while astroglia provide the most

substantial information processing, integration and storage functions of the brain. Evidence for the significance of glia is found in their dramatic increase, both in absolute numbers and relative to the numbers of neurons, on a phylogenetic scale, reaching the greatest complexity in the human brain [24].

One further information processing system within neurobiological systems that will be mentioned in this paper is the system of hormones that also interacts with the processes described above. Hormones are chemicals secreted by specific groups of cells that are carried by the bloodstream to other parts of the body where they act on other cells to produce specific physiological effects [2]. *Neurosecretory*, or *neuroendocrine*, cells in the hypothalamus are almost the same as neurons, except that they do not release neurotransmitters, but instead they secrete hormones into the blood stream [2]. The effects of hormones on the body include reproductive development and rhythms, water and salt balance, growth, the secretion of other hormones, metabolic rate, emotional arousal, inflammation reactions, digestion and appetite control [2]. Hormones constitute a VT system in the terms used by Verkhratsky and Butt [24]. Hormones act gradually, change the intensity or probability of behaviours, are influenced (in type and quantity released) by environmental factors, have a many-to-many relationship with cells, organs and behaviours, are secreted in small amounts and released in bursts, may vary rhythmically in levels, may be mutually interacting, and are graded in strength (unlike the digital nature of neuronal action potentials [2]). Hormones, neural systems, behaviours and their consequences, are highly interactive and integrated. Hence an understanding of the processes and consequences of the nervous system, including the achievement of HLI, can only be complete when understood together with the parallel signalling and information processing system mediated by hormones.

3 NEUROPHYSIOLOGY AND COMPUTER SYSTEMS, ESSENTIAL DIFFERENCES

A fundamental question in the quest for synthetic HLI is that of which levels of abstraction or description represent the lowest level necessary for the realisation of HLI. Human levels of intelligence are poorly defined and poorly constrained. Cognitive science and cognitive psychology have made some progress in the top-down decomposition of human intelligence into functional parts and facilitating capacities. Knowledge of how cognitive constructs map onto brain structures and processes at lower physiological levels are being increasingly provided by correlation studies with neuroimaging, lesions, etc.. But since there are as yet no convincing demonstrations of synthetic or artificial HLI, it is not yet be certain where the limits of necessary detail are. There is also the conceptual question of the degree to which constructs such as *computation*, *information* and *information processing*, and which particular understandings of these constructs, are helpful (or not) in creating a working model of the relationship between physiological processes at different levels of abstraction/description and the achievement of HLI. Answers to this question provide foundations for considering which technologies may provide suitable media for the achievement of the required forms of computation and information processing. However, some differences between biological brains and computational machines as we know them may make *machines* as such incapable of achieving the intelligence demonstrated by biological brains irrespectively of issues of abstract computation and information models.

Potter [15] presents a number of differences between natural intelligence (NI) and AI, suggesting features that *could*, at least in principle, potentially make AI more brain-like. These include:

- brains don't have a CPU, they are highly distributed; NI uses lots of parallelism
- biological memory mechanisms are not physically separable from processing mechanisms
- biological memories are dynamic and continually reshaped by recall
- the brain is asynchronous and continuous; resulting phase differences can encode information. As noted by Crnkovic [4], the asynchrony of brain processing means that it does *not* conform to Turing computation.
- brains do not separate hardware from software; i.e. computation and information processing are not abstracted from the physical level, and the physical level is continuously changing (e.g. in mechanisms of plasticity noted above)
- NI thrives on feedback and circular causality. The nervous system is full of feedback at all levels, including the body and the environment in which it lives; it benefits in a quantifiable way from being embodied and situated.
- NI uses lots of sensors
- NI uses lots of cellular diversity
- delays are part of the computation. The brain computes with timing, not Boolean logic.

Further differences may also be noted, including:

- the brain is *analog*, where computers are *digital*. The digitisation of atemporal quantities leads to quantisation errors, and the quantisation of time leads both to potential quantisation errors and aliasing (the appearance of high frequency content in the form of illusory low level frequency components) although it is unclear how critical these errors are in functions underlying the achievement of HLI.
- neural systems are coextensive with the human body. This leads to layered and partially hierarchical control, achieved to a degree in some AI architectures (e.g. [3]).
- power distribution is decentralised and coextensive with information processing; hence human metabolic processes are implicated in information processing
- brains and nervous systems are intimately integrated with other bodily systems and processes. It may be reasonable to suggest that more abstract cognitive functions (e.g. abstract problem solving, mathematics) are understandable without considering parallel or underlying physiological processes. But even the most abstract brain operations are in practice constrained by factors in their physiological substrate (e.g. successful high level reasoning requires energy and sleep).
- much of the organisation of the brain is topological, from sensory processing to the contralateral organisation of the cerebral hemispheres
- physical locations can matter. A good example of this is the use of differential timing of the arrival of aural information

via separate dendrites as a cue for sound localisation (see [13])
- NI systems are built from the bottom up in processes that are self-assembling, self-organizing, and adaptively self-maintaining (characterised by Crnkovic [4] as self-* processes), based upon a ubiquitous (genetic) instruction set that is expressed in ways that vary according to highly local conditions and their recursively embedded contexts
- the foundations of NI have not been designed, but have evolved

The last two points are critical. There is currently no demonstration proof of the achievement of HLI in a way in which its mechanisms and contents are fully comprehensible within human consciousness. However, to achieve an SI that is capable of HLI should not require directly building a fully capable system. Rather, it can be based upon initiating processes of self-assembly and self-organisation that can create a sufficiently complex microstructure to achieve an adaptive, learning and growing nascent SI. The nascent SI must be capable of maturing through self-organisation in interaction with its environment to full HLI and beyond, just as in biological HLI.

4 NEUROPHYSIOLOGICAL PROCESSING AS INFORMATION PROCESSING

In considering the nature of the brain as an information processing system, it is necessary to be clear about what kind of information processing system it is, and according to what understandings of the term *information*. A widely used formal definition of information was first formulated by Shannon [19]. Shannon's concept of information can be summarised in the following way: if there are n possible messages, then n is a measure of the information produced by the selection of one message from the set, when all messages are equally likely. That information can be expressed by $\log_2 n$. This represents a number to the base 2 which can be represented by a sequence of bits (binary digits) of length $\log_2 n$, where any specific bit pattern of length $\log_2 n$ can represent a particular message among the set of n possible messages.

Shannon and Weaver [20] describe three levels of communication problems: "A. How accurately can the symbols of communication be transmitted? B. How precisely do the transmitted symbols convey the desired meaning? C. How effectively does the received meaning affect conduct in the desired way?" The mathematical theory of communication is concerned with A. This conception of information has been used in many analyses of neural system function, providing methods of measuring probability distributions, supporting analysis of information bottlenecks, and providing a view of cortical systems as systems that maximise information [1]. Information maximisation includes maximising the richness of representations, heuristic indentification of underlying causes of an input, to provide economies of space, weight and energy, and as a reasonable heuristic for describing models [1]. Potential disadvantages with the use of mathematical information theory including the need for vast amounts of data for generating reliable probability distributions, the need for independent sources of the usefulness of an encoding scheme, the uncertain nature of neural encoding of information, and the assumed stationarity of probability distributions known to an information receiver [1]. Nevertheless, it has been suggested that the overall goal of the brain and individual neurons is to minimise uncertainty, which corresponds with the maximisation of information. The 'neurocentric' approach of Fiorillo ([6], [7]) proposes that the activity of individual neurons can be fully described in Bayesian terms grounded in information theory, where a single neuron integrates information from molecular sensors to reduce uncertainty about the state of its world. In this case, the state of the world is the local environment of the neuron, where information input to the neuron is a property of biophysical mechanisms from the level of single molecules and up, rather than being inferred by a scientific observer from external properties of the environment. Hence the computational goal of the nervous system is the minimization of uncertainty (of the brain about its world) exclusively based upon the information and mechanics of the system, a view closely related to Friston's [8] theory of free energy minimization but with an internal rather than external view of information sources and their resulting probabilities.

Fiorillo [6] emphasises that the neurocentric approach uses probabilities only to *describe* the biophysical information of a neuron. "There is no physical step that must occur within the nervous system to "calculate" probabilities from information. Probabilities are a quantitative property of information in much the same way that mass is a quantitative property of matter. Likewise, information is an intrinsic property of matter and energy. Information follows the rules of physics, and Bayesian principles allow us to quantify information using probabilities." Crnkovic [4] goes further than this, proposing that "Information may be considered the most fundamental physical structure. Info-computationalist naturalism understands the dynamical interaction of informational structures as computational processes." This view goes much further than Shannon's view of information, which is essentially an epistemological one, to one that equates the epistemological with the ontological. The further development of this view is *morphological computation*, where biological computation is conceived as a "computational mechanism based on natural physical objects as hardware which at the same time acts as software or a program governing the behavior of a computational system" [4]. Morphological computation captures many of the essential differences between biological brains and semiconductor-based computers noted earlier in this paper.

5 CONCLUSIONS: IMPLICATIONS FOR SYNTHETIC INTELLIGENCE

All of the hierarchical levels of the human nervous system have been simulated in detail using digital computer technology [22]. However, both cognitive and neural simulations have had limited success to date and certainly fall far short of HLI. Lindley [12] has argued that cognitive approaches in particular do not appear to be promising in isolation. Here it may be added that an essential problem for cognitivist AI is an implicit Cartesian dualism, where AI has focussed upon modelling the mind, and the symbol grounding problem, a central problem for AI, is the computationalist's version of the problem for dualists of the nature of the 'mechanism' by which the mind and the body interact. In the spirit of Rorty [18], it is possible to shift the

emphasis away from the nature of mind towards the physiological foundations of the generation and use of mentalist concepts. The focus then shifts to the physiological foundations of human intelligence, the simulation of which has also not yet demonstrated anything remotely approaching HLI. However, the physiological project has two major advantages over the neo-Cartesian cognitivist project. Firstly, as considered in this paper, neural computation includes many broad frontiers of ongoing knowledge development, including cellular, sub-cellular and molecular processes, the role of dendritic computation, the role of astroglia, and the embedded interdependencies between brain systems, bodies and their contexts. Simply put, we understand biological intelligence so incompletely that it provides an ongoing wealthy source of methods, principles and foundations yet to be comprehensively understood, let alone transferred into the *design* of intelligence as an engineered artefact.

The second great advantage of the physiological project as an *engineering* project is that it is no longer limited to twentieth century engineering media. It is possible to apply molecular regulation, transgenic and viral techniques to selectively modify neurons and neural populations to generate "designer dendrites" [13] and other neural structures having specific computational properties and topologies. It is also possible to explore the creation of brain system analogs in different chemistries, including organic and inorganic systems different from 'wild' neural biochemistry. These systems can implement the analog, asynchronous and necessarily self-* characteristics of biological nervous systems in ways that are not possible with digital simulations. In implementing the morphological computing foundations of biological intelligence, these systems can be realisations of synthetic intelligence, rather than simulations or metaphors.

REFERENCES

[1] R. Baddeley Introductory Information Theory and the Brain. *Information Theory and the Brain*, R. Baddeley, P. Hancock and Peter Földiák, Eds., Cambridge University Press, 2000.

[2] S. M. Breedlove, N. V. Watson and M. R. Rosenzweig *Biological Psychology*, 6th Edn., Sinauer Associates, Inc. Publishers, Sunderland Massachusetta, 2010.

[3] R. A. Brooks (1999) *Cambrian Intelligence*, MIT Press.

[4] G. D. Crnkovic: Info-computationalism and Morphological Informational Structure. *1st Annual Conference on Integral Biomathics*, Stirling University, Scotland, 29-31 August 2011.

[5] C. Eliasmith and C. H. Anderson (2003). Neural Engineering: Computation, representation and dynamics in neurobiological systems. MIT Press.

[6] C. D. Fiorillo. A new approach to the information in neural systems. *1st Annual Conference on Integral Biomathics*, Stirling University, Scotland, 29-31 August 2011.

[7] C. D. Fiorillo. Towards a general theory of neural computation based on prediction by single neurons. *PLoS ONE* 3: e3298, 2008.

[8] K. Friston. The free energy principle: a unified brain theory? *Nat Rev Neurosci* 11: 127-138, 2010.

[9] R. Granger R Engines of the Brain: The Computational Instruction Set of Human Cognition, AI Magazine , 27:15-32, 2006.

[10] S. Hameroff. The "conscious pilot"—dendritic synchrony moves through the brain to mediate consciousness. *Journal of Biological Physics* Vol. 36 Issue 1. DOI: 10.1007/s10867-009-9148-x. Published: 2009-12-10

[11] E. R. Kandel, J.H. Schwartz, T.M. Jessell. *Principles of Neural Science, 4th ed.* McGraw-Hill, New York, 2000.

[12] C. A. Lindley "Synthetic Intelligence: Beyond A.I. and Robotics", *1st Annual Conference on Integral Biomathics*, Stirling University, Scotland, 29-31 August 2011.

[13] M. London and M. Hausser, *Dendritic computation*, Annu. Rev. Neurosci., 28 (2005), 503–32.

[14] Newell A and Simon HA (1975) Computer Science as Empirical Inquiry: Symbols and Search. CACM 19:113-126.

[15] S. M. Potter What Can AI Get from Neuroscience? M. Lungarella et al. (Eds.): *50 Years of AI*, Festschrift, LNAI 4850, pp. 174–185, 2007. Springer-Verlag Berlin Heidelberg.

[16] W. Rall and G.M. Shepherd. Theoretical reconstruction of field potentials and dendrodendritic synaptic interactions in olfactory bulb. *J. Neurophysiology*, 3l (1968), 884–9l5.

[17] J. Rinzel. Distinctive Roles for Dendrites in Neuronal Computation. *SIAM News*, Volume 40, Number 2, March 2007

[18] R. Rorty. *Philosophy and the Mirror of Nature*. Princeton: Princeton University Press, 1979.

[19] C. E. Shannon: A Mathematical Theory of Communication, *Bell System Technical Journal*, Vol. 27, pp. 379–423, 623–656, 1948.

[20] C. E. Shannon and W. Weaver: *The Mathematical Theory of Communication*. The University of Illinois Press, Urbana, Illinois, 1949.

[21] L. R. Squire, F. E. Bloom, N. C. Spitzer, S. du Lac, A. Ghosh, D. Berg,. *Fundamental Neuroscience*, Third Edn., Academic Press, 2008.

[22] D. Sterratt, B. Graham, A. Gillies and D. Willshaw. *Principles of Computational Modelling in Neuroscience*, Cambridge University Press, 2011.

[23] A. M. Turing. Computing Machinery and Intelligence. *Mind* (59): 433–460, 1950, Oxford University Press.

[24] A. Verkhratsky and A. Butt. *Glial Neurobiology: A Textbook*. John Wiley and Sons Ltd., 2007.

Learning to Hypercompute? An Analysis of Siegelmann Networks

Keith Douglas[1]

Abstract. This paper consists of a further analysis (continuing that of [11]) of the hypercomputing neural network model of Hava Siegelmann ([21]).

1 INTRODUCTION

This paper consists of a further analysis (continuing that of Douglas [11][2]) of the hypercomputing neural network model of Hava Siegelmann ([21]). It consists of three large sections. In the first section, a brief description of Siegelmann's model is presented. This section will be followed by a discussion of the merits of taking this model as a factual model (pace the "abstract" approach of Sieg [20]). Third, a discussion of one of Siegelmann's key, heretofore poorly explored, assumptions (the "linear precision suffices" claim) will be addressed and is the primary focus of the paper. This discussion will proceed along the following three subsections of analysis: it will discuss (1) a not-fully "noticed" suggestion of Arlo-Costa ([1]) and Douglas ([11])[3] that the Siegelmann network model actually requires a supertask to perform; (2) the merits of treating Siegelmann's proposal as one involving an idealization in the sense of Norton ([15]). The latter two will also allow a brief discussion of a factual interpretation of the arithmetic, analytic and similar familiar hierarchies; (3) that pace Davis and Scott ([9]) "non-recursive black boxes" are not exactly untestable, making use of the work of Kelly and Schulte ([16]). Subsections (2) and (3) are not independent and yield similar findings.

I end with summary conclusions. The conclusions will largely be more negative and "skeptical" about the merits of the Siegelmann network than those of herself or some of those (e.g. Copeland) who have defended it, but hope to provide more details on the areas of the model where interested parties could work on improving its plausibility and (nomological) possibility.The paper is organized as follows.

2 SIEGELMANN NEURAL NETWORKS

Hava Siegelmann's monograph ([21]) is the principle source of detailed discussion of her model (hereafter SN), and includes much information about the computational strengths of various forms of neural networks, their properties from the perspective of computational complexity and much off topic for our present purpose. Subsequently here, we only need to focus on aspects that are unique or unusual in her presentation with regards to the question of computability. I assume the reader is familiar enough with usual artificial neural network models (hereafter, ANN) to follow the discussion involving such matters as nodes and weights (see, e.g., [8]).

These unique/unusual aspects are: (1) the necessity of using an "output protocol", (2) her claims about the (real-valued) weights in the network and (3) the "sensitivity" of the network, i.e., a matter of interpreting the activation function.

This section of the paper will simply remind or inform the audience of these features as they do play a role later on, and not critically discuss them completely at this point. I bring these up to show that there are additional problems with SNs not discussed as much as the problem of weights already familiar in the literature and because they will play a crucial role in the discussions of idealizations later on.

In the case of "output protocol", what is meant is the convention adopted by Siegelmann (see in particular, [21], pp. 23-24) to indicate when a SN is finished its computation and is returning the value so calculated to its users. A state flag called the "output validation line" is set to 1 and is held in this value for the duration of the significant output and is set and held at 0 at all other times. One can then, during the time this line is 1, read off the value of the output from another, more conventional, output line. The "hypercomputing" part of this proposal is somewhat significant and yet hidden in her presentation.

In particular, how does this flag get set? Take the case of a recursively undecidable problem, for which these networks are supposedly useful at solving, like the halting problem for Turing machines (hereafter, TMs). In this case the output is a single encoded output, so in this case, the flag will be set to 1 at one "tick"[4] sometime in the future while (say) 1 comes out of the output line if the encoded TM halts and 0 otherwise. How does one know how to set this flag as a "programmer" of one of the networks? This depends on how the function is calculated, presumably. One has to know that the calculation is finished, that whatever state the network is in is the correct one. But this itself is a hyper-computational task; and so a regress seems to threaten. Moving on then to the case of the real valued weights of the network. This feature is the root of the hypercomputational power of the SN. Siegelmann does not tell us how these are to be obtained; merely calculates approximately how many digits of precision are needed after a given amount of run time. Since (by hypothesis) the network does not use registers, it is unclear what gaining digits of precision could refer to. An unspecified learning procedure is appealed to for the source of this extra precision, but without details this is simply

[1] philosopher.animal@gmail.com

[2] This paper is dedicated to the memory of Horacio Arlò-Costa and, of course, to that of Alan Turing, who I would like to think would be astonished by the amount of work building on his we have today and by which the results are so omnipresent.

[3] I am not claiming such should have been noticed, per se, but I find it strange that these considerations have not made it into the discussion of our topic by critics of hypercomputing (or considered as a "devil's advocate" objection by proponents).

[4] This assumes the network is somehow equipped with a clock (which at least some ANNs do not have), but in the interest of manageability of this paper, I'll simply grant this part of the SN ex hypothesi.

an unknown as well. Notice that there are two concerns here - both the learning procedure and how its use gets "recorded", "stored", etc. are at stake. As for the activation functions, their "embodiment" or "implementation" also raises unanswered questions. For example, a threshold function (as the name suggests) is typically understood to be some representation of a node's sensitivity to its inputs. In the case of SNs, these must be infinitely sensitive. Take a threshold function of the form (all of them discussed have the same problem; but since Siegelmann places special emphasis on the truncated linear one I use it here):

$$f(x) = 0 \text{ if } x < 0$$
$$= x \text{ if } 0 <= x <= 1$$
$$= 1 \text{ if } x > 1$$

To see the potential concern, consider a value of x = 0 + e, where e is some small value approximately (but not exactly) equal to zero. Represented in the usual notation, this is then some value 0.00000000000000000...1, say. The network has to be able to be able to "recognize" that value, no matter how small its difference is from 0, because the value of the output depends on it[5]. Siegelmann emphasizes truncation or rounding reduces the value represented at at a node to a rational value and hence renders the computational properties of the network nonhypercomputational. I call the property of the nodes in question "sensitivity", and as we have now seen, this is infinite in a real valued network (which allows literally any real value as a weight). Previous critics have pointed out the implausibility of finding (or knowing) that one had a hypercomputable weight in a SN (e.g., [9]); it is hopefully now clear that the problem is at least twice that, since one also needs a way for the network to make use of it, and that requires a "hypersensitive sensor" or something of the kind - subsystems that respond in infinitely precise ways to embody the activation functions. I might add in passing that this mistake or oversight is nothing new. Bunge ([5]) argues that a human brain is not suitably modeled by a TM because even a single electron can be in a continuum of states. Ignoring that this might prove too much, Bunge, like Siegelmann, has to argue that there can be an infinite number of (hyper)computationally (or, in Bunge's case[6], cognitively) relevant states and events (state transitions: [10]).

[5] Consider the required difference in output from two nodes that differ in value by 2e (e.g., one 0+e and the other 0-e). One of these will have activation 0 and the other e. It is also interesting to reflect that a relatively informal presentation of ANNs like in [8] the weights are also described as being real-valued, but nothing in their presentation hinges on it. Presumably it makes explaining the mathematics easy and ensures that a digression about computable PDEs is irrelevant. Presumably also Churchland and Sejnowski regard the plausibility of any real number as a weight to be not worth considering. Note also that the learning algorithms they discuss (pp. 96 ff.) are computable as well.

[6] I will not press the point here, but from my experience in conversation with Bunge (in the late 1990s), he does not think human brains are hypercomputers: rather, he thinks that computational notions are inapplicable to them altogether. The view, although he would be horrified by the comparison, seems to be similar to that of Wittgenstein. But this is all for another time.

3 SIEG (INDIRECTLY) ON SIEGELMANN

Sieg ([20]) has argued (in the context of a discussion of the Church-Turing thesis) that one can dispense with said thesis and instead:

"My strategy, when arguing for the adequacy of a notion, is to bypass theses altogether, and avoid the fruitless discussion of their (un-)provability. This can be done by conceptual analysis, i.e., by sharpening the informal notion, formulating its general features axiomatically, and investigating the axiomatic framework."

This viewpoint dispenses with the need to analyze the Siegelmann network in detail, at least for the present purposes - were it correct. It would make it clear that hypercomputation is doomed to failure as a subject as the axiomatic framework in question makes it perfectly clear that broadly computational devices (including potential hypercomputers[7]) do not include anything like the SN[8].

However, as has been pointed out by Stewart Shapiro ([19]), it does not appear that Sieg successfully dispenses with theses here. In other words, there is the question of whether or not the axioms are correct[9] of the real (or non-abstract, non-Platonic, etc.: replace as necessary according to your favourite philosophy of mathematics) systems under consideration. How do we (hopefully) resolve this impasse? For if Sieg is right, there is nothing to investigate; we simply see that the SNs do not fall under the axioms of computing machines he has usefully provided and that would be the end of it. This seems too hasty for the present purpose, so the concern is pressing.

Here is where Shapiro is mistaken; he thinks that (following Quine [18] and others) one is dealing with some matter which is both mathematical and empirical. For some (perhaps for Sieg) this is impossible or unlikely; instead it is like investigating axioms for (say) groups[10]. If Sieg were right it would be a matter

[7] Nothing in the Sieg-Gandy presentation actually rules out accelerated Turing machines ([2]) for example. However, it is unlikely at best that either Sieg or Gandy would approve; the advantage to the SN over many models of computation is that it explicitly includes a clock (a feature it admittedly shares with some ANN models) and thus can be used to more precisely make claims for or against "tricks with time" like the accelerated Turing machine requires. I'd hazard a conjecture that such a machine also requires no lower bound on the size of its parts if described as a Sieg-Gandy machine, and hence runs afoul of the finiteness requirements that way, but such an argument would require delicate physical hypotheses I do not wish to address in the present work.

[8] Since I disagree with Sieg that this approach is suitable, I shall not investigate precisely (in the present paper) where Sieg-Gandy machines rule out SNs, however it seems likely they run afoul of the "finite parts" conditions. Sieg and Gandy represent parts by the hereditary finite sets, so, presumably, a similar approach to the SN would need to use hereditary countable sets. This seems to suggest either or both of an infinite number of parts or an infinite magnitude of a property of one.

[9] I suspect that Sieg would claim that there is no thesis involved here; one simply investigates whether or not the axioms are fruitful, lead to desired consequences, etc. However useful that approach is for his and many other very important purposes, it amounts to begging the question against hypercomputation without further ado.

of getting (as he borrows a phrase from Hilbert in saying) the "Tieferlegung der Fundamente" right; Shapiro claims instead one has to look to the world too. I claim both are mistaken because they have overlooked the possibility that the matter is not about mathematics at all.

I argue that the debate should be construed as one about doing mathematics (or at least doing calculations or computations). Turing, as Sieg has rightly emphasized, analyzed a human "computor" (in the sense of Gandy [13]). Similarly, Gandy, him, and others have analyzed calculations by machine as well. Using Bunge's ([3]) theory of reference and Sieg's admirable presentation ([20]) of "Gandy machines", one sees that the theory of Gandy machines is, indeed, about computing machines. This makes the subject matter a factual[11] one in Bunge's sense[12]; see also Douglas [12]. In other words, it is not a matter of mathematics - one can (and should) use mathematics as a tool to describe the characteristics of the computers and computors, but this does not make the field mathematics anymore than using differential equations in the theory of reaction mechanisms makes chemistry a branch of mathematics.

Hence Shapiro is right in his claim: Sieg does not dispense with theses - or, if preferred, Church's thesis is in need of "empirical" confirmation and hence SNs' "usefulness" as a model of computing cannot be dismissed so hastily. Also hence in particular, we must address the question of whether SNs are empirically plausible. It is here that we run quickly into previous criticisms of her proposals from the likes of Martin Davis and Dana Scott.

Davis' ([9]) paper quotes Scott concerning how we would recognize a "nonrecursive black box". I feel this quotation is also slightly mistaken: it proves too much. I agree that no finite amount of interaction with a black box could show that it performs hypercomputational processes. However, no finite amount of observation could tell you that a black box contains a Turing machine. Any finite experimentation with input and output is consistent with the black box being a (perhaps very large) finite state automaton[13]. This is not to say Scott and Davis are mistaken concerning the difficulty of determining that one has a hypercomputer of some kind, but instead that it is important not to overstate this difficulty. He emphasizes how hard it would be to tell that one had a non- recursive "transparent box" (i.e. a black box with much of its workings well known). It seems to me that Scott and Davis have adopted almost an instrumentalist attitude towards (what Bunge would call factual) scientific theories here. Since instrumentalism is controversial amongst philosophers of science, we should be wary of this approach[14]. After all, how does (say) Newtonian dynamics (ND) get verified? This presumably factual theory uses continuous functions and such; whereas any measurement is only of finite precision and hence renders direct confirmation impossible. Davis and Scott thus "prove too much" with this approach. They might rejoin that one could state ND in terms of computable analysis. However, assuming it could be done in this case does not show it could be done in general. Also, since the theory is then different, how does one decide between the computable version and its (usual) noncomputable counterpart? It would seem one would have to apply more general principles about the nature of theories in (factual) science. Since these are arguably under debate, we are now back where we started.

Nevertheless, Davis and Scott have correctly (in my view) treated SNs as to what sort of proposal they are - namely a family of factual hypotheses. I have mentioned earlier (section 1) that there are what one might call "nomological" areas of discussion (problems, conterproposals, etc.) with the SN approach. I now turn to three of these.

3 NOMOLOGICAL CONSIDERATIONS ABOUT "LINEAR PRECISION SUFFICES"

The first of these stems from Arlò-Costa ([1]) and adapts prior remarks of Douglas ([11]) to that end. He asks whether or not the SN require a supertask to implement and hence "inherit" the implausibility of the accelerated Turing machine (see, e.g., Boolos and Jeffrey [2]) which most would agree is a purely "notional" device. In particular, note the difficulty even in computing a constant function with a SN. Since the weights of each node in a SN are of infinite precision, outputting their value directly is impossible by the protocol described. This arises because such a constant is still an infinite precision number, and so outputting its value requires an infinite amount of time[15], followed by a signal to indicate that the output is finished. At best this would require a supertask. A suitable re-encoding[16] would have to be found, and that is not suggested anywhere by Siegelmann. Moreover, such would have to handle rational

[10] Using an algebraic analogy here, as opposed to (say) using a geometric one is important. By contrast, say, analysis would lead to questions immediately about the "real" continuum and whether spacetime is or could be discrete; geometry raises similar questions about dimensionality, curvature, etc.

[11] Bunge ([3], [5]) is a mathematical fictionalist and contrasts factual to formal sciences; once again one can translate into one's appropriate philosophy of mathematics idiom. The important matter is that group theory is not the correct analogy; instead, a theory in (say) chemistry - like (say) a theory of solutions - is a better comparison. Using physics would raise questions about "rational mechanics" that might prolong the debate unnecessarily and other sciences would raise equally irrelevant questions for our present purpose. I will use "factual" in his sense throughout.

[12] Bunge (see, e.g., [4]) would claim that this use of "empirical" (traditional in most philosophy of science) is wrong, however, I shall use it here to emphasize what I intend.

[13] Matters are actually not quite this simple. See below about [15]).

[14] Disclosure: As may be noticed, I am a scientific realist (of a somewhat unnamed sort), so I have (what I take to be) good reasons against instrumentalisms. But to be charitable to such esteemed scholars as Scott and Davis, I have tried to avoid dismissing their seemingly instrumentalist views out of hand and tried to find a way to allow both them and Siegelmann the benefit of the doubt about the plausibility of certain hypotheses.

[15] That is, unless one could in every case "program" the Siegelmann network to tell when it had an irrational number and flag rationals appropriately. This ability itself seems to be hypercomputational.

[16] Siegelmann's book spends a lot of time talking about Cantor sets and changes in number bases, etc. As far as I can tell, qua engineering proposal (and one does take SNs as such when one takes them factually, as we are doing) this is largely irrelevant without knowing what physical properties does the representation in the engineering sense. Obviously no registers are involved, and so re-encoding is not well defined at present. This problem is (needless to say) another instance of the same one that we keep encountering: how *do* the weights work?

values as well as surds, transcendental values, and even non-Turing computable numbers, like Chaitin's constant. Of course, giving a finite representation of the latter sort of value cannot in general be done. My earlier remarks about the "output protocol" loom large here.

Similarly, if the precision of an infinitely precise real number is not available at the beginning of the run of a SN, and the precision increases uniformly in time (see further my discussion of "learning networks") it will take an infinite amount of time for the network to become infinitely precise. This entails immediately that the networks are actually Turing-equivalent for any finite period of time. Here, let me note further that this puts Siegelmann's model in an unfortunate dilemma much as the precision consideration proper above provokes. Once again, either the network is infinitely precise in finite time (throwing away the "linear precision suffices" result), in which case the Siegelmann network is implausible from the sensitivity considerations I have canvassed and from related concerns, or it is only infinitely sensitive in infinite time, in which case using it to perform super-Turing computations would again require a supertask. Thus, it seems that Arlò-Costa is in fact correct, though a proof would be nice to have - but in the interests of time I have omitted such. I thus turn to a question which stems also from [11]).

This concerns the nature of idealizations and approximations. It might be argued that the critics of SNs are taking the model too literally. Instead, it should be treated as one involving either an idealization or an approximation. For example, a SN-fond opponent of the critics of hypercomputing may well point out the critic will ask: why should we not grant relevant idealizations to the Siegelmann network? after all, the TM itself (or, equally, a "Gandy-Sieg machine") makes idealizations concerning computing agents and their resources. For example, these are held to have an unbounded amount of memory, do not break down ever, can calculate without running out of energy no matter how long they run, etc. So, the opponent asks, why not grant idealizations to the SN?

One could attempt to respond to this opponent by (1) counting idealizations or (2) intuitively trying to evaluate their merits and plausibilities. In the case of (1) it is likely correct to conclude that the SN includes all those of the TM and then some, it does not seem fruitful to simply claim that the SN has more and hence is more implausible, for what if the TM was already regarded as sufficiently implausible to not merit adoption? Or, in other words, does this objection prove too much? Also, how does one know what is "too many" idealizations and approximations anyhow? Better to look at (2), instead.

To focus attention, let us discuss one particular family of idealizations, that of the weights of the nodes[17] in the SN. Norton ([15]) has circulated a manuscript on idealizations which is useful to apply to the present purpose (cf. also the brief remarks in [14] and [7]). Norton's paper centers around what he has called "the problem of limits", distinguishing between the case when the limit property and the limit system agree on the one hand, and when there is no limit system on the other. I shall argue, based on considerations based on the arithmetic hierarchy (e.g., [16], pp. 362 ff.) that Siegelmann's proposal falls into the latter category. This is because at any finite time, a Siegelmann network is equivalent to a TM in power; only "at" the limit of infinite time is the network super-Turing in power.

To adopt Norton's analysis, one thus has to identify the limit property of a SN and what the limit system could be (if there is one). As just suggested, the limit property is the node weight (it does not matter if we take all the nodes or one, as by interleaving their expansion one can see there is effectively only one "weight" anyway[18]). Let us also assume for definiteness that this weight is Chaitin's constant, ξ[19]. Then the question becomes how the weight gets used in an idealization. Here I do not know what to do to proceed. The weight is still not explained: in order to evaluate the idealization we have to know what this property actually is. For the sake of definiteness again, assume that we are dealing with a length. Then the idealization involves that of limiting lengths, the idealized value of which is ξ. This is a case where the limit property is not a property of the limit system. This is because the increases in length precision do not correspond at all the steps on the arithmetic hierarchy: if the idealization here were plausible then each finite increase in precision would correspond to some finite n in the usual labeling of the hierarchy[20]. One simple "imaginable possibility" would be to think that a weight of precision n corresponded to a "strength" of $\Delta^0_{f(n)}$ where f(n) is some finite (i.e., non-divergent) function of n so that an infinite precision would deliver Δ^0_ω as required[21]. But this does not work, no matter how slowly the function f grows, as Δ^0_0 is already the recursive functions. In that sense, since each finite n is still sub-recursive, the limit in

[17] There is an "informal duality" between the weights and what I have called "sensitivity". All the arguments I raise, as far as I can tell, apply to it as well.

[18] For example, taking the digits from each weight value: 0.a1a2a3a4... and 0.b1b2b3b4... becomes 0.a1b1a2b2a3b3a4b4... in the case of a two node network. A similar procedure can be done for any SN, as they all have a finite number of nodes.

[19] A referee asked if we know that this particular constant is calculable. Since it is representable as a function from natural numbers to natural numbers, it is. SNs can can calculate all such functions ([21], pp. 59 ff.).

[20] It is vital not to get the order of this (very impressionistic) proposal wrong. Since SNs grow in computational by increasing precision, and in the conventional theory of computation, more computational power means "climbing" the arithmetic hierarchy, all I am suggesting is how one would have to reconcile these aspects.

[21] There is some potential confusion here, so a clarification is true. It is quite correct to point out that this level of the hierarchy is "infinitely more" than is needed to do super-Turing computations. But precisely what is wrong (in one way) with the SN proposal is that it skips that entire part of the hierarchy. Why? Because that's precisely what a jump from finite precision real numbers (i.e., rational weights) to the full infinitely precise weights of the SN does. On the one hand one needs the infinite precision; on the other hand the jump is an inappropriate idealization for that reason. There is no way an SN, as described, can climb through the hierarchy over finite time and wind up, in the infinite limit, at the ability to calculate all functions from natural numbers to natural numbers. Any finite increase in precision adds at best a finite ability. It is not as if some magic bit, say at the 31337th decimal place, suddenly allows all the sigma 0 1 functions to be computed, etc.

question of the SNs is then the recursive functions and not the arithmetic ones the SN need, never mind the analytic ones (i.e. all the functions from natural numbers to natural numbers). Hence, the property of the limiting system and the limit property do not agree. Hence further it looks like an inappropriate idealization in Norton's sense. We thus have a clearer way of stating the difference over the idealizations of the SN versus those of the TM.

These brief appeals to the arithmetic hierarchy also allow an answer to Scott and Davis (above) and avoid protracted debates over operationalism. A non-recursive "oracle" is indeed hard to investigate, however, Kelly and Schulte ([15]) draw important connections between the arithmetic hierarchy and the learning of theories with uncomputable predictions. While I will not prove any results here, I suggest that rather than an "operationalist" response to Siegelmann, one can in principle give a learning-theoretic answer at least for some possible uses of the network. The goal in this section is to answer residual worries about operationalism and merely gesture at an area of future investigation, particularly connecting the properties of Siegelmann network to other hypercomputing proposals as has been suggested by the Wikipedia contributors ([22]).

Let us turn to specifics. Kelly and Schulte ([15]) classify (following Gold and Putnam) hypercomputational problems into learning theoretic classes. For example, a hypothesis of the form Π^0_1 is one which is "refutable with certainty". However, what is interesting from the perspectives of this paper (and symposium) is that a Δ^0_3 sentence is sufficiently "complex" that there is no way to investigate it in a computable way[22]. This is "infinitely far" away from the level $\Delta^0_\omega \quad \left(=\Delta^1_0\right)$ that characterizes the complete SNs. However, a brief look at learning again might prove useful. If the increase in precision of real valued weights increased through the learning-theoretic hierarchy in a useful way - say, some fixed bound moved the strength of the system from Δ^0_1 to Π^0_1, that would be a useful finding. Unfortunately, it seems to be nowhere in offering, once again for the same reason. To reiterate: any finite bound in increase of precision of the networks preserves their behaviour vis-a-vis the arithmetic hierarchy. This makes it implausible, to say the least, that SNs could increase their precision in a relevant way by "learning" as she proposes (without a supertask). This is not to say that real valued weights in a network could not increase precision by external influence (learning) but rather that they could not do so in a relevant way that makes hypercomputation plausible (or nomologically possible).

The lesson for this subsection is then: Scott and Davis are right to be skeptical of our abilities to investigate purported capabilities of a supposed non-recursive black box. However, they are wrong to say that it is impossible in principle, but SNs provide no way for this investigation to proceed. Partisans of hypercomputation wanting to answer Scott and Davis must look elsewhere (including refining their models).

[22] Claiming to investigate it in a hypercomputable way would of course beg the question against the critic of the SNs and also be useless for the proponent of them. After all, if one has a known "Δ^0_ω device" or procedure already, why use a SN?

4 CONCLUSIONS

Investigation into the arithmetic hierarchy-related properties of Siegelmann style networks show how they are implausible relative to a Turing machine model of computation for they invoke various versions of the same inappropriate idealization. In future work, I hope to discuss whether any models of hypercomputation meet these requirements. I also hope to provide more details in these areas of specific criticism and also more rigorously analyze the Turing machine model from the perspective of idealizations.

REFERENCES

[1] H. Arlò-Costa. Unpublished comment in discussion with the author. 2003

[2] G. Boolos and R. Jeffrey. *Computability and Logic*. (2e) New York: Cambridge University Press. 1980.

[3] M. Bunge. *Sense and Reference* (Vol. 1 of *Treatise on Basic Philosophy*). Reidel: Dordrecht. 1974.

[4] M. Bunge. *Epistemology and Methodology I: Exploring the World*. (Vol. 5 of *Treatise on Basic Philosophy*). Reidel: Dordrecht. 1983.

[5] M. Bunge. 1985a. *Epistemology and Methodology III: philosophy of science and technology: Part I. Formal and Physical Sciences* (Vol. 7 of *Treatise on Basic Philosophy*). Reidel: Dordrecht. 1985.

[6] M. Bunge. *Epistemology and Methodology III: philosophy of science and technology: Part II: Life Science, Social Science and Technology* (Vol. 7 of *Treatise on Basic Philosophy*). Reidel: Dordrecht. 1985b

[7] M. Bunge. *Philosophy of Science* (2 vols.) New Brunswick: Transaction. 1998.

[8] P. Churchland and T. Sejnowski, Terrence. The Computational Brain. Cambridge: MIT Press. 1996.

[9] M. Davis. (adopting remarks of D. Scott). "The Myth of Hypercomputation". In Christof Teuscher (ed.) *Alan Turing: Life and Legacy of a Great Thinker*. Berlin: Springer. 2006.

[10] K. Douglas. "A Special Davidsonian Theory of Events". Unpublished MA Thesis, Department of Philosophy, University of British Columbia. 2001. Available online at http://www.philosopher-animal.com/papers/thesis.pdf.

[11] K. Douglas. "Super-Turing Computation: A Case Study Analysis". Unpublished M.S. Thesis, Carnegie Mellon University. 2003. Available online at http://www.philosopher-animal.com/papers/take6c.PDF

[12] K. Douglas. "What Does a Computer Simulation Have to Reproduce? The Case of VMWare". Unpublished paper presented at IACAP 2010.

[13] R. Gandy. "Church's Thesis and Principles for Mechanisms." In Barwise, John, Keisler, H. Jerome., Kunen, Kenneth. (eds). *The Kleene Symposium*. Amsterdam: North-Holland. 1980

[15] K. Kelly and O. Schulte. "The Computable Testability of Theories Making Uncomputable Predictions". Erkenntnis 43: 29-66.

[14] I. Niiniluoto. 2004. *Critical Scientific Realism*. Oxford: Oxford University Press. 2004 (1999).

[15] J. Norton. "Approximation and Idealization: Why the Difference Matters". 2011. Unpublished manuscript, available at http://www.pitt.edu/~jdnorton/

[16] P. Odifreddi. *Classical Recursion Theory: The Theory of Functions and Sets of Natural Numbers*. Amsterdam: Elsevier Science. 1989.

[17] A. Olszewski, Adam. J Woleński and R. Janusz (eds.). *Church's Thesis after 70 years*. Heusenstamm: Ontos Verlag. 2006.
[18] W. V. O. Quine. "Two Dogmas of Empiricism". The Philosophical Review 60: 20-43. 1951
[19] S. Shapiro. 2006. "Computability, Proof, and Open-Texture". In [17]. 2006.
[20] W. Sieg. "Calculation by Man and Machine: Mathematical Presentation". Technical Report No. CMU-PHIL-105. 2000.
[21] Siegelmann, Hava. 1999. Neural Networks and Analog Computation: Beyond the Turing Limit. Boston: Birkhäuser.
[22] Wikipedia contributors. "Hypercomputation" in Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/wiki/Hypercomputation (accessed January 29, 2012).

Oracle Turing machines faced with the verification problem

Florent Franchette [1]

Abstract. The main current issue about hypercomputation concern the following thesis: it is physically possible to build a hypercomputation model. To prove this thesis, one possible strategy could be to physically build an oracle Turing machine. More precisely, it is about finding a physical theory where a hypercomputation model will be able to use some external information from nature. Such an information could be regarded as an oracle that provide an additional element in order to go beyond Turing machines limits. However, there is a recurring epistemological problem about the physical construction of an oracle Turing machine. This problem called "verification problem" may be worded as follows: if we assume we have such a hypercomputation model physically built, it would be impossible to verify that this model is able to compute a non Turing-computable function. In this paper I will propose an analysis of the verification problem in order to show that it does not explicitly dispute the strategy about a physical construction of an oracle Turing machine.

1 INTRODUCTION

Alan Turing is widely known in logic and computer science to have devised the computing model today named "Turing machine". In computer science because the Turing machine is the theoretical model of modern computers and in logic because it is the formalization of a computable function [14].

Nonetheless Turing is also behind the oracle Turing machine, a Turing machine equipped with an "oracle", namely a black box whose behaviour is not specified, which is able to provide some non computable functions results [15, p. 167]. From its architecture and computational power, the oracle Turing machine is not a standard model according to computability theory but a hypercomputation model, term that is used to denote the possibility of computing non Turing machine computable functions (non Turing-computable functions) [4].

Contrary to computability theory, hypercomputation is not fully accepted within scientific and philosophical communities. Although numerous hypercomputation models have been devised from a logical point of view [13], current issues are more about the physical domain. Indeed, these issues directly concern the following claim that I will call "the hypercomputation thesis": it is physically possible to build a hypercomputation model.

To prove the hypercomputation thesis, one possible strategy could be to physically build an oracle Turing machine. More precisely, it is about finding a physical theory where a hypercomputation model will be able to use some external information from nature. Such an information could be regarded as an oracle that provide an additional element in order to go beyond Turing machines limits [1] [11].

[1] University of Paris 1, France, email: florent.franchette@gmail.com

However, there is a recurring epistemological problem about the physical construction of an oracle Turing machine [6, p. 13] [4, pp. 490–491]. This problem called "verification problem" may be worded as follows: if we assume we have an oracle Turing machine physically built, it would be impossible to verify that this model is able to compute a non Turing-computable function.

In this paper I will propose an analysis of the verification problem in order to show that it does not explicitly dispute the strategy about a physical construction of an oracle Turing machine.

2 HOW TO BUILD AN ORACLE TURING MACHINE?

The oracle Turing machine as defined by Alan Turing is not very detailed, it is why Jack Copeland and Diane Proudfoot have proposed an another definition of that hypercomputation model [5]. From their point of view an oracle Turing machine is a Turing machine which has two other elements: firstly a device (the oracle), which is able to make measures with an infinite precision, secondly a memory space that contains a real number called "τ". τ is an infinite binary string, which represents the results of a non Turing-computable function. If such a non Turing-computable function is denoted d, the nth symbol of τ represents $d(n)$, namely 0 or 1. And if we would like to have acces to $d(239208)$ the device measures the symbol number 239208 and provides the corresponding value. Therefore, an oracle Turing machine that have a non Turing-computable number inside its memory is able to compute more functions than the Turing machine.

In order to build an oracle Turing machine based on non computable information that comes from nature, it is necessary in the first place to locate this information within nature. The idea of finding this information from quantum randomness comes from both the standard model of quantum physics and Richard Feynman's works. In the one hand, the standard model postulates quantum randomness from the Born postulate[2]. On the other hand, Feynman concludes in his paper *Simulating Physics with Computers* that "it is impossible to represent the results of quantum mechanics with a classical universal device" [8] (p. 476). More recently, Cristian Calude has proposed to devise a hypercomputation model by using quantum randomness as an oracle to exceed the Turing machine's power [1].

Calude's strategy consists of fixing on a computer a device able to generate a string of random numbers from a quantum process. For example, the ID quantique company[3] has created a device whose name is "Quantis", which generates a string of random numbers from an

[2] The Born Postulate is the idea that a measurement of a particle will yield a result which follows probability distribution $|\psi|^2$, where ψ is the particle's wave function.

[3] **http://www.idquantique.com/**.

elementary quantum optics process [10]. More specifically, photons (light particles) are sent one by one onto a semi-transparent mirror and detected. The exclusive events (reflection - transmission) are associated to "0", "1" bit values and each of them have a probability at 50% to occur. The operation of Quantis is continuously monitored to ensure immediate detection of a failure and disabling of the random bit stream.

In theory, a computer equipped with Quantis might provide an arbitrarily long string of quantum random strings. However, this computer would be considered as a hypercomputation model only if the quantum random string cannot be generated by a Turing machine, that is to say only if the string includes an infinite number of bits. Actually a simple consideration shows that, with probability one, the sequence produced by the random process is not Turing-computable. There are indeed uncountably many infinite strings of digits, but there are only contably many Turing-computable strings. Therefore, assuming that each infinite string has the same probability of occurring as a result of a random process, the probability that a random process would generate a Turing-computable string of digits is zero, whereas the probability that the string of digits is not Turing-computable is one [2]. In that case Quantis would be seen as an oracle able to provide non computable information from nature.

Although the physical construction of an oracle Turing machine is sufficient *prima facie* to prove the hypercomputation thesis, an epistemological problem nevertheless remains. This problem is raised in the case where we have an oracle Turing machine and may be set out as follows: even if we build an oracle Turing machine we will not be able to prove the hypercomputation thesis because it would be impossible to verify that the machine is able to compute a non Turing-computable function. I am going to analyse this problem in order to suggest a way to overcome it.

3 HOW TO SOLVE THE VERIFICATION PROBLEM?

The verification problem has been set out by Jack Copeland in the form of a thought experiment:

> " There is an epistemological problem with the idea of hypercomputation. Suppose Laplace's genius says 'Here is a black box for solving the Turing-machine halting problem' (The problem arises no matter which non Turing-machine-computable function is considered.) Type in any integer x and the box will deliver the corresponding value of the halting function $H(x)$ or so Laplace's genius assures you. Since there is no systematic method for calculating the values of the halting function, you have no means of checking whether or not the machine is producing correct answers. Even simulating the Turing machine in question will not in general help you, because no matter how long you watch the simulation, you cannot infer that the machine will not halt from the fact that it has not yet halted" [4, p. 471].

The verification problem as set out by Copeland is particularly relevant in the case of oracle Turing machines because they are considered as black boxes whose internal behaviour is not specified. Nevertheless we can ask why the absence of a verification is a real problem about oracle Turing machines. According to Marc Gold, it is indeed impossible, exclusively from input-output Turing machine's behavior, to check the function that is computed by the Turing machine [9]. Intuitively, this is due to the fact that we only have at our disposal a finite number of results, which could every time correspond to other functions. Hence why the verification problem would be an obstacle to oracle Turing machines while it seems relevant about Turing machines?

Actually, here is the real problem about verification in hypercomputation: even if identification of the computing function is impossible both in effective computation and hypercomputation, we can verify in principle, that is to say regardless of computational resources, that a standard computer provides a correct result. Since a standard computer can be studied from its theoretical model, namely the Turing machine, we can have access to its results in principle checking step by step the computation from input to output. By contrast, we cannot proceed in the same manner with an oracle Turing machine physically built because we are not able to check each computational step of a hypercomputation model due to the absence of an effective procedure. Therefore if we have an oracle Turing machine physically built, we will not able to prove the hypercomputational power of the machine.

Some authors such as Cleland [3, p. 223], Shagrir and Pitowski [12, p. 99] brought several arguments to overcome the verification problem. However no complete account have been proposed. I am going to try to resume their arguments to rebuild such an account.

The central thesis of this account is to say that computation does not presuppose verification. This thesis is based on a distinction between two types of verification:

1. The verification in principle, which disregards computational resources.
2. The verification in practice, which takes into account computational resources.

The verification problem as set out by Copeland uses a verification in principle. We have showed that such a verification could challenge oracle Turing machin because this type of verification is possible about Turing machines. Therefore, to overcome this problem we have to consider the other type of verification, namely a verification in practice.

Actually, it turns out that we regard a function as computed by a standard computer even if we are not able to verify in practice the computed results. Take a particular function as an example (the argument works no matter which function is considered). Let p the function defined by $p(n) =$ the nth decimal of the expansion of π. It is easy to see that we cannot verify in practice (due to a lack of resources) whether the 10^{12}th decimal of π recently computed by a computer is 5. Yet, although it is impossible in practice to verify that a computer correctly computes p, we would tend to say nonetheless that the computer computes this function. But where does such a trust come from? This trust arises from the fact that it is possible to use some empirical methods (e.g. probabilistic causal relations, counterfactual suppositions grounded in physical law) to claim in a plausible way that a computer computes a function even if no perfect verification is possible. Therefore, computation does not presuppose verification because in practice we claim that a computer computes although we are not able to verify this claim.

From this point a view, the verification problem should not be considered as a thought experiment as Copeland did but as an empirical hypothesis. In other words, the problem could not be solved as long as one will suppose the construction of an oracle Turing machine; on the contrary we must to have an oracle Turing machine physically build to achieve empirical tests and therefore to claim that it computes a non Turing-computable function.

To clarify this last point, take as an example the oracle Turing machine proposed by Calude. Let us recall that it uses quantum randomness to provide a random numbers string that cannot be generated by a Turing machine. In theory, the main obstacle to claim whether this hypercomputation model is able to compute a non Turing computable function is based on the true-randomness of quantum physics that comes from the Born postulate. More precisely, there are two types of random processes: true-random processes and pseudo-random processes. Pseudo-random processes generate strings of numbers from pseudo-random methods (e.g. the linear congruence method), which numbers "appear" random but that are actually provided by deterministic formulae. Hence, if quantum processes are pseudo-random processes, a Turing machine would be able to simulate them since Turing machines that use pseudo-random algorithms are equivalent to standard Turing machines [7, pp. 183–212].

To solve the verification problem in an empirical way would be to increase the plausibility of the claim that the oracle Turing machine computes a non Turing-computable function. It would be to achieve tests on random strings in order to show with a high probability that they are not pseudo-random. If such tests could be achieve and that we conclude that a string is true-random, then we could claim that this string represents the results of a non Turing-computable function. However, the disadvantage is that all current pseudo-random number generators provide strings, which are in practice impossible to distinguish from true-random number strings. Nevertheless we cannot dismiss the possibility to have some day reasonable grounds to believe that a string is true-random. In the same manner, we cannot dismiss the possibility to have some day reasonable grounds to believe that an oracle Turing machine physically built is able to go beyond the Turing machine.

4 CONCLUSION

The verification problem from a point of view in principle could be seen as a threat to the physical construction of an oracle Turing machine. However, there is still a way to overcome this problem if we consider it from a practical point of view. We should achieve tests on random strings in order to show with a high probability that they are not pseudo-random. If such tests could be achieve and that we conclude that a string is true-random, then we could claim that this string represents the results of a non Turing-computable function. Yet this solution is not entirely satisfactory from a pragmatic point of view because even if we know that the oracle Turing machine computes a non Turing-computable function we would not know what is this function.

ACKNOWLEDGEMENTS

I would like to thank the referees for their comments which helped improve this paper. In particular, many thanks to Cristian Calude for his very interesting advices. Finally I am very grateful to the university paris 1 Doctoral school for supporting this work.

REFERENCES

[1] C.S. Calude, 'Algorithmic randomness, quantum physics, and incompleteness', *CDMTCS Research Report Series*, **248**, 1–17, (2004).
[2] C.S. Calude and K. Svozil, 'Quantum randomness and value indefiniteness', *Advanced Science Letters*, **1 (2)**, 107–127, (2008).
[3] C. Cleland, 'The concept of computability', *Theoretical Computer Science*, **317**, 209–225, (2004).
[4] J. Copeland, 'Hypercomputation', *Minds and Machines*, **12**, 461–502, (2002).
[5] J. Copeland and D. Proudfoot, 'Alan Turing's forgotten ideas in computer science', *Scientific American*, **208**, 76–81, (2009).
[6] M. Davis, *The myth of hypercomputation*, Alan Turing: life and legacy of a great thinker (2004), Berlin : Springer-Verlag, 2004.
[7] K. De Leeuw, E.F. Moore, C.E. Shannon, and N. Shapiro, *Computability by probabilistic machines*, Automata Studies : Princeton University Press, 1956.
[8] R. Feynman, 'Simulating physics with computers', *International Journal of Theoretical Physics*, **21**, 467–488, (1982).
[9] M. Gold, 'Limiting recursion', *The Journal of Symbolic Logic*, **30**, 28–48, (1965).
[10] T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, and A. Zeilinger, 'A fast and compact quantum random number generator', *Review of Scientific Instruments*, **71**, 1–23, (2000).
[11] B. Loff and J.F. Costa, 'Fives views of hypercomputation', *International Journal of Unconventional Computing*, **5**, 193–207, (2009).
[12] O. Shagrir and I. Pitowski, 'Physical hypercomputation and the church-turing thesis', *Minds and Machines*, **13**, 87–101, (2003).
[13] M. Stannett, *Hypercomputation models*, Alan Turing: life and legacy of a great thinker (2004), Berlin : Springer-Verlag, 2004.
[14] A.M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, The undecidable (1965), Mineola, New York : Dover, 1936.
[15] A.M. Turing, *Systems of logic based on the ordinals*, The undecidable (1965), Mineola, New York : Dover, 1936.

What Makes a Computation Unconventional? or, there is no such thing as Non-Turing Computation

S. Barry Cooper
University of Leeds, UK

Turing's standard model of computation, and its physical counterpart, has given rise to a powerful paradigm. There are assumptions underlying the paradigm which constrain our thinking about the realities of computing, not least when we doubt the paradigm's adequacy.

There are assumptions concerning the logical structure of computation, and the character of its reliance on the data it feeds on. There is a corresponding complacency spanning theoretical – but not experimental – thinking about the complexity of information, and its mathematics. We point to ways in which classical computability can clarify the nature of apparently unconventional computation. At the same time, we seek to expose the devices used in both theory and practice to try and extend the scope of the standard model. This involves a drawing together of different approaches, in a way that validates the intuitions of those who question the standard model, while providing them with a unifying vision of diverse routes "beyond the Turing barrier".

The consequences of such an analysis are radical in their consequences, and break the mould in a way that has not been possible previously. The aim is not to question, invalidate or supplant the richness of contemporary thinking about computation. A modern computer is not *just* a universal Turing machine. But the understanding the model brought us was basic to the building of today's digital age. It gave us *computability*, an empowering insight, and computing with *consciousness*. What is there fundamental that unconventional computation directs us to? What *is it* makes a computation unconventional? And having fixed on a plausible answer to this question, we ask: To what extent can the explanatory power of the mathematics clarify key issues relating to emergence, basic physics, and the supervenience of mentality on its material host?

This article has been published as a chapter in the book Computing Nature, Dodig-Crnkovic G. and Giovagnoli R., Eds. SAPERE, Springer, Heidelberg, 2013

Dualism of Selective and Structural Information in Modelling Dynamics of Information

Marcin J. Schroeder[1]

Abstract. Information can be defined in terms of categorical opposition of one and many, leading to two manifestations of information, selective and structural. These manifestations of information are dual in the sense that one always is associated with the other. The dualism can be used to model and explain dynamics of information processes. Such dynamical processes are analyzed in contexts of two domains, computation and foundations of living systems. In conclusion, it is proposed that the similar dynamics of information processes allows considering computational systems of increased hierarchical complexity resembling living systems.

1 INTRODUCTION

The concept of information has several very different definitions. In this large variety, only few qualify as correct and intelligible. Too frequently, definitions simply refer to intuitive understanding of the explanatory concepts. It is quite rare that the formulation of the definition refers to any particular philosophical background. However, there are two clearly distinguishable tendencies in the understanding of information. One is referring either explicitly or implicitly to selection or alternatively to distinction. The other has the general idea of the form or structure as the focal point of explanation.

The definition of information used in this paper was introduced and extensively analyzed in earlier articles of the author. Its desirable feature is that both ideas of selection and of structure can be found as alternative ways of its interpretation.

Moreover, it turns out that the selective and structural manifestations of information are dual in the sense that one always is associated with the other. The dualism can be used to model and explain dynamics of information processes. Dynamical processes of this type are analyzed in contexts of two domains, computation and foundations of living systems. In conclusion, it is suggested that the similar dynamics of information processes allows considering computational systems of increased complexity resembling living systems.

Due to the scope and limitation of format of this paper more detailed presentation of the technical issues in description of information dynamics will be published elsewhere.

2 DUALISM OF SELECTIVE AND STRUCTURAL INFORMATION

The concept of information is understood here in the way it was defined in earlier papers of the author [1] as identification of a variety. Thus, starting point in the conceptualization of information is in the categorical opposition of one and many.

[1]Akita International University, Japan, email: mjs@aiu.ac.jp

The variety in this definition, corresponding to the "many" side of the opposition is a carrier of information. Its identification is understood as anything which makes it one, i.e. which moves it into the other side of the opposition. The preferred word "identification" (not the simpler, but possibly misleading word "unity") indicates that information gives an identity to a variety, which does not necessarily mean unification, uniformization or homogeneization. However, this identity is considered an expression of unity, "oneness".

There are two basic forms of identification. One consists in selection of one out of many in the variety, the other of a structure binding many into one. This brings two manifestations of information, the selective and the structural. The two possibilities are not dividing information into two types, as the occurrence of one is always accompanied by the other, but not on the same variety, i.e. not on the same information carrier. For instance, information used in opening a lock with corresponding key can be viewed in two alternative ways. We can think about a proper selection of the key, out of some variety of keys, or we can think about the spatial structure of the key which fits the structure of the lock. In the first case, the variety consists of the keys, in the second the variety consists of material units forming appropriate shape of the key. Thus, we can consider selective and structural information as dual manifestations of one concept.

The identification of a variety may differ in the degree. For the selective manifestation this degree can be quantitatively described using an appropriate probability distribution and measured using for instance entropy, or more appropriate measure [2]. For the structural manifestation the degree can be characterized in terms of decomposability of the structure [3].

Selective-structural duality of information is reflected in a variety of contexts. An example of very general character can be found in the way how we form concepts. One way is focusing on the denotation and selection of objects which we want to include in denotation. Another way is to focus on the connotation and configuration of characteristics which describe it.

Another example can be found in the analysis of scientific inquiry. In his philosophical analysis of the methods of science and history Wilhelm Windelband [4] introduced frequently revoked distinction, or even opposition of nomothetic and idiographic methodologies. The former has its starting point in the acknowledgement of the differences, but assumes the existence of similarities which produce grouping within the variety, and therefore it is looking for comparable aspects. The latter is assuming the uniqueness of the object of study and therefore is focused on elements which constitute this uniqueness. Although, the distinction is between methodologies of inquiry, not between manifestations of information, association is quite clear.

Similar, but much more frequently used distinction in the context of cultural studies has been introduced more than a half

century later by Keneth L. Pike [5]. He called his methodological schemata etic and emic methodologies, deriving their names from phonetic and phonemic studies of language. Here too, the distinction is based on the differences in the perspective of the study. In the first case the subject of study is viewed in a comparative manner as a member of a variety in which differences and similarities are used to establish its unique characteristics. In the second case, the subject of the study, whose uniqueness is already assumed, is viewed from the inside with the aim to reconstruct its internal structure.

In these examples, as well as in all instances of the reflection of the selective-structural duality in methodological analysis, it is considered obvious that the choice of a particular method is dictated by the discipline of inquiry. Physics for instance is recognized always as a paradigm of the approach corresponding to selective information. After all, probability distributions describe the state of a system, collective one in classical physics, and individual in quantum physics. But closer look reveals that actually in this domain both methodological positions are omnipresent. It is enough to recall tendency of geometrization in physics continuing beyond the General Relativity Theory, or the special role of the field theory to recognize the presence of the view associated with structural information.

The most significant is association of the selective-structural dualism of information with the dualism of function and structure in the foundation studies of living systems, which constitutes the central theme of the work of Humberto Maturana and Francisco Varela [6] on autopoiesis. Here it becomes clear that this dualism is not just a matter of the choice of a method of inquiry, but it is a characteristic of living systems. Function determines structure and structure determines function. Maturana and Varela were looking for the resolution of this convolution in autopoiesis, self-construction of living systems. However, from the point of view of information studies, there is no need to restrict this dualism to living systems, as it is simply reflection of the universal dualism of selective and structural information. Functions of the elements of a system give them identity by distinguishing them from, and giving them their place in the differentiated variety. On the other hand, this distinction is a consequence of the specific structural characteristics that they posses, their internal structure allows them to play specific roles in the system. It is not a matter of the right or wrong perspective of the study, but an inherent feature of all information systems.

Mathematics provides several different examples of dualism which can be very clearly associated with that of selective and structural information. The most fundamental can be traced back to the 19th Century when Felix Klein formulated in his 1872 Erlangen Program the view of geometry as a theory of invariants for the group of transformations of a geometric space. Instead of identification of the objects of geometric studies through analysis of their internal structure, the structure of transformations of the plane or space is selected, and only then geometric objects appear as those subsets of points which are transformed into themselves, although their points may be exchanged. Such an approach, in which instead of inquiry of internal structure of objects, the structure of transformations preserving the identity of these objects is analyzed, has become commonly used in a wide range of mathematical theories leading to the development of the theory of categories and functors.

In the past, the dualism of selective and structural information has been present in information studies only in the form of a competition between two, apparently conflicting views on the "proper" answer to the question "What is information?" [1]. The dominating position focusing on the selective manifestation of information and neglecting the structural one was supported by the practical success of Shannon's quantitative characterization of information in terms of entropy. But the failure in establishing equally successful semantics for information understood exclusively in terms of selection was driving the efforts to shift studies of information to its structural manifestation.

The dual approach achieved through the definition of information used in the present paper has more advantages than just reconciliation between adherents of competing views on information. It also helps to model dynamics of information in processes of evolution or computation.

3 DYNAMICS OF INFORMATION IN COMPUTING

The definition of information in terms of the one-many opposition has been a starting point for author's attempt to formulate a theoretical framework for information [7]. This framework has a static form reminding a logical structure, at least in the sense of a similar mathematical formalism. However, the formalism can be used to model process of information integration [3].

The change of the level of information integration is not a dynamical process, understood as transformation resulting from the interaction of different information systems. For this reason, information integration, although modelled by a theoretical device called a Venn gate in the earlier papers of the author should not be confused with computation.

What is computation in the present conceptual framework? First, we have to clarify some quite common confusion related to the distinction between analog and digital computing. The distinction of "analogy and digital" principles, automata, or machines introduced by John von Neumann [8] at the time when first computers were being constructed was referring to the way the numbers are represented, by certain physical quantity, or by "aggregates of digits."

For von Neumann the main issue here was in handling errors. He wrote "Thus the real importance of the digital procedure lies in its ability to reduce the computational noise level to an extent which is completely unobtainable by any other (analogy) procedure."

Of course, von Neumann was right about practical advantages of "digital procedure" in handling errors, but he overlooked what actually constitutes the distinction. The mistake he made is being perpetuated even now. Of course, the numbers are always represented by physical quantities, even in digital computers. "Aggregates of digits" do not exist independently from the physical systems constituting machines. To that extent everyone will agree with Ralph Landauer [9] that information is physical.

Thus, the actual distinction is in semantics of information. It is the way how we associate numbers with physical states of the computing machine which decides whether computing is digital or analog. Information itself is neither one, nor the other. To avoid going too far beyond the scope of this paper, simplifying assumption will be made that information is associated with the state of the physical system which is used as a computing machine. Then, observables will assign numbers to particular states, giving meaning to information [10].

Now, we can begin analysis of the process of computing modelled by Turing machines. Once again we have to be careful with traditional way of imagining of the process. Situation is similar to the way people were interpreting mechanical processes before Isaac Newton introduced his Third Principle of Mechanics. Every change in the world had to have an active agent (subject) and passive object of the action. Newton recognized that in mechanical phenomena there is no action, but only interaction. The Third Principle states that we cannot distinguish between an agent and recipient of action, as we have always mutual interaction. I cannot claim that my pushing the wall is in any way different from wall's pushing me.

From this point of view the interpretation of a head in Turing machine printing a character on the tape is an arbitrary assumption. We can simply talk about mutual interaction in which characters change (or not) on the tape in contact with the head, and the head is changing its state/instruction in contact with the tape.

More precisely, we could describe Turing machine as a device consisting of two information systems, which in order to retain traditional terminology are called a tape and a head, each consisting of independent components being themselves information (sub)systems. For the tape, components are cells. For the head, subsystems are positions of instructions on the list. At every moment both systems have finite, but unlimited number of components, and the number of components can grow without restriction.

Each component (cell or position on the list of instructions) is capable to assume one of the finite number of states (usually different for components of the tape and components of the head). For cells on the tape the states are characters from traditional description of Turing machine. For components of the head (positions on the list of instructions), there is a finite number of choices of instructions which give the position particular state.

Now, we have a crucial and restricting assumption, that these two fundamental information systems can interact only by contact of a single pair of active components (which corresponds to the traditional assumption that the head is in the state with one particular instruction, and it can read and act on a single cell).

Experience from the studies of Turing machines suggests that the assumption is not restrictive as long as the difference between one pair of active components is contrasted with clearly defined finite number of pairs. The restrictive character appears when we exclude the possibility of interaction on the scale of all systems.

The process of computing is described as follows. The active cell is changing (or not) its state (character) into one determined by the state of the active component of the head (particular instruction in the position on the list for given state). On the side of the head the change of the instruction depends on the state of the cell (character in the cell). Then both fundamental information systems change their active component. Again this change on the tape depends on the state of the active component of the head, the change in the head depends on the state of the active cell (character).

This machine is little bit more general than Turing's A-machine, as the process allows changes of instructions in the head. This machine could be called a symmetric machine (an S-machine) because the process consists in mutual interaction producing similar type of change. It is being reduced to usual Turing A-machine, if we additionally assume that the instructions in the head are not changing.

The symmetric Turing machine describes a general dynamic process of the interaction of a pair of complex systems with a restricting assumption that the interaction is in each moment through exactly one pair of components (not very strong restriction), and additional one, that the choice of the next pair is determined, not random (rather strong restriction). Of course, the possibility of removing these restrictions could be considered, but it seems to destroy the algorithmic character of the process.

Even with these two restrictions, symmetric Turing machine gives us a model of information dynamics applicable to a very wide range of information systems.

We know that computation cannot be reduced to one information system. Claude Shannon [11] showed that the head of Turing machine has to have at least two different states. Once we have a variety of two states and choice between them, we have information.

Now, the dynamics of the process of computation is revealed in the selective-structural dualism of information. For both fundamental information systems (tape and head) information is structural. The state of all tape consists of configuration of characters in its cells, but computation is an interaction in which the choice of one out of many states (characters) of the active component (cell) is being made. Similarly, the state of the head is in the configuration of instructions, but in each step of computation one out of many possible choices of instruction is being made.

Finally, we could consider an extension of the process of computation using the concept of selective-structural information dualism. While computation considered at the level of active, interacting pair of components refers to the selective manifestation of information (e.g. selection of a character for the cell), each character can be understood as structural manifestation of information. Corresponding to this structural manifestation, its selective counterpart can be subject to interaction which results in its own dynamics. This way we can consider multi-level symmetric Turing machines, which resemble systems encountered in the study of the foundations of life.

4 DYNAMICS OF EVOLUTION

Before we enter the analysis of evolutionary mechanisms, it is necessary to consider more general issue of control systems. In this domain the most fundamental principle has been formulated by W. Ross Ashby as the Law of Requisite Variety "A model system or controller can only model or control something to the extent that it has sufficient internal variety to represent it" [12,13]. This principle in the informal, intuitive form and in application to the process of generation, not to the modelling or controlling has been until the end of the 18th Century used as an argument for the hierarchy of beings and the need for supremely intelligent creator acting intentionally to generate them [14].

It seemed obvious that any complex system can be generated only by a system of higher level of organization. This reasoning is based on the assumptions that creation is an action (not interaction) and requires a design. Following the Law of Requisite Variety such a design, i.e. internal model is impossible without higher degree of variety. Evolutionary model of the development of life disposed of the design putting this higher

level of variety in the environment. Thus the species are getting increasingly complex by the interaction with the environment, which of course is a carrier of a huge amount of information.

Let's start from a dualistic model of relatively simple mechanism of feedback control. It requires interaction of two information systems. Selection of a state of one of them through interaction is accompanied by the selection of a state of the other, which in turn has its reflection in the structural manifestation of information. This structural manifestation of information in the second system is determining the structural information of the first system. And this corresponds to the modification of the selection of its state.

For instance, using classical example of a governor controlling work of the steam machine, we have two information systems which can be in a variety of states. One is a valve whose state (described by the amount of steam coming by it) decides about the speed of the work of the machine. The other is a pair of balls hanging on arms rotating around the vertical axis whose rotation is propelled by the machine. Its state (velocity of rotation) is selected by the work done by machine. From the structural point of view, information is manifested by the geometric structures of the systems, diameter of the valve and extension of the arms on which the balls are attached. The higher is extension of arms, the smaller diameter of the valve.

The governor is a simple case of an artefact invented by humans, originally with the intention to control the work of windmills. It is more complicated situation when we want to explain the dynamics of information in systems which were naturally generated without any intentional design.

Now, we can proceed to the dualistic description of the evolutionary process. Here, in distinction from the earlier example where the function was a result of human invention and the structure followed the needs of implementation, we can encounter confusion which puzzled generations of biologists, but which can be easily resolved within the dualistic perspective.

The mechanism of evolution is usually reduced to natural selection in which the fittest organisms survive and reproduce transmitting and perpetuating their genetic information. The puzzling question is about the meaning of the term "fittest". Does it have any other meaning beyond the tautological statement that these are organisms which survived and reproduced?

The answer is that the meaning of the term "fittest" is expressing the relationship between two manifestations of information. While naturally, natural selection describes the dynamics of information for selective manifestation in terms of reproduction (which obviously requires survival), the fittest individuals are those whose phenotype has structural characteristics compatible with structural characteristics of the environment.

More generally, we can describe the evolutionary process as such in which two (or more) information systems interact. Interaction is determining the outcome of the selection, and therefore the dynamical view seems more natural in terms of selective manifestation. However, it is the structural manifestation of information which actually demonstrates the results of evolution. And what is most important, there is no point in asking which manifestation is more important, primary, or true. Dynamics of information has two manifestations, simply because information does.

5 DYNAMICS OF INFORMATION IN LIVING SYSTEMS

Thus far we were talking about biological evolution of species as a dynamical information process. We were concerned with the question how this process can be understood. There is another, much more difficult question why it occurs, and why in this particular way. To seek the answer, we have to consider more general issue of the dynamics of information in the living systems. Naturally, it is equivalent to the inquiry regarding the question "What is life?" We will consider here only some aspects of this extremely broad and deep problem. Specifically those related to the selective-structural dualism of information. The issues related to the necessity of holistic methodology in the study of life are presented in another article of the author [15].

The main fallacy in answers to the question "What is life?" is in the attempt to explain life by distinguishing one particular process driving all other in the multi-level hierarchical structure of the biosphere. This fallacy is being perpetuated even in most recent publications [16]. The process chosen by the authors of explanations could be photosynthesis (but, what about forms of life which do not depend on it?), metabolism, reproduction with transmission of genetic information, formation of large organic molecules, etc.

Another problem is in the restriction of attention to what is called a biosphere. In the earliest fundamental answer to the question Erwin Schrödinger [17] pointed at what he called negative entropy of the light coming from sun as the factor driving processes of life. It is also a fallacy perpetuated continuously by generations of authors who change the name of the factor (negentropy, entropy deficit, inhomogeneity, etc.) but do not notice that the light coming to earth does not have high or low entropy. It is a matter of the process in which incoming visible light, for which the atmosphere is transparent, is transformed by living systems and reradiated into cosmic space as infrared radiation of 22 times higher entropy [18].

Thus, the driving factor is a mechanism which transcends biosphere and which has its source in astronomical phenomena of huge spatial and temporal measures. But this driving factor itself would not produce life processes, but is just a necessary condition for life. It creates conditions allowing generation of information participating in the dynamic processes of life at all of its levels. Life cannot be understood by observing only one of these levels, as it is usually done. To understand working of the clock we cannot focus on the spring or battery which powers it, or on one of the wheels.

Of course, evolution of species, cycles of metabolism, photosynthesis, or reproduction are component processes of life. But neither has privileged or exclusive position. We can ask however about the common features for component processes of life. Here we can find again help in the dualistic perspective on information, which definitely is the common concept for all life processes.

The main characteristic of life processes consists in enriching information in one system of smaller variety, i.e. lower informational volume through the dynamic interaction with the other. This process was already described in a general view of the dynamics of information in the evolution of species. We need in this case generation of a large variety of objects and interaction with the other system which selects some of them (the fittest) whose structural characteristics predestine them to

survive. Thus, the collective system is increasing its organization (internal information) not because they have some design, but they fit selective information of the outer system. The crucial point is in inseparable dualism between the two manifestations of information and multi-level character of the total system.

6 CONCLUSIONS & FUTURE WORK

There are two domains where dualism of selective and structural information can be used to model dynamics of information, computation and living systems. Although in both cases dynamics is similar, there is a big contrast between the levels of complexity between them. In what here was described as a slightly more general view of Turing machines there are two information systems (tape and head) which are considered at the two levels corresponding to selective and structural information.

Life constitutes an extremely complex system of at least dozens of levels and the number of component information systems exceeding any practical limits of calculation. However, the basic mechanism involving in its description the two manifestations of information is the same as in symmetric Turing machines.

On the other hand, there is nothing which prevents us from designing computational systems of complexity going beyond two levels. This may require more complicated (multilevel) semantics of information (which in traditional Turing machines is typically an association of particular combination of the states of cells with natural numbers). Each cell may be considered a carrier of an information system with its own variety and with dynamical mechanisms of evolution adjusted to the conditioning by higher or lower levels of the hierarchical structure.

The study of such theoretical systems and their practical implementation is of some interest and has a potential wide range of applications.

REFERENCES

[1] M. J. Schroeder. Philosophical Foundations for the Concept of Information: Selective and Structural Information. In: *Proceedings of the Third International Conference on the Foundations of Information Science, Paris 2005*, http://www.mdpi.org/fis2005 (2005).

[2] M. J. Schroeder.. An Alternative to Entropy in the Measurement of Information. *Entropy*, 6: 388-412 (2004).

[3] M. J. Schroeder. Quantum Coherence without Quantum Mechanics in Modeling the Unity of Consciousness. In P. Bruza, et al. (Eds.) *QI 2009*, LNAI 5494, Springer, pp. 97-112 (2009).

[4] W. Windelband, W. (1924). *Praeludien: Geschichte und Naturwissenschaft*, Vol. II. Tubingen, pp. 136-160 (1924).

[5] K. L. Pike. Emic and Etic Standpoints for the Description of Behavior. In K. L. Pike, *Language in Relation to Unified Theory of the Structure of Human Behavior.* The Hague: Mouton (1966).

[6] H. R. Maturana and F. J. Varela. *Autopoiesis and Cognition: The Realization of the Living.* Boston Studies in the Philosophy of Science, vol. 42. Dordrecht: Reidel (1980).

[7] M. J. Schroeder. From Philosophy to Theory of Information. *International Journal Information Theories and Applications,* 18 (1): 56-68 (2011).

[8] J. von Neumann. The General and Logical Theory of Automata. In A. H. Taub (Ed.) *John von Neumann, Collected Works, Vol. V.* Pergamon Press (1948).

[9] R. Landauer. Information is Physical. *Physics Today,* May: 23-29 (1991). .

[10] J. M. Jauch, *Foundations of Quantum Mechanics*. Reading, Mass.: Addison-Wesley (1968).

[11] C. E. Shannon. A Universal Turing Machine With Two Internal States. In: C. E. Shannon and J. McCarthy (Eds.) *Automata Studies,* pp.157-165 (1956).

[12] W. R. Ashby. *An Introduction to Cybernetics.* London: Chapman & Hall (1956).

[13] F Heylighen. Principles of Systems and Cybernetics: an evolutionary perspective. In: R. Trappl (Ed.) *Cybernetics and Systems '92*. World Science: Singapore, 1992; pp. 3-10 (1992).

[14] M. J. Schroeder. Concept of Information as a Bridge between Mind and Brain. *Information,* 2 (3): 478-509 (2011) Available at http://www.mdpi.com/2078-2489/2/3/478/

[15] M. J. Schroeder. The Role of Information Integration in Demystification of Holistic Methodology. In P. L. Simeonov, L. S. Smith, A. C. Ehresmann (Eds.) *Integral Biomathics: tracing the Road to Reality. Proceedings of iBioMath'2011-Am, San Jose, CA, USA, iBioMath 2011-Eu, Paris, France and ACIB'11, Stirling, UK.* Berlin: Springer-Verlag, pp. 79-96 (2012).

[16] N. Sato. Scientific Élan Vital: Entropy Deficit or Inhomogeneity as a Unified Concept of Driving Forces of Life in Hierarchical Biosphere Driven by Photosynthesis. *Entropy,* 14: 233-251 (2012).

[17] E. Schrödinger. *What is Life?* Cambridge: Cambridge University Press (1945).

[18] M. J. Schroeder. Entropy Balance in Climate Changes. In: *Proceedings of the 44-th Coal Science Symposium, Akita, October 2007* (2007). Available at: http://ci.nii.ac.jp/naid/110006611099/

Turing Test, Chinese Room Argument, Symbol Grounding Problem. Meanings in Artificial Agents

Christophe Menant[1]

Abstract. The Turing Test (TT), the Chinese Room Argument (CRA), and the Symbol Grounding Problem (SGP) are about the question "can machines think?". We present TT, CRA and SGP as being also about generation of meaningful information like humans do. This allows addressing the question of thinking machines by analysing the possibility for Artificial Agents (AAs) to generate human-like meanings. We look at such a possibility by using the Meaning Generator System (MGS) where a system submitted to a constraint generates a meaning in order to satisfy its constraint. The system approach of the MGS allows comparing meaning generations in animals, humans and AAs. The comparison shows that in order to design AAs capable of generating human-like meanings, we need the AAs to carry human constraints. Transferring human constraints to AAs raises concerns coming from the unknown natures of life and human consciousness which are at the root of human constraints. Implications for the TT, the CRA and the SGP are highlighted. It is shown that designing AAs capable of thinking like humans needs an understanding about the natures of life and human mind that we do not have today. Following an evolutionary approach, we propose as a first entry point an investigation about integrating a living entity in an AA in order to extend her "stay alive" constraint to the AA.
Ethical concerns are raised from the relations between human constraints and human values.
Continuations are proposed.

1 PRESENTATION

The question "Can machines think?" has been addressed in 1950 by Alan Turing with a proposed test, the Turing Test (TT), where a computer is to answer questions asked by humans. If the answers from the computer are not distinguishable from the ones coming from humans, the computer passes the TT [1]. The validity of the TT has been challenged in 1980 by John Searle with a thought experience, the Chinese Room Argument (CRA), aimed at showing that a computer cannot understand human language [2]. The possibility for computers to attribute meanings to words or symbols has been formalized by Steven Harnad in 1990 through the Symbol Grounding Problem (SGP) [3].
With the question "can machines think?" understood as "can machines think like human beings think?" [4], we propose to look at these approaches to Artificial Intelligence (AI) by showing that they all address the possibility for Artificial Agents (AAs) to generate meaningful information (meanings) as humans do. The Initial question about thinking machines is then reformulated as "can AAs generate meanings like humans do?"
In order to compare meaning generation in humans and in AAs we use an existing system approach to meaning generation: the Meaning Generator System (MGS) where a system submitted to a constraint generates a meaning when it receives information that has a connection with the constraint [5]. We first look at TT and CRA where relations with meaning generation can be considered as rather explicit. The case of SGP is addressed separately as its relations with meaning generation deserve more details. These analysis show that AAs cannot today generate human-like meanings because human constraints cannot be transferred to AAs. This because we do not understand the natures of life and human mind which are the base ground of human constraints. Consequently, today AAs cannot think as humans do. A better understanding about the natures of life and human mind is needed for designing really intelligent AAs capable of thinking like humans do. We propose an entry point to be investigated: the integration of living entities into AAs. This in order to allow the extension of the "stay alive" constraint into AAs.
Ethical concerns are also raised as the coverage of human values by human constraints in terms of meaning generation is to be explored.
Continuations are proposed in order to develop with more details several points used here.

2 TURING TEST, CHINESE ROOM ARGUMENT AND MEANING GENERATOR SYSTEM

The TT is about the capability for a computer to understand questions formulated in human language and to answer these questions as well as humans would do. Regarding human language, we can consider that understanding a question is to grasp the meaning of the asked question. And answering a question also goes with generating the meaning of the answer. So we can consider that the TT is about meaning generation.
The CRA challenges the TT by showing that a computer can pass the TT without understanding symbols. A person not speaking Chinese and exchanging Chinese symbols with people speaking Chinese can make them believe she speaks Chinese if she chooses the symbols following precise rules written by Chinese speaking persons. The person not speaking Chinese passes the TT. A computer following the same precise rules would also pass the TT. In both cases the meaning of the Chinese symbols is not understood. The CRA wants to show that the TT is not valid for testing machine thinking capability as it can be passed without associating any meaning to the exchanged information. The TT does not ensure understanding. Here also, the understanding of the symbols goes with generating the

[1] IBM France (Non Active Employee). christophe.menant@hotmail.fr.

meanings related to the symbols. So we can consider that the TT and the CRA are about the possibility for AAs to generate human-like meanings. This brings the question about machines capable to think to a question on meaning generation. Can AAs generate meanings as we humans do?

In order to compare the meanings generated by humans and by AAs, we use the Meaning Generator System (MGS) [5]. The MGS models a system submitted to a constraint that generates a meaning when it receives information that has a connection with the constraint. The generated meaning is precisely the connection existing between the received information and the constraint, and it is used to determine an action that will be implemented in order to satisfy the constraint. The MGS is simple. It can model meaning generation in elementary life. A paramecium moving away from acid water can be modelled as a system submitted to a "stay alive" constraint that senses acid water and generates a meaning "presence of acid not compatible with the "stay alive" constraint". That meaning is used to trigger an action from the paramecium: get away from acid water. It is clear that the paramecium does not possess an information processing system that would allow her to have access to an inner language. But a paramecium has usage of sensors that can participate to a measurement of the acidity of the environment. The information made available with the help of these sensors will be part of the process that will generate the move of the paramecium in the direction of less acid water. So we can say that the paramecium has overall created a meaning related to the hostility of her environment in connection with the satisfaction of her vital constraint. Fig 1 illustrates the MGS with this example.

The MGS is a simple tool modelling a system submitted to a constraint. It can be used as a building block for higher level systems (agents) like animals, humans or AAs, assuming we identify clearly enough the constraints corresponding to each case [2].

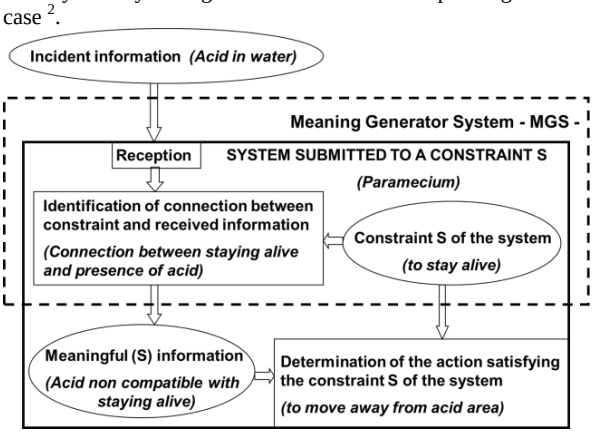


Figure 1. The Meaning Generator System

The MGS is also usable to position meaning generation in an evolutionary approach. The starting level is basic life with a "stay alive" constraint (for individuals and for species). The sight of a cat generates a meaning within a mouse, as well as a passing by fly within a hungry frog. We however have to keep in mind that staying alive refers to life, the nature of which is unknown as of today. We can easily identify and understand the actions implemented to satisfy a "stay alive" constraint without accessing the nature of the constraint. For humans, the constraints are more difficult to identify as they are linked to human consciousness and free will which are both mysterious entities for today science and philosophy. Some human constraints are however easy to guess like "look for happiness" or "limit anxiety" [3]. Reference to the Maslow pyramid can also be used as an approach to human constraints [6]. In all cases the action implemented to satisfy the constraint will modify the environment, and so the generated meaning. Meanings do not exist by themselves. They are agent related and come from generation processes that link the agents to their environments in a dynamic mode.

Most of the time agents contain several MGSs related to different sensorimotor systems and different constraints to be satisfied. An item of the environment generates different interdependent meanings that build up networks of meanings representing the item to the agent. These meaningful representations embed the agent in its environment through constraints satisfaction processes [6].

To see if AAs can generate meanings like humans do, we have to look at how human meaning generation processes could be transferred to AAs. Fig 1 indicates that the constraint is the key element to consider in the MGS. The other elements deal with data processing that is transferrable. When looking at transferring constraints to AAs, we have to consider that the natures of human and animal constraints are unknown as of today. Take for instance the basic "stay alive" constraint that we share with animals. We know the actions that are to be implemented in order to satisfy that constraint, like keep healthy and avoid dangers. But we do not really know what life is. We understand that life came out of matter during evolution, but we do not know how life could be today built up from inanimate matter. We have many definitions for life, but the nature of life is today a mystery. Consequently, we cannot transfer a "stay alive" constraint to AAs. The same applies for human specific constraints which are closely linked to human consciousness. We do not know exactly what is "look for happiness" or "limit anxiety". We know (more or less) the physical or mental actions that should be implemented in order to satisfy these complex constraints, but we do not know the nature of the constraints. And this is because we do not know the nature of human mind which is, as is the nature of life, a mystery for today science and philosophy. So we have to face the fact that the transfer of human constraints to AAs is not today possible as we cannot transfer things we do not understand.

We cannot today build AAs able to generate meanings as we humans do because we cannot transfer human constraints to AAs. The computer in the TT cannot be today in a position to generate meanings like humans do. The computer cannot understand the questions nor the answers as humans do. It cannot pass the TT. Consequently, the CRA is right. Today AAs cannot think like humans think. Strong AI is not possible today. A better understanding about the nature of life and human mind is

---

[2] The MGS approach is based on meaning generation for constraint satisfaction. It is different from "action oriented meaning". In the MGS approach, the constraint to be satisfied is the cause of the generated meaning. The action is a consequence of the meaning and comes after it. More on this at [6].

[3] "Anxiety limitation" has been proposed as a constraint feeding an evolutionary engine in a human evolutionary scenario [12, 14].

necessary for a progress toward the design of AAs capable of thinking like humans think. Important research activities are in process in these areas [8, 9]. Some possible short cuts may however be investigated, at least for the transfer of animal constraints (see hereunder).

3 SYMBOL GROUNDING PROBLEM AND MEANING GENERATOR SYSTEM

The Symbol Grounding Problem is generally understood as being about how an AA computing with meaningless symbols could generate meanings intrinsic to the AA. "How can the semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads? How can the meanings of the meaningless symbol tokens, manipulated solely on the basis of their (arbitrary) shapes, be grounded in anything but other meaningless symbols?" [3].

This is again about asking how AAs can generate meanings as humans do. The conclusions reached in the previous paragraph apply: AAs cannot today generate meanings as we humans do because we are not in a position to transfer human constraints to AAs. The SGP cannot today have a solution [4].

Some researchers tend to disagree on the fact that a solution to the SGP can be sufficient for providing meaningful mental states. They consider that meaningful thoughts have to be at the same time conscious mental states [7].

This position can be addressed with the MGS approach where human constraints have to be transferred to AAs so that the AAs can generate meanings like humans do. Such meanings go with human constraints, closely linked to human consciousness.

The requested intrinsic aspect of the semantic interpretation also brings argument in favour of no solution to the SGP when using the MGS approach. Putting aside metaphysical perspectives, we can say that the generation of meaningful information appeared on earth with the first living entities. There is no meaning generation in a world of inanimate matter. Life is submitted to an intrinsic and local "stay alive" constraint that exists only where life is. Today AAs are made with inanimate matter. The constraints that AAs can carry come from the designer of the AA. The constraints are derived from the designer and cannot be intrinsic to the AA where there is no life [6]. Consequently, it is only for living entities that the meaning of the symbols can be "intrinsic to the system". Symbol grounding in a material world does not bring intrinsic meaning generation. This comment on the notion of intrinsicness confirms the position expressed above: in the today world of material AAs, the SGP cannot have a solution. We need a better understanding about the nature of life in order to address the possibility for intrinsic meaning generations in AAs. As said, many researches are currently on going on these subjects and there are perspectives for progresses [9].

Another area of investigation for intrinsic constraints in AAs is to look for AAs capable of creating their own constraints. Whatever the possible paths in this area, it should be clearly highlighted that such approach would not be enough to allow the design of AAs able to think like humans do. The constraints that the AAs might be able to generate by themselves may be different from human ones or managed differently by the AAs. These future AAs may think, but not like humans think. This also brings up ethical concerns for AI where AAs would not be managing constraints and meanings the same way humans do.

[4] Several proposals have been made as solutions to the SGP. Most have been recognized as not providing valid solutions [13].

4 ARTIFICIAL INTELLIGENCE, ARTIFICIAL LIFE AND MEANING GENERATION

The above usage of the MGS with the TT, the CRA and the SGP has shown that machines cannot today think like humans do because human constraints are not transferrable to AAs.

It is worth recalling that the basic "stay alive" constraint is part of these constraints. And not being able to transfer a "stay alive" constraint to AAs implies that we cannot design AAs managing meanings like living entities do. We can only imitate some performances of life. So not only can't we design AAs able to think like humans think, we can't even design AAs able to live like animals live. As shown, the blocking points are relative to our lack of understanding about the natures of life and human consciousness.

In terms of increasing complexity, these subjects can be positioned following an evolutionary approach. It is agreed that life came up on earth before human mind. So it looks logic to address first the problem of the "stay alive" constraint not transferrable to AAs. Even if we do not know the nature of life, we are able to manipulate it. And we could, instead of trying to transfer the performances of life to AAs, look at how it could be possible to extend life with its performances to AAs. Somehow bring life in AAs. Sort of "meat in the computer". The AA would then be submitted to the "stay alive" constraints brought in by the living entity, while keeping some control on that living entity. Such approach is different from trying to get organic elements obeying computer logic [10] or trying to use within AAs the sensori-motor performances of living entities, like insect-machine hybrids [11].

The idea of Cyborgs is not new. What we propose to look at is about the possibility to have a living entity resident in an AA and bringing in it a "stay alive" constraint to which the AA would be submitted. This would allow the AA to generate meanings like animals do and interface with its environment like animals do. Such possible progresses in having AAs submitted to resident animal constraints does not bring much about AAs submitted to human constraints. We can however take this as a first step in an evolutionary approach to AAs containing human constraints.

5 MEANING GENERATION, CONSTRAINTS, VALUES AND ETHICAL CONCERNS

The MGS approach has shown that our current lack of understanding about the nature of life and human consciousness makes impossible today the design of AAs able to think or feel like humans do. This because we do not know how to transfer to AAs the human constraints that we do not understand. These human constraints are closely related to free will and consciousness which are mysteries for today science or philosophy. But human constraints do not a priori include human values. Some find happiness with the suffering of others. The way by which human constraints can take into account human

values is not obvious, nor clear. This brings to highlight here ethical concerns that address two directions at least.
First, researches about the nature of human consciousness should consider how human values could be linked to human constraints. This is a challenging subject as human values are not universal. But the nature of human consciousness is still to be discovered and we can hope that its understanding will shed some light on the diversity of human values.
As addressed above, another case is the one about AAs becoming capable of generating by themselves their own constraints. Such approach should be careful keeping human values in the background of these constraints so such AAs are not brought in a position to generate meanings and actions too distant from human values.

6 CONCLUSIONS

Meaning generation for constraint satisfaction shows that we cannot today design AAs capable of generating meanings like humans do. Our lack of understanding about the natures of life and human mind makes impossible the transfer of human constraints to AAs. The consequence is that human-like meaning generation is not today possible within AAs. Considering the TT, the CRA and the SGP as being about meaning generation, we can say that today AAs cannot think like we humans do, they cannot pass the TT. The CRA is correct, and the SGP cannot have a solution. Strong AI is not possible. Only weak AI is possible. Imitation performances can be almost perfect and make us believe that AAs generates human-like meanings, but there is no such meaning generation as AAs do not carry human constraints. AAs do not think like we do and have no feeling about what is going on as they do not carry human constraint and cannot generate meanings like we do. Another consequence is that it is not possible today to design living machines, as we do not know the nature of life. True AL is not possible today.
Understandings about the nature of life and human consciousness are needed to design AAs capable of behaving like animals and thinking like humans. As life is less complex and easier to understand than consciousness, the transfer of a "stay alive" constraint should be addressed first. An option could be to extend life with its "stay alive" constraint to AAs. The AA would then be submitted to the constraints brought in by the living entity.

7 CONTINUATIONS

We have proposed here that TT, CRA and SGP can be understood as being about the capability for AAs to generate human like meanings. The question "can machines think?" is then reformulated as "can AAs generate human-like meanings?" The focus is then on meaning generation in animals, humans and AAs. The MGS approach applied to TT, CRA and SGP has shown that the constraints to be satisfied are at the core of a meaning generation process. We feel that an evolutionary approach to the nature of constraints could allow an interesting perspective on their nature. Identifying the origin of biological constraints relatively to physico-chemical laws may allow to introduce an evolutionary theory of meaning. Work is in process on these subjects [6, 16]. But human constraints remain ill-defined and not really understood. As said, they are tightly linked to self-consciousness and free will which are mysteries for today science and philosophy. Significant work is to be done in this area, where a better understanding of human mind is needed.
The MGS approach also offers the possibility to define meaningful representations that embed agents in their environments [6]. Such representations can be used as tools in an evolutionary approach to self-consciousness where the human constraints play a key role. Work is in process in this area also [12].
An evolutionary approach to human constraints brings to focus first on the ones shared with animals, and more precisely on the "stay alive" constraint. As introduced above, we feel it could be interesting to look at extending life with its performances to AAs in order to bring the "stay alive" constraint to AAs. Some living entity in the AA, with the AA keeping control on it and being submitted to the "stay alive" constraint, like animals are. Investigating such an approach calls for developments which are beyond the scope of this paper.
Ethical concerns have been raised through the possible relations between human constraints and human values. If AAs can someday carry human constraints, they may not carry human values. An evolutionary approach to human consciousness could bring some openings on that perspective by positioning an evolutionary background for constraints and values. Such concern applies also to the possibility of AAs creating their own constraints that may be different from human ones and consequently not linked to human values.

REFERENCES

[1] Turing, A.M. (1950). "Computing machinery and intelligence. *Mind*, 59, 433-460.
[2] Searle, J. R. (1980) "Minds, brains and programs". *Behavioral and Brain Sciences* 3: 417-424.
[3] Harnad, S. (1990). "The Symbol Grounding Problem" *Physica:* 335-246.
[4] Harnad, S. (2008) "The Annotation Game: On Turing (1950) on Computing, Machinery, and Intelligence" in: *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*, Springer. ISBN 13: 978-1-4020-6708-2
[5] Menant, C. "Information and Meaning" *Entropy* 2003, 5, 193-204, http://cogprints.org/3694/1/e5020193.pdf
[6] Menant. C. (2010). "Computation on Information, Meaning and Representations. An Evolutionary Approach" in *Information and Computation: Essays on Scientific and Philosophical Understanding of Foundations of Information and Computation*, G. Dodig-Crnkovic, M.Burgin. ISBN-10: 9814295477
[7] Rodriguez, D. and all: "Meaning in Artificial Agents: The Symbol Grounding Problem Revisited". *Minds & Machines*, Dec 2011.
[8] Philpapers "Nature of Consciousness" http://philpapers.org/s/nature%20of%20consciousness
[9] Philpapers "Nature of life" http://philpapers.org/s/nature%20of%20life
[10] "Scientists create computing building blocks from bacteria and DNA" *News and Events. Imperial College London. 18 Oct 2011.* http://www3.imperial.ac.uk/newsandeventspggrp/imperialcollege/newssummary/news_18-10-2011-16-7-29
[11] "Hybrid Insect Micro Electromechanical Systems (HI-MEMS).. http://www.darpa.mil/Our_Work/MTO/Programs/Hybrid_Insect_Micro_Electromechanical_Systems_%28HI-MEMS%29.aspx].
[12] Menant, C. "Evolutionary Advantages of Inter-subjectivity and Self-Consciousness through Improvements of Action Programs". TSC 2010 http://cogprints.org/6831/

[13] Taddeo, M., Floridi, L. “Solving the Symbol Grounding Problem: Critical Review of Fifteen Years of Research” *Journal of Experimental & Theoretical Artificial Intelligence*, Volume 17, Number 4, Number 4/December 2005.

[14] Menant, C. " Information and Meaning in Life, Humans and Robots".*In Procs.Foundation of Information Sciences 2005, Paris*. http://cogprints.org/4531/1/F.45.paper.pdf

[15] Menant, C. " Evolution and Mirror Neurons. An introduction to the Nature of Self-Consciousness". *TSC 2005*.

[16] Riofrio, W. (2007) Informational Dynamic Systems: Autonomy, Information, Function. *In Worldviews, Science, and Us: Philosophy and Complexity*, edited by C. Gershenson, D. Aerts, and B. Edmonds. World Scientific, Singapore, pp. 232-249. http://www.worldscibooks.com/chaos/6372.html

Alan Turing's Legacy: Info-Computational Philosophy of Nature

Gordana Dodig-Crnkovic[1]

Abstract. Alan Turing's pioneering work on computability, and his ideas on morphological computing support Andrew Hodges' view of Turing as a natural philosopher. Turing's natural philosophy differs importantly from Galileo's view that the book of nature is written in the language of mathematics (The Assayer, 1623). Computing is more than a language of nature as computation produces real time physical behaviors. This article presents the framework of Natural Info-computationalism as a contemporary natural philosophy that builds on the legacy of Turing's computationalism. Info-computationalism is a synthesis of Informational Structural Realism (the view that nature is a web of informational structures) and Natural Computationalism (the view that nature physically computes its own time development). It presents a framework for the development of a unified approach to nature, with common interpretation of inanimate nature as well as living organisms and their social networks. Computing is understood as information processing that drives all the changes on different levels of organization of information and can be modeled as morphological computing on data sets pertinent to informational structures. The use of info-computational conceptualizations, models and tools makes possible for the first time in history the study of complex self-organizing adaptive systems, including basic characteristics and functions of living systems, intelligence, and cognition.

1 Turing and Natural Philosophy

Andrew Hodges [1] describes Turing as a Natural philosopher: "He thought and lived a generation ahead of his time, and yet the features of his thought that burst the boundaries of the 1940s are better described by the antique words: natural philosophy." Turing's natural philosophy differs from Galileo's view that the book of nature is written in the language of mathematics (The Assayer, 1623). Computation is not just a language of nature; it is the way nature behaves. Computing differs from mathematics in that computers not only calculate numbers, but more importantly they can produce real time physical behaviours.

Turing studied a variety of natural phenomena and proposed their computational modeling. He made a pioneering contribution in the elucidation of connections between computation and intelligence and his work on morphogenesis provides evidence for natural philosophers' approach. Turing's 1952 paper on morphogenesis [2] proposed a chemical model as the basis of the development of biological patterns such as the spots and stripes that appear on animal skin.

[1] School of Innovation, Design and Engineering, Mälardalen University, Sweden. Email: gordana.dodig-crnkovic@mdh.se

Turing did not originally claim that the physical system producing patterns actually performs computation through morphogenesis. Nevertheless, from the perspective of info-computationalism, [3,4] argues that morphogenesis is a process of morphological computing. Physical process, though not computational in the traditional sense, presents natural (unconventional), physical, morphological computation.

An essential element in this process is the interplay between the informational structure and the computational process – information self-structuring. The process of computation implements physical laws which act on informational structures. Through the process of computation, structures change their forms, [5]. All computation on some level of abstraction is morphological computation – a form-changing/form-generating process, [4].

In this article, info-computationalism is identified as a new philosophy of nature providing the basis for the unification of knowledge from currently disparate fields of natural sciences, philosophy, and computing. An on-going development in bioinformatics, computational biology, neuroscience, cognitive science and related fields shows that in practice biological systems are currently already studied as information processing and are modelled using computation-theoretical tools [6,7,8].

Denning declares: "Computing is a natural science" [9] and info-computationalism provides plenty of support for this claim. Contemporary biologists such as Kurakin [10] also add to this information-based naturalism, claiming that "living matter as a whole represents a multiscale structure-process of energy/matter flow/circulation, which obeys the empirical laws of nonequilibrium thermodynamics and which evolves as a self-similar structure (fractal) due to the pressures of competition and evolutionary selection". [11, p5]

2 Universe as Informational Structure

The universe is, from the metaphysical point of view, "nothing but processes in structural patterns all the way down" [12, p228]. Understanding patterns as information, one may infer that information is a fundamental ontological category. The ontology is scale-relative. What we know about the universe is what we get from sciences, as "special sciences track real patterns" [12, p242]. This idea of an informational universe coincides with Floridi's Informational Structural Realism [13,14]. We know as much of the world as we explore and cognitively process:

"Reality in itself is not a source but a resource for knowledge. Structural objects (clusters of data as relational entities) work epistemologically like constraining affordances: they allow or invite certain constructs (they are affordances for the

information system that elaborates them) and resist or impede some others (they are constraints for the same system), depending on the interaction with, and the nature of, the information system that processes them." [13, p370].

Wolfram [15] finds equivalence between the two descriptions – matter and information:

"[M]atter is merely our way of representing to ourselves things that are in fact some pattern of information, but we can also say that matter is the primary thing and that information is our representation of that. It makes little difference, I don't think there's a big distinction – if one is right that there's an ultimate model for the representation of universe in terms of computation." [16, p389].

More detailed discussion of different questions of the informational universe, natural info-computationalism including cognition, meaning and intelligent agency is given by Dodig Crnkovic and Hofkirchner in [17].

3 The Computing Universe – Naturalist Computationalism

Zuse was the first to suggest (in 1967) that the physical behavior of the entire universe is being computed on a basic level, possibly on cellular automata, by the universe itself, which he referred to as "Rechnender Raum" or Computing Space/Cosmos. Consequently, Zuse was the first pancomputationalist (natural computationalist), [18]. Chaitin in [19, p.13] claims that the universe can be considered to be a computer "constantly computing its future state from its current state, constantly computing its own time-evolution account!" He quotes Toffoli, pointing out that "actual computers like your PC just hitch a ride on this universal computation!"

Wolfram too advocates for a pancomputationalist view [15], a new dynamic kind of reductionism in which the complexity of behaviors and structures found in nature are derived (generated) from a few basic mechanisms. Natural phenomena are thus the products of computation processes. In a computational universe new and unpredictable phenomena emerge as a result of simple algorithms operating on simple computing elements such as cellular automata, and complexity originates from the bottom-up emergent processes. Cellular automata are equivalent to a universal Turing Machine. Wolfram's critics remark, however, that cellular automata do not evolve beyond a certain level of complexity; the mechanisms involved do not produce evolutionary development. Wolfram meets this criticism by pointing out that cellular automata are models and as such surprisingly successful ones. Also Fredkin [20] in his Digital philosophy builds on cellular automata, suggesting that particle physics can emerge from cellular automata. For Fredkin, humans are software running on a universal computer.

Wolfram and Fredkin, in the tradition of Zuse, assume that the universe is, on a fundamental level, a discrete system, and is thus suitably modelled as an all-encompassing digital computer. However, the computing universe hypothesis (natural computationalism) does not critically depend on the discreteness of the physical world, as there are digital as well as analog computers. On a quantum-mechanical level, the universe performs computation on characteristically dual wave-particle objects [21], i.e. both continuous and discrete computing. Maley [22] demonstrates that it is necessary to distinguish between analog and continuous, and between digital and discrete representations. Even though typical examples of analog representations use continuous media, this is not what makes them analog. Rather, it is the relationship that they maintain with what they represent. Similar holds for digital representations. The lack of proper distinctions in this respect is a source of much confusion on discrete vs. continuous computational models.

Moreover, even if in some representations it may be discrete (and thus conform to the Pythagorean ideal of number as a principle of the world), computation in the universe is performed at many different levels of organization, including quantum computing, bio-computing, spatial computing, etc. – some of them discrete, others continuous. So computing nature seems to have a use for both discrete and continuous computation, [23].

4 Information Processing Model of Computation

Computation is nowadays performed by computer systems connected in global networks of multitasking, interacting devices. The classical understanding of computation as syntactic mechanical symbol manipulation performed by an isolated computer is being replaced by the information processing view by Burgin, [24]. Info-computationalism adopts Burgin definition of computation as information processing.

In what follows, I will focus on explaining this new idea of computation, which is essentially different from the notion of context-free execution of a given procedure in a deterministic mechanical way. Abramsky summarizes this changing paradigm of computing as follows:

"Traditionally, the dynamics of computing systems, their unfolding behaviour in space and time has been a mere means to the end of computing the function which specifies the algorithmic problem which the system is solving. In much of contemporary computing, the situation is reversed: the purpose of the computing system is to exhibit certain behaviour. (…)

We need a theory of the dynamics of informatic processes, of interaction, and information flow, as a basis for answering such fundamental questions as: What is computed? What is a process? What are the analogues to Turing completeness and universality when we are concerned with processes and their behaviours, rather than the functions which they compute?" [25, p483]

According to Abramsky, there is a need for second generation models of computation, and in particular there is a need for process models such as Petri nets, Process Algebra, and similar. The first generation models of computation originated from problems of formalization of mathematics and logic, while processes or agents, interaction, and information flow are genuine products of the development of computers and Computer Science. In the second generation models of computation, previous isolated systems with limited interactions with the environment are replaced by processes or agents for which interactions with each other and with the environment are fundamental.

As a result of interactions among agents and with the environment, complex behaviour emerges. The basic building block of this interactive approach is the agent, and the fundamental operation is interaction. The ideal is the computational behaviour of an organism, not mechanical machinery. This approach works at both the macro-scale (such as processes in operating systems, software agents on the Internet, transactions, etc.) and on the micro-scale (from program implementation, down to hardware).

The above view of the relationship between information and computation presented in [25] agrees with ideas of info-computational naturalism of Dodig-Crnkovic [3] which are based on the same understanding of computation and its relation to information. Implementation of info-computationalism, interactive computing (such as, among others, agent-based) naturally suits the purpose of modelling a network of mutually communicating processes/agents, see [3,4,5].

5 Natural Computation

Natural computing is a new paradigm of computing which deals with computability in the natural world. It has brought a new understanding of computation and presents a promising new approach to the complex world of autonomous, intelligent, adaptive, and networked computing that has emerged successively in recent years. Significant for Natural computing is a bidirectional research [7]: as natural sciences are rapidly absorbing ideas of information processing, computing is concurrently assimilating ideas from natural sciences.

The classical mathematical theory of computation was devised long before global computer networks. Ideal, classical theoretical computers are mathematical objects and they are equivalent to algorithms, Turing machines, effective procedures, recursive functions or formal languages. Compared with new computing paradigms, Turing machines form the proper subset of the set of information processing devices, in much the same way as Newton's theory of gravitation presents a special case of Einstein's theory, or Euclidean geometry presents a limited case of non-Euclidean geometries, [5].

Natural/Unconventional computing as a study of computational systems includes computing techniques that take inspiration from nature, use computers to simulate natural phenomena or compute with natural materials (such as molecules, atoms or DNA). Natural computation is well suited for dealing with large, complex, and dynamic problems. It is an emerging interdisciplinary area closely related to artificial intelligence and cognitive science, vision and image processing, neuroscience, systems biology and bioinformatics, to mention but a few.

Computational paradigms studied by natural computing are abstracted from natural phenomena such as self-* attributes of living (organic) systems (including -replication, -repair, -definition and -assembly), the functioning of the brain, evolution, the immune systems, cell membranes, and morphogenesis.

Unlike in the Turing model, where the Halting problem is central, the main issue in Natural computing is the adequacy of the computational response (behaviour). The organic computing system adapts dynamically to the current conditions of its environments by self-organization, self-configuration, self-optimization, self-healing, self-protection and context-awareness. In many areas, we have to computationally model emergence which is not algorithmic according to Cooper [26] and Cooper and Sloman [27]. This makes the investigation of computational characteristics of non-algorithmic natural computation (sub-symbolic, analog) particularly interesting.

In sum, solutions are being sought in natural systems with evolutionary developed strategies for handling complexity in order to improve complex networks of massively parallel autonomous engineered computational systems. Research in theoretical foundations of Natural computing is needed to improve understanding of the fundamental level of computation as information processing which underlies all computing.

6 Information as a Fabric of Reality

"Information is the difference that makes a difference. " [29]

More specifically, Bateson's difference is the difference in the world that makes the difference for an agent. Here the world also includes agents themselves. As an example, take the visual field of a microscope/telescope: A difference that makes a difference for an agent who can see (visible) light appears when she/he/it detects an object in the visual field. What is observed presents a difference that makes the difference for that agent. For another agent who may see only ultra-violet radiation, the visible part of the spectrum might not bring any difference at all. So the difference that makes a difference for an agent depends on what the agent is able to detect or perceive. Nowadays, with the help of scientific instruments, we see much more than ever before, which is yet further enhanced by visualization techniques that can graphically represent any kind of data.

A system of differences that make a difference (information structures that build information architecture), observed and memorized, represents the fabric of reality for an agent. Informational Structural Realism [13] [30] argues exactly that: information is the fabric of reality. Reality consists of informational structures organized on different levels of abstraction/resolution. A similar view is defended in [12]. Dodig Crnkovic [3] identifies this fabric of reality (Kantian 'Ding an sich') as *potential information* and makes the distinction between it and *actual information* for an agent. Potential information for an agent is all that exists as not yet actualized for an agent, and it becomes information through interactions with an agent for whom it makes a difference.

Informational structures of the world constantly change on all levels of organization, so the knowledge of structures is only half the story. The other half is the knowledge of processes – information dynamics.

It is important to note the difference between the potential information (world in itself) and actual information (world for an agent). Meaningful information, which is what in everyday speech is meant by information, is the result of interaction between an agent and the world. Meaning is use, and for an agent information has meaning when it has certain use. Menant [31] proposes to analyze relations between information, meaning and representation through an evolutionary approach.

7 Info-Computationalism as Natural Philosophy

Info-computationalist naturalism identifies computational process with the dynamic interaction of informational structures. It includes digital and analog, continuous and discrete, as phenomena existing in the physical world on different levels of organization. Our present-day digital computing is a subset of a more general Natural computing. In this framework, computational processes are understood as natural computation, since information processing (computation) is not only found in human communication and computational machinery but also in the entirety of nature.

Information represents the world (reality as an informational web) for a cognizing agent, while information dynamics (information processing, computation) implements physical laws through which all the changes of informational structures unfold.

Computation, as it appears in the natural world, is more general than the human process of calculation modelled by the Turing machine. Natural computing takes place through the interactions of concurrent asynchronous computational processes, which are the most general representation of information dynamics [5].

8 Conclusions

Alan Turing's work on computing machinery, which provided the basis for artificial intelligence and the study of its relationship to natural intelligence, together with his computational models of morphogenesis, can be seen as a pioneering contribution to the field of Natural Computing and the Computational Philosophy of Nature. Today's info-computationalism builds on the tradition of Turing's computational Natural Philosophy. It is a kind of epistemological naturalism based on the synthesis of two fundamental cosmological ideas: the universe as informational structure (informationalism) and the universe as a network of computational processes (pancomputationalism/naturalist computationalism).

Information and computation in this framework are two complementary concepts representing structure and process, being and becoming. Info-computational conceptualizations, models and tools enable the study of nature and its complex, dynamic structures, and uncover unprecedented new possibilities in the understanding of the connections between earlier unrelated phenomena of non-living and living nature [28].

REFERENCES

[1] A. Hodges, *Turing. A Natural philosopher*. London: Phoenix, 1997.

[2] A. M. Turing, The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society of London*, 237(641), 37-72, 1952.

[3] G. Dodig-Crnkovic, *Information and Computation Nets. Investigations into Info-computational World*. Information and Computation (pp. 1-96). Saarbrucken: Vdm Verlag, 2009.

[4] G. Dodig-Crnkovic, The Info-computational Nature of Morphological Computing. In V. C. Müller (Ed.), *Theory and Philosophy of Artificial Intelligence* (SAPERE., p. forthcoming). Berlin: Springer, 2012.

[5] G. Dodig-Crnkovic, Significance of Models of Computation from Turing Model to Natural Computation. *Minds and Machines*, *21*(2), 301-322. Springer, 2011.

[6] G. Rozenberg, & L. Kari, The many facets of natural computing. *Communications of the ACM*, 51, 72–83. 2008.

[7] L. F. Landweber, and, & L. Kari, The evolution of cellular computing: nature's solution to a computational problem. *Biosystems*, 52(1-3), 3-13. 1999.

[8] K. Van Hornweder, Models and Mechanisms of the Morphogenesis of Biological Structures (Technical Report UT-CS-11-681). ftp://cs.utk.edu/pub/TechReports/2011/ut-cs-11-681.pdf, 2011.

[9] P. Denning, Computing is a natural science. *Comm. ACM*, 50(7), 13-18 http://cs.gmu.edu/cne/pjd/PUBS/CACMcols/cacmJul07.pdf, 2007.

[10] A. Kurakin, Scale-free flow of life: on the biology, economics, and physics of the cell. *Theoretical biology & medical modelling*, 6(6). 2009.

[11] A. Kurakin, The self-organizing fractal theory as a universal discovery method: the phenomenon of life. *Theoretical Biology and Medical Modelling*, 8(4). http://www.tbiomed.com/content/8/1/4 2011.

[12] J. Ladyman, D. Ross, D. Spurrett, & J. Collier, *Everything must go: metaphysics naturalised*. Oxford: Clarendon Press. http://www.worldcat.org/title/everything-must-go-metaphysics-naturalised/oclc/84150765&referer=brief_results 2007.

[13] L. Floridi, (2008). A defense of informational structural realism. *Synthese*, 161(2), 219-253.

[14] L. Floridi, Against digital ontology. *Synthese*, 168(1), 151–178, 2009.

[15] S. Wolfram, *A New Kind of Science*. Wolfram Media. 2002.

[16] H. Zenil, *Randomness Through Computation: Some Answers, More Questions*. Singapore: World Scientific Pub Co Inc. (2011).

[17] G. Dodig Crnkovic, and, & W. Hofkirchner, Floridi's "Open Problems in Philosophy of Information", Ten Years After. *Information*, 2(2), 327-359, 2011.

[18] K. Zuse, *Rechnender Raum*. Braunschweig: Friedrich Vieweg & Sohn, 1969.

[19] G. Chaitin, Epistemology as Information Theory: From Leibniz to Ω. In G. Dodig Crnkovic (Ed.), *Computation, Information, Cognition – The Nexus and The Liminal*, 2-17. Newcastle UK: Cambridge Scholars Publishing, 2007.

[20] E. Fredkin, Finite Nature. XXVIIth Rencotre de Moriond, 1992.

[21] S. Lloyd, *Programming the universe: a quantum computer scientist takes on the cosmos* (1st ed.). New York: Knopf, 2006.

[22] C. J. Maley, Analog and digital, continuous and discrete. *Philos. Stud*, 155, 117–131, 2010.

[23] A. Lesne, The discrete versus continuous controversy in physics. *Mathematical Structures in Computer Science*, 17, 185–223, 2007.

[24] Burgin, M. *Super-Recursive Algorithms* (Monographs., pp. 1-320). New York: Springer-Verlag New York Inc., 2004.

[25] S. Abramsky, Information, Processes and Games. In J. Benthem van & P. Adriaans (Eds.), *Philosophy of Information*, 483-549. Amsterdam, The Netherlands: North Holland, 2008.

[26] S. B. Cooper, B. Löwe, & A. Sorbi, *New Computational Paradigms. Changing Conceptions of What is Computable. Springer Mathematics of Computing series, XIII.* (S. B. Cooper, B. Löwe, & A. Sorbi, Eds.). Springer. 2008.

[27] G. Dodig-Crnkovic, and, & M. Burgin, *Information and Computation*, Singapore: World Scientific Pub Co Inc., 2011.

[28] G. Dodig-Crnkovic, and, & V. Müller, A Dialogue Concerning Two World Systems: Info-Computational vs. Mechanistic. In G. Dodig Crnkovic & M. Burgin (Eds.), *Information and Computation*, 149-184, World Scientific. (2011).

[29] G. Bateson, *Steps to an Ecology of Mind*. Ballantine, NY, pp. xxv-xxvi. 1972.

[30] K. M. Sayre, *Cybernetics and the Philosophy of Mind*, Routledge & Kegan Paul, London. 1976.

[31] C. Menant, Computation on Information, Meaning and Representations. An Evolutionary Approach. In G. Dodig Crnkovic & M. Burgin *Information and Computation*, World Scientific. (2011).

All the links accessed at 08 06 2012

Natural Computation—A Perspective from the Foundations of Quantum Theory

Philip Goyal[1]

Abstract. The framework of classical physics is based on a mechanical conception of nature, a conception which is mirrored in the Turing model of computation. Quantum theory has, however, fundamentally challenged this conception. The mathematical formalism of quantum theory consists of a set of postulates, most of which are at odds with the corresponding postulates of classical physics. For example, quantum measurements may have a finite number of possible outcomes, are probabilistic, are disturbing of the measured system, and in general only yield information about a fraction of the state of the measured system. Moreover, the quantum formalism does not specify what kind of physical process constitutes a measurement. In the eighty-five years since its creation, these and other non-classical features have defied any coherent understanding in terms of a new conception of nature. In recent years, there have been numerous attempts to derive the mathematics of quantum theory from a small number of informationally-inspired physical postulates, and this is providing a new, clearer perspective on just what physical ideas are implicit in the quantum postulates. In this paper, I will outline one such derivation, describe some of its implications for the assumptions implicit in Turing's model of computation.

[1] University Albany (SUNY), USA, email: pgoyal@albany.edu

Nature-Like Computation and a Measure of Programmability[1]

Hector Zenil [2]

Abstract. I will propose an alternative behavioural definition of computation based on whether a system is capable of reacting to the environment—the input—as reflected in a measure of *programmability*. This will be done by using a phase transition coefficient previously defined in an attempt to characterise the evolution of cellular automata and other systems. This transition coefficient measures the sensitivity of a system to external stimuli and will be used to define the susceptibility of a system to being (efficiently) programmed in the context of a nature-like definition of computation.

Keywords: Nature-like computation; programmability; cellular automata; compressibility; philosophy of computation; Turing universality.

1 APPROACHES TO THE QUESTION OF COMPUTATION

What is (a) computation? What does it mean to compute something and how much sense does it make to talk about computation outside of mathematics? These fundamental questions have not yet received a satisfactory answer according to [23], and despite the well-known "Church-Turing thesis" (CT thesis).

The most important notion of computation, however, is the notion of digital computation, and its most important feature is that of programmability. Turing's abstract idea of a universal computer has turned out to be technologically feasible, showing that if physics does not compute, it at least supports computation as we can build concrete devices whose behaviour, despite being governed by the laws of physics, effectively implement general-purpose digital computation. More formally, given a fixed description of Turing machines, we say that a Turing machine U is universal if for all input s and a Turing machine M, U applied to $(< M >, s)$ halts if M halts on s and provides the same result as M with input s, and does not halt if M does not halt for s. In other words, U is capable of simulating M with input s, with M and s an arbitrary Turing machine and an arbitrary input for M.

So far the study of the limits of computation has succeeded in offering us insight into what computation might be. The borderline between the decidable and the undecidable has provided an essential intuition in our search for a better understanding of computation. One can, however, wonder just how much can be expected from such an approach, and whether other, alternative approaches to understanding computation may complement the knowledge and intuition it affords, specially in modern uses of the concept of computation in the context of nature and physics corresponding to situations in which objects or events are seen as computers or computations.

One such approach involves not the study of systems lying "beyond" the uncomputable limit (also called the Turing limit), but rather the study of the minimum requirements for reaching universal computation, through a focus on the 'smallest' possible systems capable of universal computation—how easy or complicated it is to build a universal Turing machine, and how efficient such machines may be. This minimalistic bottom-up approach is epitomised by Wolfram's programme [27] in its quest to study simple programs. The question behind is what enables universality in a computational setup. Does it originate from a rich supply of basic operations? Is universality a pervasive property of computational systems As Wolfram [27] has captured in his Principle of Computational Equivalence, and as more recently other authors (e.g. Davis [6]) has adopted? Meaning that it takes little to reach universality.

According to the semantics approach, a computation is a function that maps input onto output [17]. In most accounts of computational processes as realised by physical mechanisms, it is also often assumed that there is a one-to-one correspondence between the causality of the physical states and the states of a computation, as defined by some abstract model in which these can be represented. The traditional mapping-states definition of physical computation is probably inspired by formal semantics, in that it requires that a mapping be established between a model and a physical system, meaning that states and events in the model are used to label states and events observed in the system treated as mathematical objects. Nature, however, is not like standard computation. One cannot, for example, assign a meaning to a natural phenomenon to be mapped to the concept of a halting state without making arbitrary choices, nor is it always known what path nature has taken to produce a given outcome, regardless of whether we see this path as constituting a computation or not.

Usually in computation a system is prepared in an initial state, and is allowed to evolve through a trajectory of events occurring within the space of successive states, until it eventually reaches a state labeled as final. In nature, however, there is usually no such thing as an initial or final state; everything is part of a causal chain of other events, much more like cellular automata that do not naturally halt (other than in reaching some stable configuration, for example) unless they are arbitrarily stopped, and more in the context of random initial configurations rather than infinite blank tapes.

This more nature-like approach to defining computation is closely related to the common view that computation is information processing. David Deutsch [8], for one, has often claimed that the theory of computation has traditionally been studied almost entirely in the

[1] Invited Talk. *Symposium on Natural/Unconventional Computing and its Philosophical Significance*, **AISB/IACAP World Congress 2012 - Alan Turing 2012**.

[2] Behavioural and Evolutionary Theory Lab, Department of Computer Science, The University of Sheffield, UK, email: h.zenil@sheffield.ac.uk

abstract, as a topic in pure mathematics. Deutsch argues that computers are physical objects, and computations are physical processes, hence both computers and computations are governed by the laws of physics and not by pure mathematics.

Computation has, however, traditionally been defined in terms of mathematical functions or in terms of how a function is calculated. This has motivated to view computation either as the outcome of a mathematical function or as the study of the time that an algorithm takes to compute a function, which has been evidently extremely successful. Nevertheless, a purely behavioural definition of computation (and of a computer) in terms of whether a system is capable of reacting to the environment–the input–and proceeding to do something with it, may provide a definition that focuses on whether a system is capable of (efficiently) transferring information from its input to its output, which is in a strong sense what it means to program a system. It is this capacity for efficiently transferring information which will serve to indicate the system's susceptibility to being programmed. Clearly, by this definition one may not call something a computer if it takes in the input but leaves it unchanged, or if for any input one gets always the same output, but my claim is that between these two cases there is room for a behavioural definition. It will be, thus, whether one can program a system what makes it a computer.

2 A BEHAVIOURAL APPROACH TO NATURE-LIKE COMPUTATION

Significant effort has been invested in definitions of computation in denotational, operational and axiomatic terms. For example, most approaches prove that a computation and its object denotationally coincide (leading to the CT thesis), some have adopted operational approaches [7] with questions of whether their definitions are just too broad. The axiomatic approach has also been developed with some interesting results [11, 19]. Nevertheless, some authors have extended the definition of computation to physical objects and physical processes at different levels of physical reality [25, 9, 10, 27, 8, 21] ranging from the digital to the quantum. In [27], for instance, Wolfram states that "... all processes, whether they are produced by human effort or occur spontaneously in nature, can be viewed as computations."

Klaus Sutner [20] has this to say in regards to Wolfram's conception of computation in nature: "This [Wolfram's] assertion is not particularly controversial, though it does require a somewhat relaxed view of what exactly constitutes a computation—as opposed to an arbitrary physical process such as, say, a waterfall." However, the work of several of the aforementioned physicists and computer scientists does indeed permit us to claim that a waterfall is (or can be viewed as) a computational process.

Whether one regards the universe as performing a computation or all natural processes as computations, when something is identified in a particular way because it has a specific property, the aim is to construct a category that includes certain things which share that property and exclude those things that do not, so that one can distinguish one thing from another and claim that one has established a concept with a finite extension which is set apart within the domain of discourse.

But to make sense of the term "computation" in these contexts (modern views of physics), I propose a behavioural notion of nature-like computation (similar in spirit to the way the term physics-like computation has been coined [22, 20]) compatible with digital computation but meaningful in broader contexts independent of representations and possible carriers. This will require a measure of the degree of programmability of a system by means of a compressibility index ultimately rooted in the concept of algorithmic complexity. I ask whether two computations are the same if they look the same and I try to answer with a specific tool possessing the potential to capture a notion of qualitative behaviour.

The fact that we need hardware and software is an indication that we need a programmable substratum that can be made to compute something for us but Turing's main contribution vis-à-vis the concept of computational universality is that data and programs can be stored together in a single memory without any fundamental distinction. One can always write a specific-purpose machine with no input to perform any computation, and one can always write a program describing that computation as the input for a (universal) Turing machine, so in a strong sense there is a non-essential distinction between program and data. This is crucial, in that the same void distinction holds between hardware and software, as software can be seen as both data and program, and hardware can always be emulated in a program (even if it may appear obvious that hardware is ultimately needed to undertake a computation).

A programmer uses memory space and cpu cycles in a regular computer to perform a computation, but this is by no means an indication that computation requires a computer (say a PC), only that it needs a substratum. The behaviour of the substratum is the underlying property that makes something a computation, and what carries out the computation a computer.

The behavioural approach takes this abstraction from the substratum to the extreme (keeping it physical as opposed to mathematical), with its central question being whether one can program a system to behave in a desired way. This approach that bases itself on the extent to which a system can be programmed tells us to what degree a given system resembles a computer. It can therefore serve as an epistemological framework for interpreting the computational nature of a system in the broader modern sense of computation, particularly in a physical context.

As suggested by Sutner [20], it is reasonable to require that any definition of computation in the general sense, rather than being a purely logical description (e.g. in terms of recursion theory), should capture some sense of what a physical computation might be. While Sutner's suggestion [20] has similar motivations to ours, it differs from ours in that his aim is to map the behaviour of a system to the theory of computation, notably computational degrees. Sutner aligns his approach with his reading of the following claim made by Searle: [18] "Computational states are not discovered within the physics, they are assigned to the physics." Sutner adds "A physical system is not intrinsically a computer, rather it is necessary to interpret certain features of the physical system as representing a computation." This obliges Sutner to take into consideration the act of interpretation of a physical system and the observer. Sutner's observer's language maps the physical object to an interpretation of what the object does as a computational process. In Sutner's view the observer may in the process of interpretation slightly modify the computation without adding to or carrying out the computation attributed to the physical object. One can see Sutner's model as consisting of a pair of coupled automata, where one is the physical object and the other the observer. The observer is defined as an automaton constrained in computational power, capable of mapping (interpreting)–by way of a transducer–a physical object onto a computational process using electrical signals.

Sutner's approach [20] is dependent on aspects of the traditional theory of computation in that it requires a mapping, and strong assumptions are made as regards the physical object, the observer and

the mapping itself. We don't focus on these mappings but on the qualitative behaviour of a system, regardless of whether the mapping is known, can be known or even exists, although such a mapping should in principle exist under certain (strong) assumptions, but it only cares about the qualitative character of a computational process and not its inner workings.

We know that systems that nobody ever designed as computers are able to perform universal computation, for example Wolfram's Rule 110 [27, 4], and that this like other remarkably simple systems are capable of universal computation (e.g. Conway's game of Life or Langton's ant). These systems may be said to readily arise physically, as they have not been deliberately designed. There is, however, no universal agreement as regards the definition of what a computer may or may not be, or as to what exactly a computation might be, even though what computation is and what a computer may be are well grasped on an intuitive level.

A program can be defined as that which turns a general-purpose computer into a special-purpose computer. This is not a strange definition, since in the context of computer science a computation can be regarded as the evolution undergone by a system when running a program. However, while interesting in itself, and not without a certain affinity with our approach, this route through the definition of a general-purpose computer is a circuitous one to take to define computation. For it commits one to defining computational universality before one can proceed to define something more basic, something which ideally should not depend on such a powerful (and even more difficult-to-define) concept. Universal computation is without a doubt the most important feature of computation, but every time one attempts to define computation in relation to universal computation, one ends up with a circular statement [computation is (Turing) universal computation], thus merely leading to a version of a CT thesis.

It encompasses minds and computers while excluding almost everything else, investing minds and computers with a special status while viewing most of the rest of reality as computationally vacuous. I think this approach is weak, however. Think of the billiard ball computational model. It is designed to perform as a computer and can therefore be trivially mapped onto the states of a digital computer. Yet it is a counterexample of what the semantic account sets out to do, viz. to cordon off minds and computers (believed capable of computation) from things like billiard balls, tables and rocks (believed to be incapable of computation).

3 CASE STUDY: CELLULAR AUTOMATA

A cellular automaton (CA) is a computational model that has been shown to be an interesting object of study both as a computational device per se and for modelling all kinds of phenomena [27, 13]. A CA consists of an array of cells where each takes a value from a finite set of states. Every cell updates its value depending on the state of its neighbouring cells. Hence the global behaviour of the automaton depends on the local interaction of its cells.

But what does a CA compute? As shown by Wolfram [27], the evolution of a system like a cellular automaton can be viewed as a computation. As shown in [27] (page 638), ECA Rule 132 (R132) is a simple cellular automaton whose evolution effectively computes the remainder after division of a number by 2. Starting from a row of n black cells, 0 black cells survive if n is even, and 1 black cell survives if n is odd. So in effect this cellular automaton can be viewed as computing whether a given number is even or odd. Wolfram provides other CA examples computing functions in the traditional sense (e.g. R94 as enumerating even numbers; R62 that can be thought of as enumerating numbers that are multiples of 3; the central column of the pattern of R129 that can be thought of as enumerating numbers that are powers of 2; or a CA, with 16 states, as capable of computing prime numbers).

The CA community has developed a strong intuition for determining the ability of a CA to transmit information and eventually be considered a candidate for universal computation. Evident properties of rules like the game of life [5] (a 2-dimensional cellular automaton proven to be computationally universal) and of rules like R110 [27], (a one-dimensional nearest neighbourhood) simple cellular automata, are structures persisting over time but sensitive to perturbations. These structures transmit information through a system, for example, in the form of characteristic gliders and all sorts of other well-known structures. These structures are unpredictable in a fundamental way if the system is capable of universal computation (as we will learn below from the work of Gödel and Turing). Predictable rules, or rules with no persistent structures, are often discarded as incapable of carrying messages and behaving as universal computers. Nevertheless, CAs computing in a one-dimensional space, with only 2 states and nearest neighbour have already sufficient internal richness, in spite of this simplicity, to simulate a cyclic tag system for implementing a universal computing device [4, 27].

Wolfram noticed [27] this richness, and by careful visual inspection of the evolution of two-dimensional space-time orbits, he was able to classify all the various behaviours into 4 general classes for systems starting with a random initial condition. Wolfram provided a 4-group classification of behaviour (particularly for cellular automata, specially the so-called elementary i.e. 1-range neighbourhood). His classes can be regarded as reflecting how information from the initial state is retained in the final configuration in a system (e.g. a cellular automaton). Class I, for example, is either unable to transfer any information to future states or simply transfers all or a portion of the information exactly as it came in. For Class II, however, information always remains completely localised into rigid patterns (e.g. fractals). On the other hand, Class III can be seen as scrambling the information from the input, allowing little chance to recover the information from the output because it generates a sort of noise (what Wolfram calls intrinsic randomness) even from the simplest inputs (e.g. a single black cell). Class IV, however, transfer information from the input through the system, interacting with other structures, but neither unfolding into simple structures such as those in Class II nor scrambling the information as happens in Class III.

A measure based on the change of the asymptotic direction of the size of the compressed evolutions of a system for different initial configurations (following a proposed Gray-code enumeration of initial configurations) was presented in [28]. It gauges the resiliency or sensitivity of a system vis-à-vis its initial conditions. This phase transition coefficient led to an interesting characterisation and classification of systems, which when applied to elementary CA, yielded exactly Wolfram's four classes of systems behaviour, with no human intervention. The coefficient works by compressing the changes of the different evolutions through time, normalised by evolution space, and it is rooted in the concept of algorithmic complexity.

3.1 A measure of programmability

Based on the principles of algorithmic complexity, one can use the result of the compression algorithms applied to the evolution of a system to characterise the behaviour of a system [28] by comparing it to its uncompressed evolution as it is captured in eq. 1. If the evolution is too random, the compressed version won't be much shorter

than the length of the original evolution itself. It is clear that one can characterise systems by their behaviour [28]: if they are compressible they are simple, otherwise they are complex (random-looking). The approach can be taken further and used to detect phase transitions, as shown in [28], given that one can detect differences between the compressed versions of the behaviour of a system for different initial configurations. This second measure allows us to characterise systems by their sensitivity to the environment: the more sensitive the greater the variation in length of the compressed evolutions. A classification places at the top systems that can be considered to be both efficient information carriers and highly programmable, given that they react succinctly to input perturbations. Systems that are too perturbable, however, do not show phase transitions and are grouped as inefficient information carriers. The efficiency requirement is to avoid what is known as Turing tarpits [14], that is, systems that are capable of universal computation but are actually very hard to program. This means that there is a difference between what can be achieved in principle and the practical ability of a system to perform a task. This approach is therefore sensitive to the practicalities of programming a system rather than to its potential theoretical capability of being programmed.

The transition coefficient is derived from a characteristic exponent and is defined as follows: Let the characteristic exponent c_n^t be defined as the mean of the absolute values of the differences between the compressed lengths of the outputs of the system M running over the initial segment of initial conditions i_j with $j = \{1, \ldots, n\}$ following the numbering scheme devised in [28] based on the Gray-code, and running for t steps in intervals of n. Formally,

$$c_n^t = \frac{|C(M_t(i_1)) - C(M_t(i_2))| + \ldots + |C(M_t(i_{n-1})) - C(M_t(i_n))|}{t(n-1)} \quad (1)$$

Let C denote the transition coefficient defined as $C(U) = f'(S_c)$, the derivative of the line that fits the sequence S_c by finding the least-squares as described in [28] with $S_c = S(c_t^n)$ for a chosen sample frequency n and running time t. $S(c_t^n)$ is simply a sequence of c_t^n for increasing t and fixed n. That is, to capture the asymptotic behaviour of M_t. The value $C_n^t(U)$ (simply C until the discussion of definitions in the next section) will be therefore an indicator of the degree of programmability of a system U relative to its external stimuli (input). The larger the derivative, the greater the change of U and therefore the possibility of program U to perform a task encoded in some form.

For example, according to this coefficient (or index) C, cellular automata (CA) with rule numbers 0 and 30 are close to each other because they remain the same despite the change of initial conditions (despite the choice of t and n), and they are hardly perturbable. The measure indicates that rules like rule 0 or rule 30 (denoted from now on as R0, R30, etc.) are incapable of transmitting information, given that they do not react to changes in the input. In this sense they are alike because there is no change in the qualitative behaviour of these CA when fed with different inputs, regardless of how different the inputs may be–and this is what C measures. Rule 0, for example, remains entirely blank, while R30 remains mostly random-looking, with no apparent emergent coherent propagating structures (other than the regular and linear pattern on one of the sides).

On the other hand, rules such as 122 and 89 have C close to each other because they are sensitive to initial conditions. As is shown in [28], they are both highly sensitive to initial conditions and present phase transitions which dramatically change their qualitative behaviour when starting from different initial configurations. This means that rules like 122 and 89 can be used to transmit information through the system, from the input to the output.

Values of C for the subclass of CA referred to as elementary (the simplest one-dimensional closest neighbourhood, also known as ECA [27]) have been calculated and published in [28], and a further investigation of the relation between this transition coefficient and the computational capabilities of certain known (Turing) universal machines has been undertaken in [30]. We will refrain from exact evaluations of C to avoid distracting the reader with numerical approximations that may detract from our particular goal in this paper. The aim here is to propose a behavioural definition of computation based on this measure rather than to evaluate specific values that have already been calculated in [30].

This transition coefficient will be used to dynamically define computation based on the *degree of programmability* of a system. The advantage of using the transition coefficient C is that it is indifferent to the internal states, formalism or architecture of a computer or computing model; it doesn't even specify whether a machine has to be digital or analog, or what its maximal computational power is. It is only based on the behaviour of the system in question. It allows us to minimally characterise the concept of computation on the basis of behaviour alone.

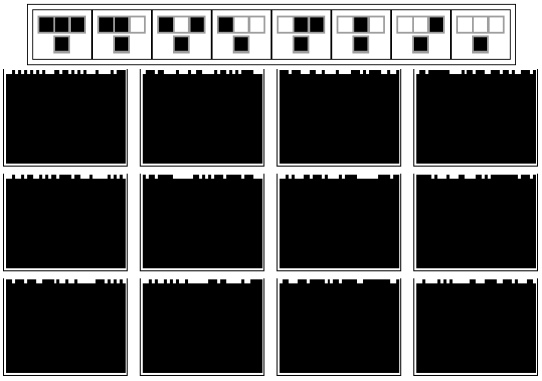


Figure 1. ECA R255 (equivalent by colour inversion to R0, R255 is used here for visual convenience) is stuck, unable to perform any computation– it does not react to any external stimulus. This is an illustration of a C-computer for C close (or equal) to zero [28]. The picture shows a series of evolutions for 12 random inputs (3 per row) next to the cellular automaton rule (top).

Let's denote as a C-computer (see Fig. 3.1) a system with programmability coefficient C capturing the capability of the system to transfer information from its input towards its output. Under this notation, R255 in Wolfram's one-dimensional elementary cellular automata (ECA) enumeration (Fig. 1), for example, is a 0-computer, that is a computer unable to carry out any operation because it cannot transfer any information from the input to the output (another way to say this is that R255 does not compute). ECA R255 cannot by any means be programmed to perform any task, despite the input. We have then captured the sense of what it means not to be a computer with the following definition:

Definition 1. A 0-computer is not a computer in any intuitive sense because it is not capable of carrying out any calculation.

A system capable of (Turing) universal computation (see Fig. 3.1) would therefore have a non-zero C limit value. C also captures some of the universal computational efficiency of the computer in that it has the advantage of capturing not only whether it is capable of re-

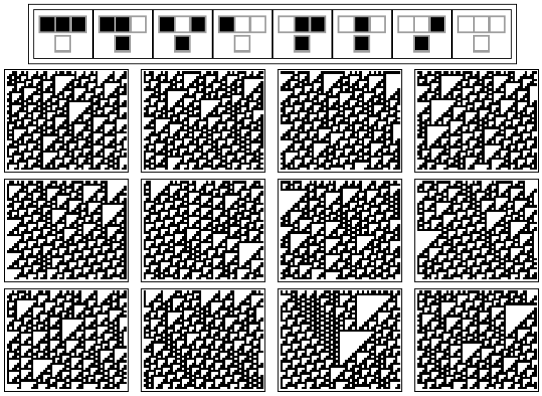


Figure 2. ECA R110 is efficient at carrying information through persistent local structures through the output reacting to external stimuli. Its C_n^t value for sensible choices of t and n [28] is compatible with the fact that it has been proven that R110 is capable of universal computation (it has been proven [27, 4] for a particular semi-periodic initial configuration).

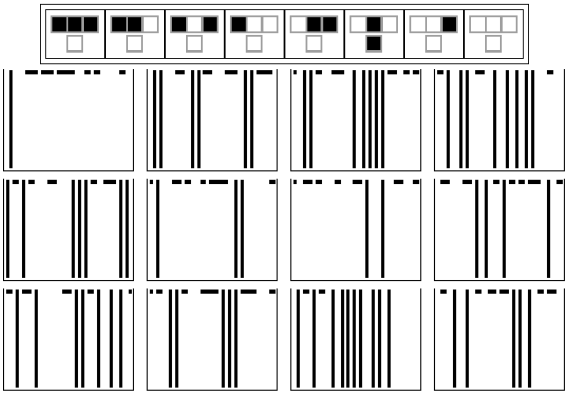


Figure 3. ECA R4 is a kind of program filter that only transfers bits in isolation (i.e. when its neighbours are both white). It is clear that one can perform some very simple computations with this automaton. One could not, for example, implement a typical logic gate based on its particular behaviour. It cannot clearly carry (Turing) universal computation. it has a low C for random chosen n and t [28].

acting to the input and transferring information through its evolution, but also the rate at which it does so. So C is an index of both capability in principle and ability in practice. A non-zero C means that there is a way to codify a program to make the system behave (efficiently) in one fashion or another, i.e. to be programmable. Something that is not programmable cannot therefore be taken to be a computer.

One can also see that things that seemed to behave like computers but were not called computers can indeed be considered computers under this approach. Mathematical functions, for example, can be considered C-computers for some C determined by the domain of the function. That a function can be considered a computer does not controvert the theory wherein a computer is defined in terms of a function and a domain, and a function in terms of an algorithm having the input as its arguments and the output as its function evaluation. A function, however, seems to require a carrier. Usually that carrier is a piece of paper and a pencil being wielded by a person, but it can also be a physical computer. Can the simple description of the function be considered a computer or a C-computer? I think it should not be. Something static shouldn't be considered to have a behaviour, and I think it can be captured by C. To evaluate C one needs to actually run a program, otherwise it remains unevaluated.

This makes for a clear distinction between, for example, a vision of the universe as a mathematical structure and a vision of the universe as a computer. While the latter may account for the physical carrier, implying that the computation is being carried out by the universe itself, it does not seem clear how a mathematical structure can come equipped with the carrier on which it should be executed, unless it becomes a computer program and therefore a computer.

Toy computers (e.g. Fig. 3) can also be considered C-computers, as indeed can everyday things like fridges or lamps. When one turns on a lamp the lamp is programmed to do something, in this case to turn on. Likewise when it is turned off. Even if trivial, it reacts to the input by producing light as the outcome. A fridge can be seen as cooling objects that are introduced into it, the output being the cooling– after an interval– of the objects in question. That both a lamp and a fridge can be viewed as C-computers for a very limited C, given that they have limited programmability (to perform a single, specific task), should not be surprising, at least not in light of the definition of a C-computer. With the advantage that one can now ask whether a lamp or a fridge is or isn't a computer without trivialising either the question or the answer. Under our formal and precise definition they are, as long as it is stated that they are limited in scope, as indicated by their behaviour as captured by the coefficient C, while an ordinary static table may be some kind of C-computer, certainly for C very close to 0, if it is thought to be computing anything at all. On the other hand, the universe as a whole can now legitimately be seen and treated in this context as a computer, as it is a C-computer for maximal C given that it contains all possible C-computers.

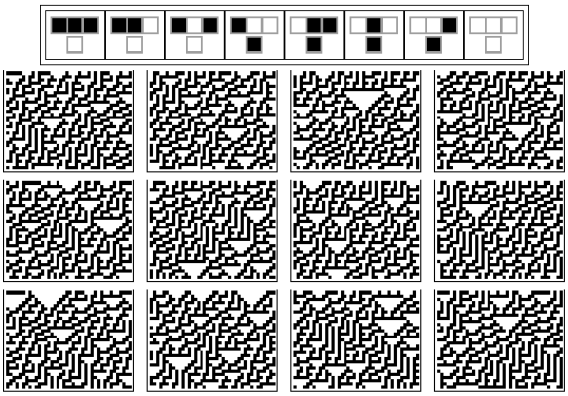


Figure 4. It is an open question whether ECA R30 can be programmed to perform computations. Its C value is low [28], meaning that it is not efficient for transferring information because it always behaves in the same fashion– too randomly–without reacting differently to external stimuli.

3.2 Reversibility, 0-computers and conservation laws

In [22], Margolus asserts that reversible cellular automata can actually be used as computer models embodying discrete analogues of classical notions in physics such as space, time, locality and microscopic reversibility. He suggests that one way to show that a given rule can exhibit complicated behaviour (and eventually universality) is to show (as has been done with the game of Life [5] and R110

[4, 27]) that "in the corresponding 'world' it is possible to have computers" starting these automata with the appropriate initial states, with digits acting as signals moving about and interacting with each other to, for example, implement a logical gate for digital computation. Wolfram reinforces this vision by suggesting, through his Principle of Computational Equivalence, that it is indeed the case that non-trivial behaviours inevitably lead to universal computation.

This does not mean that a system must necessarily be bijective (hence reversible) in its input/output mapping in order to be universal. But it is actually reversible CA with high entropy (number of possible states) which will tend to show the greatest behavioural richness and therefore be considered the best candidates for being classified as computers. In other words, the greater the richness a system is capable of, the greater C coefficient it will have. A reversible CA (RCA) has the property that starting it from a random state is like starting from a maximum entropy state in a thermodynamical system, because the RCA is not allowed to get simpler in its evolution, the only way to get simpler being to collapse the number of states, making it irreversible. Entropy in a randomly initiated RCA can only increase, but if it reaches maximum entropy it can't get any more complicated, and so nothing much happens. This is also captured by C, in that the RCA always look the same and are immune to evolutionary changes, presenting homogeneous local entropy everywhere.

RCA are interesting because they allow information to propagate, and in some sense they can be thought of as perfect computers– indeed in the sense that matters to us. If one starts an RCA from a non-uniformly random initial state, the RCA evolves, but because it cannot get simpler than its initial condition (for the same reason given for the random state) it can only get more complicated, producing a computational history that is reversible and can only lead to an increase in entropy. The RCA, however, is only reshaping the message that it got at the beginning in the form of an initial configuration, and so the amount of information in the RCA evolution remains the same. Which makes it a perfect example of a system with increasing entropy but consistent complexity over time. The algorithmic complexity of the RCA is the same because one can track the RCA back to the original information content represented by its initial configuration. So the state of the CA at any time always carries the same information content. In non-reversible CA, however, information can be lost, and even though the algorithmic complexity of the evolution of a CA is always the same, one cannot recover it a posteriori from any later state. In reversible CA, entropy, like information content, may increase or decrease over time. As Margolus himself states, it is one thing to know that a gas was in one corner at a given state, and another to return the gas from its expanded condition to its original position. It may thus seem that RCA in Wolfram's class III may all be chaotic, but Wolfram [26] offers examples of one-dimensional reversible cellular automata exhibiting three types of behaviour of local structures as they propagate in space.

In nature-like computation, conservation laws are important because the physical carrier on which a computation will be performed is governed by physical conservation laws (laws that conserve physical invariants such as mass, energy, momentum, etc.). In RCA, there are cases in which the simplest locally-computable invariants are cells whose values never change, and which are analogous to nature-like conservation laws. That is, laws such that for any given property, the physical state of the system does not change as the system evolves. The simplest RCA capable of doing this are those that ignore their neighbouring cells and only look at the central one, reproducing it identically. One may have doubts about calling these computers because there is no transformation of information whatsoever, with the system just letting pass through it anything that it is fed. Even worse, there are systems that may look as if they are computing the identity function while in fact performing a series of intermediate transformations which lead to the same output a few steps later. From the behavioural perspective based on the transition coefficient, under the qualitative definition the two would be behaving differently if they deliver their *richness* at different rates even if they produce the same output. This discussion helps us to see how close these computational systems are to physical phenomena and to purely behavioural descriptions, but also to address some potential concerns raised by the qualitative approach proposed herein.

4 PROGRAMMABILITY AND BEHAVIOURAL EQUIVALENCE

We can then define a system performing computation based on its behaviour simply as follows:

Definition 2. A system U computes if $C_n^t(U) > 0$ for some $t, n > 0$

Meaning that U can be programmed. Whether U can perform only certain computations or all computations will not depend only on C but on the details of U that escape the behavioural definition. Yet this definition suits a much broader sense of nature or physics-like computation as used in, for example, modern models of physics (to mention but a few examples [10, 25, 27, 8, 21]). One can see that there are systems that are not computers under this definition, simple ones such as R0 and R255 Elementary Cellular Automata (see Fig. 3.1). Notice that C depends on two parameters, t and n, from the original coefficient definition 1 indicating the number of steps that a system has run (t), and the sampling frequency (n). This is of course a downside of any *a posteriori* behavioural approach, and it is precisely what makes this empirical approach a difficult one. Nevertheless, one can do better and ideally define:

Definition 3. A system U has programmability $\lim_{t\to\infty} C_{n=1}^t(U) = m$

Meaning that the sampling frequency is $n = 1$ (i.e. the compression comparison is applied at every step for every initial condition at a time) and for all steps. Evidently this limit cannot be calculated in finite time ($t \to \infty$) by, say, a Turing machine. This is ultimately related to a problem of induction, that is, how to characterise the behaviour of a system that can start from a countable infinite number of possible states by looking only at a finite sample of them. If O' is an oracle Turing machine, however, then m can be computed, and fully describes the qualitative behaviour of U.

This means that equivalence in the theoretical sense is ultimately undecidable. In the empirical sense it can only be approached, given that the transition coefficient on which the qualitative definition of computation is based is limited by finite resources (reflected in the parameters t and n), providing only an approximate indication of the behavioural programmability of a system.

Notice that this is consistent with the behavioural approach, because if two systems have about the same C_n^t for n and t fixed it means that it does react to changes at about the same rate, so it may not only transfer or not information but if it does so or not it does so at the same rate if they both have the same C_n^t for that n and that t. By varying n and t one can also define rates of convergence to C making a refinement to the original definition (perhaps a subject for

a future continuation of this approach).

Clearly, under this definition, behaviour space is less dense than algorithm and program space because there may be different programs implementing different algorithms but leading to the same behaviour. So one can only define two behaviourally equivalent systems as follows:

Definition 4. A system U and U' are computationally equivalent in behavioural terms if $C(U) = C(U')$.

Simple examples of a behavioural computational class are C-computers for $C = 0$, i.e. they cannot be programmed, and are behaviourally equivalent. Under Def. 1 and 2, systems that are identified as 0-computers do not compute, as they are not capable of being programmed.

Experience tells us that something that behaves in a certain way will continue doing so, as we have empirically established in [29]. This can be justified by algorithmic probability, because the longer the observation time of a computing system the smaller the chance that the behaviour in question will radically change. So even though one cannot guarantee a behaviour ad infinitum, algorithmic probability may provide the stability required to make reliable generalisations. So one can weak Def. 4 by allowing $C(U)$ to be close enough to $C(U\prime)$ as follows:

Definition 5. A system U and U' are c computationally equivalent if $|C(U) - C(U')| < c$.

It is worth stressing that two systems (or computers) are not the same in any other sense if they have the same coefficient C. C is a measure of sensitivity (what I take as how programmable the system is); it cannot on its own indicate whether two computers compute the same function, and is therefore a different measure than that provided by traditional computability and formal semantics. It can tell when two computers diverge in their behaviour, because for two computers to be the same, a necessary but not sufficient condition is that they must both have the same transition coefficient (or to differ by a desired c), which would mean that they have the same capability of reacting to external stimuli, and transmit information at about the same rate. Because C itself depends on two parameters (n and t), this also means that C can only make comparisons for fixed t and n (the same runtime and the same sampling frequency) between two systems. So two C-computers are behaviourally equivalent if they have the same C.

For the same reason that one cannot tell whether a machine will halt for a given input, one cannot decide whether two computers compute the same function, but one can relate nature-like computation and abstract computation by means of Turing machines as follows: for every C-computer U, there exists a program P behaviourally equivalent to U, that is, with transition coefficient $C(U) = C(M)$ independent of n and t, because there exists a universal Turing machine T capable of reproducing the exact behaviour of U.

It is worth also noting that this behavioural definition is cumulative (but not additive), in the sense that a C-computer can be embedded in the workings of another C'-computer for $C \neq C'$. If the C$'$-computer does not impose any behavioural restriction on the C-computer, then clearly C$' \geq C$, given that the new computer will be capable of at least C-computation. This is the sense in which one may see R255 as a program in the context of a C-computer with $C \neq 0$ capable of running R255. If the C-computer is, for example, a universal computer, R255 would be a program but cannot by itself be a computer.

5 DISCUSSION

The topic and content of Nature-like computation is, on purpose, related to the question of whether the universe can be said to compute. It does, for we know there are C-computers in it capable of universal computation, but we don't really know whether the universe (e.g. as represented by its physical laws) constrains C, a limit broad enough to encompass every possible C-computer for a maximal C contained in the physical universe. One can think of the law of gravitation as *carrying out* some sort of *computation*, with the degree of programmability of such a system limited to performing a particular task (in this case pulling objects toward each other and keeping them in their gravitational trajectory). Classical mechanics guarantees that the system is deterministic, even if that doesn't mean one can predict the system for any specific parameters (e.g. 3 bodies). There is no fundamental reason, however, for following the approach described herein when assessing whether a system can compute based on its degree of programmability. Still, the fact that one can coarse grain what computation may mean by way of the parameter C, and guarantee that there are both systems with maximal C and $C = 0$ for systems that can be programmed to do something, and others that cannot be programmed at all and show no reaction to any external stimulus (e.g. see Fig. 3.1), imbues this approach and its definition of computers and computation with sense, particularly in the context of nature-like computation as proposed by some of the aforementioned authors. There are also C-computers for small values of C, meaning that the system can hardly be programmed because it does not transfer information efficiently enough (this may be the case, for example, with R30, see Fig. 3.1).

5.1 The question of scale

So far, the object of this behavioural approach to computation has been to provide a reasonable framework for assertions connecting the notion of computation to nature, and how nature may or may not compute, in light of current uses of the term 'compute'. Lloyd [21], for example, claims that since the universe is computing itself, things in the universe would therefore also be computing themselves. Think of the example of a still physical object (e.g. a desk or a sheet of paper). These objects would hardly compute anything at their macroscopic level, say an addition between any 2 numbers, yet they may be constituted at a molecular or atomic scale of particles capable of carrying out all sorts of computations, which unlike the objects, may be programmed, either as part of another system or in themselves. It is clear then that the span of behaviour at that scale is greater than at the scale of the object itself. But does it make sense to say that something computes itself? [21]. It may or it may not.

In the real world, things are constituted by smaller elements unless they are elementary particles. One therefore has to study the behaviour of a system at a given scale and not at all possible scales, otherwise the question becomes meaningless, as elements of a physical object are molecules, and ultimately atoms and particles that have their own behaviour, about which too the question about computation can be asked. This means that a C-computer may have a low or null C at some scale but contain C'-computers with $C' > C$ at another scale (for which the original object is no longer the same as a whole). A setup in which $C' \leq C$ is actually often common at some scale for any computational device. For example, a digital computer

is made of simpler components, each of which at some macroscopic level but independently of the interconnected computer is of lower behavioural richness and may qualify for a C of lower value. In other words, the behavioural definition is not additive in the sense that a C-computer can contain or be contained in another C'-computer such that $C \neq C'$.

Can R255, for example, be thought of as computing itself as it evolves? Under the qualitative definition, even if R255 is computing itself it cannot be programmed, and so is a 0-computer under our approach, a computer not capable of computation and therefore hardly a computer at all. On the other hand, R255 does not present any problem of scale as it represents itself at all scales. A table, however, is made of smaller components to which may be assigned some specific task, and one may even consider reprogramming the matter of which it is made, in the manner epitomised in the subfield of programmable matter. In which case one may say that the table is computing itself, since it could be computing something else out of its atoms. So the definition of a C-computer is scale dependent and its implementation in the real world is subtle, yet at the abstract level it seems to correspond to an interesting and well-delineated definition of computation based on its behavioural capabilities.

In the physical world, under this qualitative approach, things may compute or not depending on the scale at which they are studied. To say that a table computes only makes sense at the scale of the table, and as a C-computer it should have a very limited C, that is a very limited behaviour given that it can hardly be programmed to do something else.

5.2 Program versus physical realisation

The behavioural approach may need a major shake-up if used in a quantum context, given that our understanding of the mechanisms at the quantum scale are subject to various interpretations. For example, the standard interpretation considers quantum mechanics to be fundamentally non-deterministic, and so our definition of a deterministic computer (necessary to evaluate C) becomes inapplicable. If quantum particles are capable of, for example, being in all possible states at the same time when entangled, that means that they can perform every possible computation at the same time (which is at the core of the quantum computational paradigm as based on the concept of the quantum bit or qubit and taking advantage of quantum properties). Hence it would obviously have a transition coefficient C beyond any attained by a digital system, given that it would represent all possible behaviours at the same time–which in the macroscopic world would not be possible. If one takes atoms to be computers, or quantum computers, one can therefore trivially claim, as has been done by Lloyd [21], that the world is a quantum computer. In this case, the only content of such a claim, as opposed to its contrary (that the world is not a computer) concerns whether or not the world is a classical computer. Lloyd claims, as do Deutsch [8] and others, that it is not a classical computer, if it is a computer at all, but rather a quantum one, simply because computers, like everything else, rely on the very basic physical properties of our world.

When Gödel provided the proof of his incompleteness theorem what he did was to unify symbols and operators, just as Turing did for data and programs. Because Gödel's and Turingś approaches are extensionally equivalent, as long as one can find a Gödel numbering encoding a system, one can conclude that such a system can be interpreted as a program. For example, based on Davis' work encoding Diophantine equations, it would seem that the extension of what a program can be is formally quite large.

The consequence of universal computation is that hardware and software are not essentially different, for one can be encoded in the other. But in the real world why are hardware implementations of software faster? (e.g. the case of Intel's Pentium coprocessor: they could have certainly solved it with a patch, i.e. software, but that would perhaps have jeopardised the promise of a faster cpu). So are there real world differences between software and hardware, e.g. in execution time? It seems software always requires a transformational process– from a description to a physical embedding–in order to be executed. What makes a program, as a sequence of text, become a set of instructions that are executed? This is sometimes called the problem of computational implementation. The usual way to get round this problem is to separate programs from their physical implementations, on the grounds that the former are abstract while the latter are concrete, thus in some sense reinstating the difference between software and hardware when it comes to the physical world. At the fundamental level, however, and given that one can always (under Church-Turing's thesis) implement a program, the difference is not essential in nature.

Physically, computer programs may be a collection of punched cards or configurations in a magnetic tape. Is software part of a computer? If data and program can be exchanged one for the other, can software or hardware by themselves constitute a computer? Hardware alone may, if the computer is designed to serve a specific purpose, though it thereby loses its potential to reach computational universality. But what a purely software computer means may be unclear, and as suggested by Deutsch and Lloyd, the notion may make no sense. It may seem, for example, that the description of a Turing machine is pure software if no distinction is made between the input of the machine and its transition table (whether capable of universal computation or not). Is the difference only practical? Software seems not to have a physical execution carrier, and software before implementation may only be a description of a computation and not a computation per se, which means it cannot be executed in the real world until it finds a physical carrier. Is the description still a computer? I don't think so, but I don't aim to answer all these questions, and other authors have attempted to shed some light on such matters [24, 16]. The answer also seems related to a relativisation of software. Software written in a higher language like C or *Mathematica* is different from software written in machine language, and much closer to hardware.

The fact that we do need a computer for running, say, cellular automata rules, may be misleading, since it may suggest that there is always a need of hardware (and software) in the way we know and use it. This leads to the question of computation in nature, and ultimately to the question of the computer running the universe itself, if one embraces such an idea, and the associated question of whether such a computer, if it exists, is part of the universe or runs in some higher world.

The first thing to note is that the costs related to such a computer would be huge, on the order of the computational power of the universe itself, and likely to require even more than the energy in the universe itself, due to thermodynamical laws, if they apply at such a scale. This suggests that one need not look for a computer if one thinks that the computation comes equipped with its physical carrier or one would fall into an infinite hierarchy of worlds running each the program of a lower level universe. When the concept is at the same time matched and disentangled from its carrier in the behavioural approach, one can see that it is a particle at a time that creates the universe both carrying software and hardware together. As has been suggested, the universe would then be running itself [21] and would

be doing so rather efficiently [12] and may be quite simple [27], yet there is no need to deny the physical carrier, which is simply the performer.

6 CONCLUDING REMARKS

I have proposed a novel qualitative notion of computation based in the sensitivity of a system to external stimuli connected to a concept of programmability. A notion I have called *nature-like computation* that provides a behavioural interpretation of computation (and of computers). This goes along current lines of technology for programming molecules and cells to compute, see for example Ref. [1]. This in some way can be seen as reprogramming a cell to do certain tasks that wasn't supposed to do from their natural course. This is what in some way we have done with digital computers too, building machines out of natural matter to make them do calculations for us. All around a single concept, that of programmability, that I have suggested can be captured by a measure of behaviour rather than syntactic or even semantic approaches, given that the former requires descriptions of inner workings, even though we may not even fully understand the machinery of a cell, and the latter requires an interpretation of computation. The behavioural approach, however, is agnostic in most of these accounts, and it only cares about the qualitative behaviour of a system to transfer information by being stimulated. The concept also helps to give sense to current uses of computation in the context of natural phenomena, including the universe itself.

ACKNOWLEDGEMENTS

I would like to thank the organisers of the symposium *Natural/Unconventional Computing and its Philosophical Significance* for their kind invitation to talk at the AISB/IACAP World Congress 2012 – Alan Turing 2012.

REFERENCES

[1] S. Ausländer, D. Ausländer, M. Müller, M. Wieland and M. Fussenegger, Programmable single-cell mammalian biocomputers, *Nature*, 2012.

[2] Baiocchi, C. Three small universal Turing machines. In Margenstern M. and Rogozhin, Y. (eds), Machines, *Computations, and Universality (MCU),* volume 2055 of LNCS, pages 1–10, Springer, 2001.

[3] Invited talk by Blanco, J. *Interdisciplinary Workshop with Javier Blanco: Ontological, Epistemological and Methodological Aspects of Computer Science*, University of Stuttgart, Germany, July 7th 2011.

[4] Cook, M. Universality in Elementary Cellular Automata, *Complex Systems* 15: pp. 1–40, 2004.

[5] Berlekamp, E., Conway, K. and Guy R., Winning Ways for your Mathematical Plays, vol. 2, Academic Press, 1982.

[6] Invited talk by Davis, M. Universality is Ubiquitous, Invited Lecture, *History and Philosophy of Computing (HAPOC11),* Ghent, 8 November, 2011.

[7] Dershowitz, N. and Gurevich, Y. A natural axiomatization of computability and proof of Church's Thesis, *Bulletin of Symbolic Logic*, 14(3):299–350, 2008.

[8] Deutsch, D. *The Fabric of Reality: The Science of Parallel Universes and Its Implications*, Penguin, 1998.

[9] Feynman, R., *The Character of Physical Law*, Modern Library, 1994.

[10] Fredkin, E. Finite Nature, *Proceedings of the XXVIIth Rencotre de Moriond*, 1992.

[11] Gandy, R. Church's Thesis and principles for mechanisms. In Barwise, J., Keisler, H.J. and Kunen, K. (eds) *The Kleene Symposium,* North-Holland, 123–148, 1980.

[12] Schmidhuber, J. Algorithmic Theories of Everything, arXiv:quant-ph/0011122v2, 2000.

[13] Illachinski, A. *Cellular Automata: a Discrete Universe*, World Scientific Publishing Co, 2001.

[14] Perlis, A.J. Epigrams on Programming, *SIGPLAN Notices,* Vol. 17, No. 9, pages 7–13, 1982.

[15] Margenstern, M. Turing Machines with Two Letters and Two States, *Complex Systems,* (19)1, 2010.

[16] Moor, J.H. Three Myths of Computer Science, *The British Journal for the Philosophy of Science*, Vol. 29, No. 3, pp. 213- 222, 1978.

[17] Scott, D.S. Outline of a mathematical theory of computation, *Technical Monograph PRG-2, Oxford University Computing Laboratory,* England, November 1970.

[18] Searle, J.R. Is the Brain a Digital Computer, in *Philosophy in a New Century,* pp 86–106, Cambridge University Press, 2008.

[19] Sieg, W. Step by recursive step: Church's analysis of effective calculability (with a Postscript), forthcoming in Zenil, H. *A Computable Universe*, World Scientific, 2012.

[20] Sutner, K. Computational Processes, Observers and Turing Incompleteness, *Theoretical Computer Science,* Volume 412, pp. 183–190, 2011.

[21] Lloyd, S. Computational capacity of the Universe, *Physical Review Letters*, 88, 237901, 2002.

[22] Margolus, N. Physics-like Models of Computation, Physica, Vol. 10D, pp. 81–95, 1984.

[23] De Mol, L. Generating, solving and the mathematics of Homo Sapiens. Emil Posts views on computation, forthcoming in Zenil, H. (ed.), *A Computable Universe*, World Scientific, 2012.

[24] Turner, R. Specification, *Minds and Machines,* 21 (2): pp 135–152, 2011.

[25] Wheeler, J.A. Information, physics, quantum: The search for links. In Zurek, W. (ed.) *Complexity, Entropy, and the Physics of Information*, Addison-Wesley, 1990.

[26] Wolfram, S. Cellular Automata as Models of Complexity, *Nature*, 311, 419–424, 1984.

[27] Wolfram, S. *A New Kind of Science,* Wolfram Media, 2002.

[28] Zenil, H. Compression-based investigation of the behaviour of cellular automata and other systems, *Complex Systems,* (19)2, 2010.

[29] Zenil, H., Soler-Toscano F. and Joosten, J.J. Empirical Encounters With Computational Irreducibility and Unpredictability, *Minds and Machines,* vol. 21, 2011.

[30] Zenil, H. On the Dynamic Qualitative Behaviour of Universal Computation, *Complex Systems,* (20)3, 2012.

Does the Principle of Computational Equivalence overcome the objections against Computationalism?

Alberto Hernández-Espinosa[1] **and Francisco Hernández-Quiroz**[2]

Abstract. Computationalism has been variously defined as the idea that the human mind can be modelled by means of mechanisms broadly equivalent to Turing Machines. Computationalism's claims have been hotly debated and arguments against and for have drawn extensively from mathematics, cognitive sciences and philosophy, although the debate is hardly settled. On the other hand, in his 2002 book *New Kind of Science*, Stephen Wolfram advanced what he called the Principle of Computational Equivalence (PCE), whose main contention is that fairly simple systems can easily reach very complex behaviour and become as powerful as any possible system based on rules (that is, they are computationally equivalent). He also claimed that any natural (and even human) phenomenon can be explained as the interaction of very simple rules. Of course, given the universality of Turing Machine-like mechanisms, PCE could be considered simply a particular brand of computationalism, subject to the same objections as previous attempts. In this paper we analyse in depth if this view of PCE is justified or not and hence if PCE can overcome some criticisms and be a different and better model of the human mind.

1 INTRODUCTION

Computational Theory of the Mind (CTM) or computationalism is usually attributed to Alan Turing (for instance [15]). In fact Turing compared the human brain to a digital computing machine [36], but also to an analogue type machine [38], but we should point out that Turing never developed a formal theory of thought, despite his foundational work on computability. In contrast, McCulloch and Pitts [21] did talked about mental processes as computations, as Piccinini remind us [23].

As we said before, computationalism is not a single thesis, but it has been formulated differently by many people. According to Piccinini [23], computationalism claims that cognitive activity is achieved by means of computations carried out by specific components of the mind whose functioning is akin to that of a Turing Machine (TM) or an equivalent mechanism. The fact that cognition happens in the brain (and the brain is based on neural networks and not on TM) can be incorporated into computationalism by considering that neural computations are Turing-computable at least as they are actually realized in the human brain. This wider thesis would make some types of connectionism mere variations of computationalism.

Talking about computationalism, Piccinini [24] distinguishes two variations: 1)traditional or classic computationalism, which claims that thought can be reduced to computations made over linguistic structures and 2) connectionist computationalism, which claims that thought can be reduced to "computations" carried out by Neural Network Systems.

There are some other theses that frequently have been grouped together under the label of computationalism, for instance, the so called Strong Artificial Intelligence (SAI), which, according to Searle, claims that artificial intelligence can eventually reach the ability of becoming self-aware and exhibit human-like thought processes (Searle [31]).

We will not dwell on this specific variety of computationalism (if it really can be found beyond Searle's analysis) as we are interested only in the explaining power of computationalism for understanding the human mind and not in the question of whether computers can really think and we consider Piccinini's classification perfectly adequate for this purpose.

The debate between supporters of varieties of computationalism and their detractors has raged for decades and both sides have drawn arguments from mathematics, cognitive science and philosophy. The point is hardly settled and we do not intend to review it here even superficially. New arguments and theories keep appearing which can (or cannot) be considered variations of computationalism and claim to deal better with objections against computational explanations of the mind. Our purpose in this paper is to analyse one of this theories, namely, Stephen Wolfram's Principle of Computational Equivalence (PCE), introduced as one of the key elements of his extremely ambitious New Kind of Science program. In his book of the same name, Wolfram contends that PCE can explain complexity of any natural or artificial phenomenon, including of course the complexity of human mind.

The outline of the paper is as follows: in the second section we review some arguments against computationalism. In the third section, we summarize what we consider some of the essential claims of Wolfram's PCE as a tool for explaining the complexity of the human mind. In the fourth we ponder the ability of PCE for dealing against the cons of computationalism presented in the second section, while at the same time evaluating if PCE is or not just plain computationalism under a new disguise (although we do not offer a definite answer yet). In the final section, we point out to the challenges that PCE should deal with if it has any hope of offering a better alternative to past theories.

[1] Posgraduate Program in Philosophy of Science, UNAM (Universidad Nacional Autonoma de Mexico), MX, Email: albertohernandezespinosa@gmail.com

[2] Dept. of Mathematics, Science Faculty, UNAM (Universidad Nacional Autonoma de Mexico), MX, Email: fhq@ciencias.unam.mx.

2 FOUR TYPES OF ARGUMENTS AGAINST COMPUTACIONALISM

Cordeshi [4], Dreyfus [9, 10] and Horst [17] have brought forward diverse arguments against computationalism. We have classified them in four types for convenience.

Computationalism contends that is the only scientific explanation in offer. Their supporters argue that computational explanations of cognitive abilities like language and learning are the only viable approach to the mind. Examples of this view can be found in Fodor [13], Pinker [25] and Winograd [39]. Even if they take for granted that the mind "resides" in the brain and the brain is a gigantic neural network, they also claim that electrical signals in neural networks codify symbols and representations which are manipulated according to logical rules [30]. One consequence of this view is that the mind deals basically with representational systems [17]. A first and clear line of attack against computationalism is to challenge this contention. As Horst has pointed out [19], in the search for alternatives, philosophers and cognitive scientists are reconsidering if models like neural networks can and should be based on rules and representations or if they work in a radically different way.

On the other hand, Dreyfus [9, 10] and even Winograd and Flores [39] have argued that a significant part of what we call thought and behaviour cannot be reduced to explicit rules and therefore cannot be formalized (and translated into a computer program). In other words, a sizeable portion of mental phenomena are beyond the reach of techniques dearest to computationalists.

A third line of criticism rejects the use of symbols as the foundation of the semantics of thoughts. Symbolic semantics imply intentionality in thought either through causality [16, 27, 28] or concepts [18]. But trying to explain intentionality by symbols is a vicious circle. Searle [29] and Horst [18] go further and state that computer "representations" are not even symbolic on their own right as its symbolic nature rests on the intentions and conventions held by their human users.

Supporters of externalist theories of meaning have raised a fourth set of criticisms. Many computationalists were fond of what can be called "methodological solipsism" [12] or individualism: the view that mental states' characterization is insensitive to and independent from any external features of the cognitive subject, as the underlying computational processes only have access to mental representations. But at the same time, computationalism would have this characterization reflecting semantic properties. This is clearly difficult to reconcile with an externalist stand on meaning, which would require that the meaning of terms be at least partially determined by factors external to the cognitive subject, for instance, its physical [24] and linguistic [1, 2] environment. Of course, the argument can be turned around to reject externalism as Fodor did [11].

3 NKS AND THE PRINCIPLE OF COMPUTATIONAL EQUIVALENCE (PCE)

Stephen Wolfram wrote his book *A New Kind of Science* (NKS) [40] after twenty years of experimentation with Cellular Automata (CA) as tools for solving problems in a very wide range of domains. One of the main guidance of his proposal is the Principle of Computational Equivalence (PCE), which can be summarized by the two following theses:

1. All processes, whether they are produced by human effort or occur spontaneously in nature, can be viewed as computations.
2. In computational terms there is a fundamental equivalence between many different kinds of processes. In particular, almost all processes that are not obviously simple can be viewed as computation of equivalent sophistication. [40]

In very general terms, Wolfram contends that PCE means that there is a maximal ("universal") level of complexity in computations and this level is easily attainable by most non-trivial systems (even artificial ones). Natural systems can in principle have the same computational power as computers and vice versa. Wolfram claims that, provided a proper translation for inputs and outputs of different systems, all of them are computationally equivalent.[3]

Wolfram states that his NKS has three basic advantages over classical science:

1. An alternative view of randomness: over time, simple rules can produce very complex behaviour which becomes almost impossible to predict. Randomness is then just unpredictability arising from lack of information about deterministic phenomena. But this type of "randomness" can be approximated by means of programs based on very simple rules.
2. Scientific insight should be guided by the search of these very simple rules in all natural and human phenomena. Of course, this idea goes counter the "prejudice" that computing simulations of natural phenomena should be based in very complex software. The key, according to Wolfram, is the opposite: look for simple rules.
3. Given that all systems are based on simple rules, individual sciences can proceed to analyse their disparate subjects by means of a uniform methodology which can help to extract more general and abstract explanations.

Stephen Wolfram states explicitly that the complexity of the human mind is also covered by PCE. For instance, he claims that perception can be reduced to a process of pattern recognition and information processing [40]. At first sight, PCE seems to be just another version of classical computationalism. But it may not be so simple. For instance: does PCE imply representationalism? Other similar questions can be easily asked and their answers are not straightforward, which makes us think worthwhile to consider in depth if Wolfram's proposal can really offer a valuable alternative to classical computationalism.

4 NKS VS OBJECTIONS AGAINST COMPUTATIONALISM

Following Dodig-Crnkovic's analysis of what she calls info-computationalism (the strong thesis that the universe can be

[3] Sutner claims that Wolfram's view can be taken to mean that there are really only two levels of complexity in natural phenomena: a lower one of systems whose behaviour is decidable and the higher one of systems whose behaviour reaches universal complexity in computational terms [35]. This thesis has been called a 0/1 law of computational degrees.

better understood as a series of computational processes operating on informational structures) [5, 6, 7], we may be inclined to regard PCE as a variety of info-computationalism. Nonetheless, there are at least two reasons why Wolfram's proposal may be considered a different and probably better brew of computationalism which may be able to avoid some criticisms directed against other traditional computationalist views: 1) if he is right (and this a big "if") that there is an upper limit in complexity for all systems and this limit can be reached by simple rules, then of course computer programs can simulate any degree of complexity; 2) again, if his main thesis is right, the complexity of the mind also falls in the scope of what can be explained by computations based on simple rules.

While it is far from clear that all systems in nature have a complexity limit within the reach of the computable, computable universality is reachable by means of the simple rules advocated by Wolfram [33, 3]. The general question of a universal limit is still open and seems bound to remain so for the foreseeable future.

On the other hand, even if Wolfram were right about the existence of an upper limit in complexity, he offers no practical clues for the solution of the many problems any theory of mind (let alone a computational one) should face. His optimism becomes evident when he regards a possible explanation of free will as computationally complex decision procedures whose inner details are hidden from consciousness [40].

NKS and the PCE are then just a (sketch of a) proposal for a research program and before embracing it any prospective theoretician of mind should at least make a quick assessment of its potential:

1. A first obvious question is if we are not dealing with a mere variety of computationalism.
2. A second and more interesting one is if PCE is not simple computationalism (or even despite *being* computationalism), how it can overcome the objections faced by psychological and connectionist models [13].
3. Next it is to be seen if PCE can answer the objection that human thought and behaviour cannot be reduced to explicit rules and therefore cannot be formalized or reduced to computer programs [9, 10, and 39].
4. PCE should also offer a theory of the meaning of thought without the troubles faced by computationalism's symbolic semantics [28, 16, and 27].
5. Finally, PCE should present an alternative explanation of how mental states can be characterized independently of features external to the cognitive subject [1, 2].

Many other issues could be raised [4, 24], but we consider these some of the most relevant because they touch the core of the theory and we will dwell on them in the next section

5 PROBLEMS TO SOLVE

What are the chances of PEC dealing rightly with the previous questions? It is not our intention to give a definitive answer, but just to offer a very initial assessment and to outline how a NKS practitioner should carry on.

To begin with, the charge of being just computationalism under a different guise. Mathematically speaking, the simple rules on which NKS is based are computationally equivalent to Turing Machines and other Turing-complete models. Claiming that any system (natural or artificial) is of equivalent complexity is highly reminiscent of (a strong form of) Church-Turing's thesis, on its turn one of the pillars of computationalism. Wolfram himself seems to support this last: "But it was not until the 1980s –perhaps particularly following some of my work – that it began to be more widely realized that Church's Thesis should best be considered a statement about nature and about the kinds of computation that can be done in our universe. The validity of Church's Thesis has long been taken more or less for granted by computer scientist, but among physicists there are still nagging doubts, mostly revolving around the perfect continua assumed in space and quantum mechanism in the traditional formalism of theoretical physics" [40]. Wolfram calls Turing's and other scientists' attempts "close approaches", acknowledging their similarity, but he also claims to have a distinctive proposal which is also based on "experimentation" on computers. Of course, these short and sometimes puzzling comments do not settle the point as (a sort of) mathematical equivalence between Church's thesis and PCE does not imply that PCE has to assume all the baggage of classical computationalism (which in turn is not a consequence of Church's thesis).

Regarding the second question, PCE should be able to attain at least the same degree of success as connectionism, an important rival of classical computationalism. According to some researchers [16] connectionism has been able to explain some intellectual abilities without resorting to syntactical representations and manipulations (let us put aside the issue of Artificial Neural Networks as they exist being mathematically equivalent to Turing Machines), performing better than actual or potential systems based on techniques dear to computationalists. Can PCE equal these supposed achievements? Again, for the time being Wolfram's NKS can only provide more optimism: "So on the basis of traditional intuition; one might then assume that the way to solve this problem must be to use systems with more complicated underlying rules, perhaps more closely based on details of human psychology or neurophysiology. But from discoveries in this book we know that this is not the case, and that in fact very simple rules are quite sufficient to produce highly complex behaviour" [40].

Searle [29, 31] and Horst [18] have provided a powerful argument against the idea that thought can be reduced to the application of simple rules in the style of a computer program, as meaning cannot be derived from rules for manipulating symbols (so covering the core of questions 4 and 5): "The problem of semantics is: How these sentences in the head get their meaning? But that question can be discussed independently of the question: How does the brain work in processing these sentences?" [29]. About this last issue Wolfram says: "One might have imagined that human thinking must involve fundamentally special processes, utterly different from all other processes that we have discussed [here Wolfram talks about thinking and perception as processes]. But just as it has become clear over the past few centuries that the basic physical constituents of human beings are not particularly special, so also –especially after the discoveries in this book (NKS) – I am quite certain that in the end there will turn out to be nothing particularly special about the basic processes that are involved in human thinking. And indeed, my strong suspicion is that despite the apparent sophistication of human thinking most of the most important processes that underlie it are very simple" [40]. To be fair (and therefore not so

pessimistic), Wolfram's phrasing of the problem does not imply that the solution should be attached to rules *for manipulating symbols*.

Finally, there is the issue of defining mental states (which are internal representations according to computationalism) and their complex relation with features external to the cognitive subject [23]. Can Wolfram's idea of intelligence being based at least partially on pattern recognition point to a different definition about what a mental state is and how it relates to the external world (the ultimate source from which the pattern is recognized)? We consider that, right now, this idea is too vague to give rise to any serious attempt to formulate the problem of mental states, let alone to lead to its solution.

To conclude: PCE hopes for being a better alternative than classical computationalism are dependent on many "if", namely: if Wolfram is right that all natural and artificial phenomena are under the scope of the kind of simple computational rules he advocates, if these rules can lead to practical ways of explaining what previous models have been unable to explain, if complex behaviour such as meaning and mental states (and their relation with the external world) can be accounted for by the same rules, then NKS can offer a way out of computationalism troubles. On a more positive note, we want to stress an implicit conclusion of our previous analysis: it is not obvious that PCE should fail where classical computationalism has already failed.

But optimism cannot be the only foundation for a scientific account of the human mind. More philosophical and empirical research is needed to see if optimism can be turned into results or, at least, concrete lines of research.

REFERENCES

[1] Burge, T. Individualism and the mental. In P. French, T. Euhling, y H. Wettstein, Eds., *Studies in Epistemology*, Midwest Studies in Philosophy, vol. 4. Minneapolis: University of Minnesota Press (1979).

[2] Burge, T. Individualism and psychology. *Philosophical Review* 95(1): 3–45 (1986).

[3] Cook, M. "Universality in Elementary Cellular Automata". Complex Systems 15, 1-40. (2004).

[4] Cordeschi Roberto. Computacionalism under attack. Cartographies of the Mind, Chapter 3, p. 37-49 (2007).

[5] Dodig Crnkovic, G., Mueller, V. A dialogue concerning two world systems: info-computational *vs.* mechanistic. In *Information and Computation*; Dodig Crnkovic, G., Burgin, M., Eds.; World Scientific Publishing Co., Inc.: Singapore; pp. 149–184, (2009).

[6] Dodig Crnkovic, G. *Biological Information and Natural Computation*. In *Thinking Machines and the Philosophy of Computer Science: Concepts and Principles*; Vallverdú, J., Ed.; Information Science Reference (IGI Global): Hershey, PA, USA (2010).

[7] Dodig Crnkovik G. "Dynamics of Information as Natural Computation", 2(3), 460-477; doi:10.3390/info2030460, Information (2011).

[8] Dodig Crnkovic G.: Physical Computation as Dynamics of Form that Glues Everything Together. Information 3 (2): 204-218 (2012)

[9] Dreyfus, H. *What Computers Can't Do*. Nueva York: Harper and Row (1972).

[10] Dreyfus, H. What Computers Still Can't Do. Cambridge, MA: MIT Press (1992).

[11] Fodor, J. Metodological silopsism considered as a research strategy in cognitive science. Behavioral and brain sciences 3: 63-73

[12] Fodor, J. *Representations*. Cambridge, MA: MIT Press (1981).

[13] Fodor, J. *A Theory of Content and Other Essays*. Cambridge, MA: MIT Press (1990).

[14] Fodor, J. The Elm and the Expert. Cambridge, MA: MIT Press. *[El olmo y el experto, Barcelona, Paidós*, 1997] (1993).

[15] Fodor, J. A. Concepts, Clarendon Press, Oxford (1998).

[16] Garrido A. "Connectionism vs. Computationalism Theory of Mind", BRAIN (Broad Research in Artificial Intelligence and Neuroscience) Vol. 1, Issue 1, January 2010, ISSN 2067-3957

[17] Horst, S. Mind Computationalism Theory. Berkeley y Los Ángeles: University of California Press (1986).

[18] Horst, S. *Symbols, Computation and Intentionality: A Critique of the Computational Theory of Mind*. Berkeley y Los Ángeles: University of California Press (1996).

[19] Horst, S., "The Computational Theory of Mind", The Stanford Encyclopedia of Philosophy (Spring 2011 Edition), Edward N. Zalta (ed.). url http://plato.stanford.edu/archives/spr2011/entries/computational-mind/, 30/03/2011

[20] Kelemen Josef and Alica Kelemenova, The New Computationalism: a Lesson from Embodied Agents. Silesram University. Opava, Slovakia. Institute of Computer Science Press (2008).

[21] McCulloch, W. S. and W. H. Pitts: 'A Logical Calculus of the Ideas Immanent in Nervous Activity', *Bulletin of Mathematical Biophysics* 7, 115–133. (1943).

[22] Piccinini G. The first Computational theory of mind and brain: a close look at McCulloch and Pitts´s "Logical calculus of ideas immanent in nervous activity", Synthese 141: 175-215 (2004).

[23] Piccinini G. "Computationalism, the Church-Turing thesis, and the Church-Turing falacy", Synthese, 154.1. pp.97-120 (2007).

[24] Piccinini G, "Computatonalism in the Philosophy of Mind" (2009).

[25] Pinker, S. "The Language Instinct, the New Science of Language and Mind", London: Penguin (1995).

[26] Pinker, S. "How the Mind Works", London: Penguin (1997).

[27] Putnam, H. The Meaning of "Meaning." En K. Gunderson, Ed., *Language, Mind and Knowledge*. Minnesota Studies in the Philosophy of Science, vol. 7. Minneapolis: University of Minnesota Press (1975).

[28] Sayre, K. Intentionality and information processing: An alternative model for cognitive science. *Behavioral and Brain Sciences* 9(1): 121–138 (1986).

[29] Searle, J. Minds, brains and programs. Behavioral and Brain Sciences 3: 417–424 (1980).

[30] Searle, J. Presidential address. *Proceedings of the American Philosophical Association* (1990).

[31] Searle, J. "The Chinese room" (1999).

[32] Searle, J. "Is the Braid to digital computer?" Proceedings and Addresses of American Philosophical Association 64 (Novenber): 21-37

[33] Smith, A. Universality of Wolfram's 2,3 Turing Machine (2007)

[34]Sutner K.. Cellular automata and intermediate degrees. Theoretical Computer Science, 296:365–375, 2003.

[35] Sutner K. "Universality and Cellular Automata", MCU, 2004: 50-59

[36] Turing, A. M. 'Lecture to the London Mathematical Society on 20 February 1947', in D. Ince (ed.), Mechanical Intelligence. North-Holland, Amsterdam, pp. 87–105 (1947).

[37] Turing, A. M. 'Intelligent Machinery', in D. Ince (ed.) Mechanical Intelligence. North-Holland, Amsterdam, pp. 87–106 (1948).

[38] Turing, A.M. "Computing machinery and intelligence. Mind", 59, 433-460 (1950)

[39] Winograd, T., y F. Flores. Understanding Computers and Cognition. Norwood, NJ: Ablex (1986).

[40] Wolfram, S. "The New Kind of Science", ISBN 1-57955-008-8 (2002).

Intelligence and reference. Formal ontology of the natural computation

Gianfranco Basti[1]

Abstract. In a seminal work published in 1952, "The chemical basis of morphogenesis" — considered as the true start point of the modern theoretical biology —, A. M. Turing established the core of what today we call "natural computation" in biological systems, intended as self-organizing dynamic systems. In this contribution we show that the "intentionality", i.e., the "relation-to-object" characterizing biological morphogenesis and cognitive intelligence, as far as it is formalized in the appropriate ontological interpretation of the modal calculus (formal ontology), can suggest a solution of the reference problem that formal semantics is in principle unable to offer, because of Gödel and Tarski theorems. Such a solution, that is halfway between the "descriptive" (Frege) and the "causal" (Kripke) theory of reference, can be implemented only in a particular class of self-organizing dynamic systems, i.e., the dissipative chaotic systems characterizing the "semantic information processing" in biological and neural systems.

1 INTRODUCTION

1.2 Natural computation and algorithmic computation

Today *natural computation* (NC) is considered as an alternative paradigm to the *algorithmic computation* (AC) paradigm in natural and computer sciences, being the paternity of only the latter one generally ascribed to Alan Mathison Turing (1912-1954) seminal work. On the contrary, after the publication of his famous seminal work on algorithmic computation in 1936 [1] by the notions of Turing Machine (TM) and Universal Turing Machine (UTM), Turing worked for widening the notion of "computation" in the direction of what today we define as "natural computation".

Before all, he defined the notion of Oracle-machine(s) – i.e., a TM enriched with the output of operations not computable by a TM, endowing the TM with the primitives of its computable functions – and of their transfinite hierarchy, in his doctoral work at Princeton, under the Alonso Church supervision, published in 1939 [2].

Afterward, in 1947 in a lecture given at the *London Mathematical Society* [3], and hence in an unpublished communication for the *National Physical Laboratory* in 1948 [4], he sketched the idea of computational architectures made by undefined interacting elements, that can be suitably trained, so to anticipate the so-called Artificial Neural Networks (ANN) computational architectures.

Finally, in a much more known contribution on a new mathematical theory of morphogenesis, published in 1952 [5], Turing was the first who studied a model of pattern formation via non-linear equations, in the specific case of chemical reaction-diffusion equations simulated by a computer.

This pioneering work on non-linear systems, and their simulation via computers, is, indeed, among all the pioneering works of Turing, the most strictly related with the new paradigm of NC, because of its wide field of application in practically every realm of mathematical and natural sciences, from cosmology and fundamental physics, to thermodynamics, chemistry, genetics, epigenetics, biology, and neurosciences; but also in human sciences, from cognitive and social sciences, to ecology, to economical sciences, to linguistics, …, and wherever a mathematical modeling of empirical data makes sense[2].

In a recent paper devoted to illustrate the new paradigm of NC in relationship with the old paradigm of AC [6], G. Dodig-Crnkovic emphasizes the main differences between the two paradigms that can be synthesized according to the following, main dichotomies:

1. *Open, interactive agent-based computational systems* (NC) [3] vs. *closed, stand-alone computational systems* (AC);
2. *Computation as information processing and simulative modeling* (NC) vs. *computation as formal (mechanical) symbol manipulation* (AC);
3. *Adequacy of the computational response via self-organization as the main issue* (NC) vs. *halting problem (and its many, equivalent problems) as the main issue in computability theory* (AC);

Of course, such dichotomies must be intended, in perspective, as oppositions between complementary and not mutually exclusive characters of computation models. However, as Dodig-Crnkovic emphasizes, such a complementarity might emerge only when a foundational theory of NC will be sufficiently developed, overall as to semantic and the logic of NC. The present contribution is devoted precisely to this aim, even though it is necessary to add to the previous list other two essential dichotomic characters of NC, emphasized by Dodig-Crnkovic in other papers, overall the more recent one published on the *Information* journal [7]:

4. *Intentional, object-directed, pre-symbolic computation, based on chaotic dynamics in neural computation* (NC) vs. *representational, solipsistic, symbolic computation, based on linear dynamics typical of early AI approach to cognitive neuroscience* (AC).
5. *Dual ontology based on the energy-information distinction of natural (physical, biological and neural) systems* (NC) vs. *monistic ontology based on the energy-information equivalence in all natural systems* (AC).

[1] Pontifical Lateran University, Rome. basti@pul.it

[2] On this regard, apart from the classical book of J. Gribbin about the history of the non-linear science in the second half of the XX century [84], see the special issue of *Nature* of for the Turing centenary with, among the others, the celebrative contributions of G. Dyson [59], J. Reiniz [60], B. Cooper [61].

[3] So, she synthesizes this important fundamental character of NC approach: «Agent Based Models are the most important development in this direction, where a complex dynamical system is represented by interacting, in general adaptive, agents. Examples of such systems are in physics: turbulence, percolation, sand pile, weather; in biology: cells organs (including brain), organisms, populations, ecosystems; and in the social sphere: language, organizations, and markets».

1.3 Relevance of the reference problem in NC

In this paper, we want to suggest how a foundational approach to NC, overall as to its logical and semantic components cannot disregard the essential point of how to *integrate in one only formalism* the *physical* ("natural") realm with the *logical-mathematical* ("computation") one, as well as their relationship. That is, the passage from the realm of the *causal* necessity ("natural") of the physical processes, to the realm of the *logical* necessity ("computational"), eventually representing them either in a sub-symbolic, or in a symbolic form. This foundational task can be performed, by the newborn discipline of *theoretical formal ontology* [8, 9, 10, 11, 12], as distinguished from *formal ontology engineering* – an applicative discipline, well established and diffused in the realm of computational linguistics and semantic databases[4].

Particularly, the distinction between *formal logic* and *formal ontology* is precious for defining and solving foundational misunderstanding about the notion of *reference* that the NC approach had the merit of emphasizing, making aware of it the largest part of the computer science community – and also the rest, we hope, of the scientific community, as far as NC is spreading all over the entire realm of the natural sciences.

In fact, as A. Tarski rightly emphasized since his pioneering work on formal semantics [13], not only the *meaning* but also the *reference* in *logic* has nothing to do with the *real, physical world*. To use the classic Tarski's example, the semantic reference of the true statement "the snow is white" is not the whiteness of the crystalized water, but at last an empirical set of data to which the statement is referring, eventually taken as a *primitive* in a given formal language. In other terms *logic* is always *representational*, it concerns relations among tokens, either at the *symbolic* or *sub-symbolic* level. It has always and only to do with representations, not with real things. This is well emphasized, also, by R. Carnap's principle of the *methodological solipsism* in formal semantics [14, p. 423], that both H. Putnam [15] and J. Fodor [16] rightly extended also to the *representationalism* of cognitive science, as far as it is based in the so-called *functionalist* approach of the classic, symbolic AI, and hence of the classic AC paradigm. Finally, this is also the deep reason of what Quine defines as the "impenetrability of reference" beyond the network of equivalent statements meaning the same referential object in different languages [17].

To sum up, what satisfies (makes true) a predicate in logic are not real objects, but the terms denoting them. A class (or a set), intended as the collection of elements satisfying a given predicate (function) denoting the class (or enumerating completely the set) is an, abstract, logical entity, not a collection of real things – a "natural kind".

Now in AC, any formal theory of reference and truth is faced with the Gödelian limits making impossible a recursive procedure of satisfaction in a semantically closed formal language. What we emphasized also elsewhere [18, 19, 20], as the core of the reference problem is that such a recursive procedure for being complete would imply the solution of the *coding* problem through a diagonalization procedure; that is, the solution of the so–called "Gödel numbering" problem. In computational terms, the impossibility of solving the coding problem through a diagonalization procedure means that no TM can constitute by itself the "basic symbols", the primitives, of its own computations. For this reason Tarski rightly stated that, at the level of the propositional calculus, the semantic theory of truth has nothing to say about the conditions under which a given simple ("atomic" in L. Wittengstein's terms) proposition can be asserted. And for this very same reason, in his fundamental paper about *The meaning of "meaning"* [15], Putnam stated that no ultimate solution exists in logic both of the problem of *reference* and, at the level of linguistic analysis, of the problem of *naming.*

In this sense, Putnam stated, we would have to consider ultimately names as *rigid designators* "one - to - one" of objects in S. Kripke's sense [21]. But no room exists, also in Kripke's theory of partial reference – and hence using Kleene's genious solution of partial recursive predicates for dealing with the problem of enumeration of partial functions, that Gödelian notion of general recursion cannot approach in principle (see [22, p. 313 and 327f.]) – for defining the notion of *rigid designation* in terms of a purely logical relation, since any logical relation only holds among tokens and not between tokens and objects, as Tarski reminded us. Hence a formal language has always to suppose the existence of names (or numbers) as rigid designators and cannot give them a foundation.

To explain by an example the destructive consequences of this point for a functionalist theory of mind, Putnam suggested a sort of third version of the famous "room – metaphor" [23, p. 116ff.], after the original "Turing test" version of this metaphor, and J. Searle's "Chinese – room" version of it. Effectively, Putnam proposed by his metaphor a further test that a TM cannot solve and that has much deeper implications than the counterexample to the Turing test proposed by Searle. For instance, Putnam said, if we ask "how many objects are in this room?", the answer supposes a previous decision about which are to be considered the "real" objects to be enumerated — i.e., *rigidly designated by numerical units*. So, one could answer that the objects in that room are only three (a desk, a chair and a lamp over the desk). However, by changing the enumeration axiom, another one could answer that, for instance, the objects are many billions, because we have to consider also the molecules of which the former objects are constituted.

Out of metaphor, any computational procedure of a TM (and any AC procedure at all, if we accept Church's thesis) supposes the determination of the basic symbols on which the computations have to be carried on – the partial domain on which the recursive computation has to be carried on. Hence, from the semantic standpoint, any computational procedure supposes that such numbers are *encoding* (i.e., unambiguously naming as rigid designators) as many "real objects" of the computation domain. In short, owing to the coding problem, the determination of the *basic symbols* (numbers) on which the computation is carried on, *cannot have any computational solution* in the AC paradigm.

The *closed, stand-alone* character of AC models depends thus on the purely *syntactic* and *semantic* level in which the logical approach can develop its analysis of the reference problem, hence at a necessarily *representational/symbolic* level. Precisely for this systematic impossibility of the logical theory of reference of justifying logical truth as adequacy to outer reality H. Putnam abandoned the functionalist approach to cognitive science he himself contributed to define in 60's of last century, for an *intentional*, non-representational theory of a cognitive act, based on a *causal theory of reference* as anticipated in his early works of 1973 [24] and of 1975 [15], even though in a different sense as to other representatives of this theory like, for instance, K. S. Donnellan [25] and S. Kripke himself [21]. Putnam indeed rightly vindicated that a causal theory of reference supposes that at least at the beginning of the social chain of "tradition" of a given denotation there must be an

[4] A typical representative of researcher in both fields of formal ontology is Barry Smith (see, for instance, [40, 41, 42]).

effective *causal relation* from the denoted thing to (the cognitive agent producing) the denoting name – and, in the limit, in this causal sense must be intended also the act of perception Kripke vindicated as sufficient for the dubbing of a given object. To synthesize this position, even though Putnam never spoke in these terms, what is necessary is a "causal", "finitistic" theory of coding in which the real thing causally and progressively determines the partial domain of the descriptive function recursively denoting it.

It is thus evident the necessity of *formal ontology* for formalizing such an approach to the meaning/reference problem in the NC paradigm. That is, it is evident the necessity of *a formal calculus of relations* able to include in the same, coherent, formal framework both "causal" and "logical" relations, as well as the "pragmatic" (real, causal relations with the cognition/communication agents), and not only the "syntactic" (logical relations among terms) "semantic" (logical relations among symbols) components of meaningful actions/computations/cognitions.

2 FROM FORMAL LOGIC TO FORMAL ONTOLOGY

2.1 Extensional *vs.* intensional logic

The *modal logic* with all its *intensional* interpretations are what is today defined as *philosophical logic* [26], as far as it is distinguished from the *mathematical logic*, the logic based on the extensional calculus, and the extensional meaning, truth, and identity[5]. What is new is that also the intensional logics can be *formalized* (i.e., translated into a proper symbolic language, and axiomatised), against some rooted prejudices among "continental" philosophers, who abhor the symbolic hieroglyphics of the "analytic" ones. I.e., there exists an *intensional logical calculus*, just like there exists an extensional one, and this explains why both mathematical and philosophical logic are today often quoted together within the realm of *computer science*. This means that classical semantic and even the intentional tasks can be simulated artificially. This is the basis of the incoming "Web3 revolution", i.e., the advent of the *semantic web*. Hence, the "thought experiment" of Searle's "Chinese Room" is becoming a reality, as it happens often in the history of science.

Anyway, to conclude this part, the main intensional logics with which we are concerned in the present paper are:

1. *Alethic logics*: they are the descriptive logics of "being/not being" in which the modal operators have the basic meaning of "necessity/possibility" in two main senses:
   a. *Logical necessity*: the necessity of lawfulness, like in deductive reasoning
   b. *Ontic necessity*: the necessity of causality, that, on its turn, can be of two types:
      - *Physical causality*: for statements which are true (i.e., which are referring to beings existing) only in some possible worlds. For instance, biological statements cannot be true in states, or parts, or ages of the universe in which, because of the too high temperatures only quantum systems can exist).
      - *Metaphysical causality*: for statements which are true of all beings in all possible worlds, because they refer to properties or features of all beings such beings.
2. *The deontic logics*: concerned with what "should be or not should be", where the modal operators have the basic meaning of "obligation/permission" in two main senses: *moral* and *legal obligations*.
3. *The epistemic logic*: concerned with what is "science or opinion", where the modal operators have the basic meaning of "certainty/uncertainty". It is evident that all the "belief" logic pertains to the epistemic logic, as we see below.

2.2. Interpretations of modal logic

For our aims, it is sufficient here to recall that formal modal calculus is an extension of classical propositional, predicate and hence relation calculus with the inclusion of some further axioms. Here, we want to recall only some of them — the axioms **N, D, T, 4** and **5** —, useful for us:

N: $<(\mathbf{X}\rightarrow\alpha) \Rightarrow (\Box\mathbf{X}\rightarrow\Box\alpha)>$, where **X** is a set of formulas (language), $\Box$ is the necessity operator, and α is a meta-variable of the propositional calculus, standing for whichever propositional variable *p* of the object-language. **N** is the fundamental *necessitation rule* supposed in any normal modal calculus

D: $<\Box\alpha\rightarrow\Diamond\alpha>$, where $\Diamond$ is the possibility operator defined as $\neg\Box\neg\ \alpha$. **D** is typical, for instance, of the *deontic* logics, where nobody can be obliged to what is impossible to do.

T: $<\Box\alpha\rightarrow\alpha>$. This is typical, for instance, of all the *alethic* logics, to express either the *logic* necessity (determination by law) or the *ontic* necessity (determination by cause).

4: $<\Box\alpha\rightarrow\Box\Box\alpha>$. This is typical, for instance, of all the "unification theories" in science where any "emergent law" supposes, as necessary condition, an even more fundamental law.

5: $<\Diamond\alpha\rightarrow\Box\Diamond\alpha>$. This is typical, for instance, of the logic of metaphysics, where it is the "nature" of the objects that determines necessarily what it can or cannot do.

By combining in a consistent way several modal axioms, it is possible to obtain several *modal systems* which constitute as many syntactical structures available for different intensional interpretations. So, given that **K** is the fundamental modal systems, given by the ordinary propositional calculus **k** plus the necessitation axiom **N,** some interesting modal systems are for our aims are: **KT4** (**S4,** in early Lewis' notation), typical of the physical ontology; **KT45 (S5,** in early Lewis' notation), typical of the metaphysical ontology; **KD45** (**Secondary S5),** with application in deontic logic, but also in epistemic logic, in ontology, as w and hence in NC as we see.

Generally, in the *alethic* (either logical or ontological) interpretations of modal structures the necessity operator $\Box p$ is interpreted as "*p* is true in all possible world", while the possibility operator $\Diamond p$ is interpreted as "*p* is true in some possible world". In any case, the so called *reflexivity principle* for the necessity operator holds in terms of axiom **T**, i.e, $\Box p \rightarrow p$. In fact, if *p* is true in *all* possible

[5] What generally characterizes intensional logic(s) as to the extensional one(s) is that neither the *extensionality axiom* – reducing class identity to class equivalence, i.e., $\mathbf{A} \leftrightarrow \mathbf{B} \Rightarrow \mathbf{A} = \mathbf{B}$ - nor the *existential generalization axiom* – $Pa \Rightarrow \exists xPx$, where *P* is a generic predicate, *a* is an individual constant, *x* is an individual variable – of the extensional predicate calculus hold in intensional logic(s). Consequently, also the Fegean notion of *extensional truth* based on the truth tables holds in the intensional predicate and propositional calculus. Of course, all the "first person" (both singular, in the case of individuals, and plural, in the case of groups), i.e., the *belief* or *intentional* (with *t*) statements, belong to the intensional logic, as Searle, from within a solid tradition in analytic philosophy [45, 46, 47], rightly emphasized [39, 38]. For a formal, deep characterization of intensional logics as to the extensional ones, from one side, and as to intentionality, from the other side, see [48].

worlds, it is true also in the *actual* world (E.g., "if it is necessary that this heavy body falls (because of Galilei's law), then this body really falls").

This is not true in *deontic* contexts. In fact, "if it is obligatory that all the Italians pay taxes, does not follow that all Italians really pay taxes", i.e., $\mathbf{O}p \not\rightarrow p$ **,** where **O** is the necessity operator in deontic context. In fact, the obligation operator **O***p* must be interpreted as "*p* is true in all *ideal* worlds" different from the actual one, otherwise **O**=□, i.e., we are in the realm of metaphysical determinism where freedom is an illusion and ethics too. The reflexivity principle in deontic contexts, able to make obligations really effective in the actual world, must be thus interpreted in terms of an *optimality operator* $\mathbf{O_p}$ for *intentional agents*, i.e,

$$(\mathbf{O}p \rightarrow p) \Leftrightarrow ((\mathbf{O_p}(x,p) \wedge c_a \wedge c_{ni}) \rightarrow p)$$

Where *x* is an intentional agent, c_a is an acceptance condition and c_{ni} is a non-impediment condition.

In similar terms, in *epistemic* contexts, where we are in the realm of representations of the real world. The interpretations of the two modal epistemic operators **B**(*x*,*p*), "*x* believes that *p*", and **S**(*x*,*p*), "*x* knows that *p*" are the following: **B**(*x*,*p*) is true iff *p* is true in the realm of representations believed by *x*. **S**(*x*,*p*) is true iff *p* is true for all the *founded* representations believed by *x*. Hence the relation between the two operators is the following:

$$\mathbf{S}(x,p) \Leftrightarrow (\mathbf{B}(x,p) \wedge \mathbf{F}) \qquad (1)$$

Where **F** is a *foundation relation,* outside the range of **B,** and hence outside the range of *x* consciousness, otherwise we should not be dealing with "knowing" but only with a "believing of knowing". I.e., we should be within the realm of solipsism and/or of metaphysical nihilism, systematically reducing "science" or "well founded knowledge" to "believing". So, for instance, in the context of a *logicist* ontology, such a **F** is interpreted as a supposed actually infinite capability of human mind of attaining the logical truth [27]. We will offer, on the contrary, a different *finitistic* interpretation of **F** within NC **.** Anyway, as to the reflexivity principle in epistemic context,

$$\mathbf{B}(x,p) \not\rightarrow p$$

In fact, believing that a given representation of the actual world, expressed in the proposition *p*, is true, does not mean that it is *effectively* true, if it is not well *founded*. Of course, such a condition **F** — that hence has to be an *onto*-logical condition — is by definition satisfied by the operator **S**, the operator of sound beliefs, so that the reflexivity principle for epistemic context is given by:

$$\mathbf{S}(x,p) \rightarrow p \qquad (2)$$

2.3 Kripke's relational semantics

Kripke relational semantic is an evolution of Tarski formal semantics, with two specific characters: 1) it is related to an *intuitionistic logic* (i.e., it considers as non-equivalent excluded middle and contradiction principle, so to admit coherent theories violating the first one), and hence 2) it is compatible with the *necessarily incomplete character* of the formalized theories (i.e., with Gödel theorems outcome), and with the *evolutionary character* of natural laws not only in biology but also in cosmology. In other terms, while in Tarski classical formal semantics, the truth of formulas is concerned with the state of affairs of *one only actual world*, in Kripke relational semantics the truth of formulas depends on states of affairs of worlds different from the actual one (= possible worlds). On the other hand, in contemporary cosmology is nonsensical speaking of an "absolute truth of physical laws", with respect to a world where the physical laws cannot be always the same, but have to evolve with their referents [28, 29].

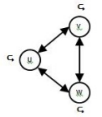
Anyway, the notion of "possible world" in Kripke semantics has not only a physical sense. On the contrary, as he vindicated many times, the notion of "possible world", as syntactic structure in a relational logic, has as many senses as the semantic models that can be consistently defined on it. In Kripke words, the notion of "possible world" in his semantics has a *purely stipulatory character.*

In the same way, in Kripke semantics, like the notion of "possible world" can be interpreted in many ways, so also the relations among worlds can be given as interpretations of the only relation of *accessibility.* In this way, a unified theory of the different intensional interpretations (alethic, ontology included, deontic, epistemic, etc.) of modal logic became possible, as well as a graphic representation of their relational semantics.

The basic notion for such a graphic representation is the notion of *frame.* This is an ordered pair , <**W**, *R*>, constituted by a domain **W** of possible worlds {*u, v, w*…}, and a by a two-place *relation R* defined on **W**, i.e., by a set of ordered pairs of elements of **W** ($R \subseteq \mathbf{W}\times\mathbf{W}$), where **W×W** is the *Cartesian product* of **W** per **W**.

E.g. with **W** = {*u*,*v*,*w*} and *R* = {*uRv*}, we have:

According to such a model, the accessibility relation *R* is only in the sense that *v* is accessible by *u*, while *w* is not related with whichever world. If in **W** all the worlds were reciprocally accessible, i.e., *R*= {*uRv*, *vRu*, *uRw*, *wRu*, *wRv*, *vRw*}, then we would have *R* only included in **W**×**W.** On the contrary, for having *R*=**W×W**, we need that each world must be related also with itself, i.e.:

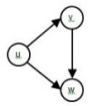


(3)

Hence, from the standpoint of the relation logic, i.e., by interpreting {*u*,*v*,*w*} as elements of a class we can say that this *frame* represents an *equivalence class.* In fact, a *R, transitive, symmetrhain ical* and *reflexive* relation holds among them. Hence, if we consider also the *serial relation:* <(om *u*)(ex *v*)(*uRv*)>[6], where "om" and "ex" are the meta-linguistic symbols, respectively of the universal and existential quantifier, we can discuss also the particular *Euclidean relation* that can be described in a Kripke frame.

The Euclidean property generally in mathematics means a weaker form of the transitive property (that is, if one element of a set has the same relation with other two, these two have the same relation with each other).

I.e.,<(om *u*) (om *v*) (om *w*) (*u*R*v* et *uRw* ⇒ v*R*w)> :



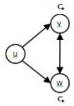
Where *et* is the meta-symbol for the logical product.

Hence, for seriality, it is true also <(om *u*)(om *v*) (*uRv*⇒*vRv*)>:

[6] For ontological applications it is to be remembered that seriality means in ontology that the causal chain is always closed, as it is requested in physics by the first principle of thermodynamics, and in metaphysics by the notion of a first cause of everything.

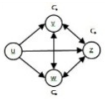
Moreover, <(om *u*) (om *v*) (om *w*) (*uRv* et *uRw* ⇒ *vRw* et *wRv*)>:

Finally, if we see at the last two steps, we are able to justify, via the Euclidean relation, a set of *secondary* reflexive and symmetrical relations, so that we have the final frame of a *secondary equivalence* relation among worlds based on an Euclidean relation with a third one:



(4)

Of course, this procedure of *equivalence constitution by a transitive and serial* (=causal) relation can be iterated indefinitely:



(5)

2.4 Double saturation S/P and its implementation

What characterizes the definite descriptions in a naturalistic formal ontology since of its Middle Age ancestors is the theory of *double saturation between Subject and Predicate (S/P),* driven by a causal action from the referential object. So Thomas Aquinas (1225-1274)[7] depicts his causal theory of reference:

> Science, indeed, depends on what is object of science, but the opposite is not true: hence the relation through which science refers to what is known is a *causal* [*real* not *logical*] relation, but the relation through which what is known refers to science is only *logical* [*rational* not *causal*]. Namely, *what is knowable (scibile) can be said as "related", according to the Philosopher, not because it is referring, but because something else is referring to it.* And that holds in all the other things relating each other like the measure and the measured, … (*Q. de Ver.*, 21, 1. Square parentheses and italics are mine).

In another passage, this time from his commentary to Aristotle book of *Second Analytics,* Aquinas explains the singular reference in terms of a "one-to-one universal", as opposed to "one-to-many universals" of generic predications.

> It is to be known that here "universal" is not intended as something predicated of many subjects, but according to some adaptation or adequation (adaptationem vel adaequation)of the predicate to the subject, as to which neither the predicate can be said without the subject, nor the subject without the predicate (In Post.Anal., I,xi,91. Italics mine).

So, Aquinas' idea is that the predicative statement, when applied denoting a singular object must be characterized by a "mutual redefinition" between the subject *S* and the predicate *P* "causally" driven by the referential object itself. A procedure formalized in Kripke's frame (4) and that A. L. Perrone first demonstrated being a finitistic computational procedure always convergent in polynomial time, even in chaotic dynamics [30].

[7] Historically, he first introduced the notion and the term of "intention" (*intentio*) in the epistemological discussion, in the context of his naturalistic ontology. The approach was hence rediscovered in the XIX century by the philosopher Franz Brentano, in the context of a conceptualist ontology, and hence passed to the phenomenological school, trough Brentano's most famous disciple: Edmund Husserl.

3 CONCLUSIONS

To conclude, it is important to emphasize that the frame (3) and the frame (4) are a graphic representation in Kripke's approach of **S5** and **KD45** modal systems respectively. If we see at these frames, we can understand immediately, why, from one side, **S5** is the only axiomatic system in modal logic, since all its elements constitutes one only equivalence class. On the other side, we can also understand immediately also why **KD45** is named "secondary **S5**". In fact, <*v*,*w*> in (4) and <*v*,*w*,*z*> in (5) constitute two equivalence classes via their Euclidean relation with <*u*>.

If **S5** is thus, in *ontology,* the common syntactic structure of all possible metaphysics, **KD45** is the common structure of any ontology of the *emergence* of a new natural law (and hence of a new natural kind) from more fundamental levels of physical causality.

Moreover, in *epistemic logic*, if <*u*> represents the *referential object*, then <*v*,*w*> and/or <*v*,*w*,*z*> represent two equivalence classes of two and/or three symbols co-denoting it. Two classes of logical symbols with their relations constituted via a common causal relation from the denoted object! It is evident that such a solution of the reference problem can be implemented only in a NC model where <*v*,*w*,*z*> can be interpreted as *cognitive agents,* particularly as (pseudo-)cycles of the same chaotic attractor, as we and others demonstrated elsewhere (See [30, 18, 31, 32]; [33, 34, 35]).

REFERENCES

[1] A. M. Turing, "On computable numbers with an application to the 'Entscheidung problem'" *Proceedings of the London Mathematical Society,* vol. 42, pp. 230-265, 1936.

[2] A. M. Turing, "Systems of logic based on ordinals" *Proceedings of the London Mathematical Society,* vol. 2, n. 45, pp. 161-228, 1939.

[3] A. M. Turing, "Lecture to the London Mathematical Society on 20 February 1947" in *Collected Works. I: Mechanical Intelligence,* D. C. Ince, A cura di, Amsterdam, North Holland, 1992, pp. 87-105.

[4] A. M. Turing, "Intelligent Machinery, report for the National Physical Laboratory, 1948" in *Collected Works, I: Mechanical Intelligence,* D. C. Ince, Ed., Amsterdam, North Holland, 1992, pp. 87-106.

[5] A. M. Turing, "The chemical basis of morphogenesis" *Phil. Trans. R. Soc. London B,* vol. 237, p. 37–72, 1952.

[6] G. Dodig-Crnkovic, "The significance of models of computation. From Turing model to natural computation" *Mind and Machine,* pp. DOI 10.1007/s11023-011-9235-1, 2011.

[7] G. Dodig-Crnkovic, "Physical computation as dynamics of form that glues everything together" *Information,* vol. 3, pp. 204-218, 2012.

[8] N. B. Cocchiarella, "Logic and ontology," *Axiomathes,* vol. 12, pp. 117-150, 2001.

[9] N. B. Cocchiarella, *Formal Ontology and Conceptual Realism,* Berlin-New York: Springer Verlag, 2007.

[10] G. Basti, "Ontologia formale: per una metafisica post-

moderna" in *Il problema dei fondamenti. Da Aristotele, a Tommaso d'Aquino, all'Ontologia Formale*, A. Strumia, A cura di, Siena, Cantagalli, 2007, pp. 193-228.

[11] G. Basti, "Ontologia formale. Tommaso d'Aquino ed Edith Stein," in *Edith Stein, Hedwig Conrad-Martius, Gerda Walter. Fenomenologia della persona, della vita e della comunità*, A. Ales-Bello, F. Alfieri and M. Shahid, Eds., Bari, Laterza, 2011, pp. 107-388.

[12] U. Meixner, *The theory of ontic modalities*, Frankfurt: Ontos Verlag, 2007.

[13] A. Tarski, "The Concept of Truth in Formalized Languages" in *Logic, Semantics, Metamathematics*, 2 a cura di, J. Corcoran, A cura di, Indianapolis, Hackett, 1983, 1935, p. 152–278.

[14] R. Carnap, "Testability and meaning" *Philosophy of science*, vol. 3, n. 4, pp. 419-461, 1936.

[15] H. Putnam, "The meaning of 'meaning'" in *Philosophical papers II: mind, language and reality*, Cambridge MA, Cambridge UP, 1975, pp. 215-271.

[16] J. A. Fodor, "Metodological solipsism considered as a resarch strategy in cognitive psychology," *Behavioral and brain sciences*, vol. 3, no. 1, pp. 63-73, 1980.

[17] W. V. O. Quine, "Sticks and stones or the ins and the outs of existence," in *On Nature, Boston Univ.Studies in Philosophy and Religion*, vol. 6, L. S. Rounder, Ed., Notre Dame, Ind.: Notre Dame UP, 1984, pp. 13-26.

[18] G. Basti e A. L. Perrone, "Intentionality and Foundations of Logic: a New Approach to Neurocomputation" in *What should be computed to understand and model brain function?- From Robotics, Soft Computing, Biology and Neuroscience to Cognitive Philosophy*, T. Kitamura, A cura di, Singapore, New York, World Publishing, 2001, pp. 239-288.

[19] G. Basti e A. L. Perrone, *Le radici forti del pensiero debole. Dalla metafisica, alla matematica, al calcolo*., Padua-Rome: Il Poligrafo and Lateran UP, 1996.

[20] G. Basti e A. L. Perrone, "Chaotic neural nets, computability, undecidability. An outlook of computational dynamics" *International Journal of Intelligent Systems*, vol. 10, n. 1, pp. 41-69, 1995.

[21] S. Kripke, *Naming and necessity*, Cambridge MA: Harvard UP, 1980.

[22] S. C. Kleene, *Introduction to metamathematics*, Amsterdam: North Holland, 1952.

[23] H. Putnam, *Representation and reality*, Camridge MA: MIT Press, 1988.

[24] H. Putnam, "Meaning and reference," *The Journal of Philosophy*, vol. 70, no. 19, pp. 699-711, 1973.

[25] K. S. Donnellan, "Reference and definite descriptions" *The Philosophical Review*, vol. 75, pp. 281-304, 1966.

[26] J. P. Burgess, *Philosophical logic (Princeton foundations of contemporary philosophy)*, Princeton NJ: Princeton UP, 2009.

[27] S. Galvan, *Logiche intensionali. Sistemi proposizionali di logica modale, deontica, epistemica*, Milano: Franco Angeli, 1991.

[28] P. Davies, "Universe from bit" in *Information and the nature of reality. From physics to metaphysics.*, P. Davies e N. H. Gregersen, A cura di, Cambridge, Cambridge UP, 2010, pp. 65-91.

[29] P. Benioff, "Towards A Coherent Theory of Physics and Mathematics: The Theory-Experiment Connection" *Foundations of Physics*, vol. 35, pp. 1825-1856, 2005.

[30] A. L. Perrone, "A formal Scheme to Avoid Undecidable Problems. Applications to Chaotic Dynamics" *Lecture Notes in Computer Science*, vol. 888, pp. 9-48, January 1995.

[31] G. Basti e A. L. Perrone, "Neural nets and the puzzle of intentionality" in *Neural Nets. WIRN Vietri-01. Proceedings of 12th Italian Workshop on Neural Nets, Vietri sul Mare, Salerno, Italy, 17-19 May 2001*, Berlin, London, 2002.

[32] G. Basti, "Logica della scoperta e paradigma intenzionale nelle scienze cognitive" in *Quale scienza per la psicoterapia? Atti del III Congresso nazionale della SEPI (Society for the Exploration of Psychotherapy Integration)*, T. Carere-Comes, A cura di, Firenze, Florence Art Edition, 2009, pp. 183-216.

[33] W. J. Freeman, *How brains make up their minds*, New York: Columbia UP, 2001.

[34] W. J. Freeman, "Intentionality" 2007. [Online]. Available: http://www.scholarpedia.org/article/Intentionality.

[35] W. J. Freeman, "Nonlinear dynamics and the intention of Aquinas" *Mind and Matter*, vol. 6, n. 2, pp. 207-234, 2008.

[36] J. Gribbin, *Deep simplicity. Chaos, complexity and the emergence of life*, New York: Penguin Book, 2003.

[37] G. Dyson, "Turing centenary: the dawn of computing" *Nature*, vol. 482, pp. 459-460, 2012.

[38] J. Reinitz, "Turing centenary: pattern formation" *Nature*, vol. 482, pp. 461-462, 2012.

[39] B. Cooper, "Turing centenary: the uncomputable reality" *Nature*, vol. 482, p. 465, 2012.

[40] B. Smith, A cura di, *Parts and Moments. Studies in Logic and Formal Ontology*, Munich: Philosophia, 1982.

[41] B. Smith, "Beyond Concepts, or: Ontology as Reality Representation" in *Formal Ontology and Information Systems. Proceedings of the Third International Conference (FOIS 2004)*, Amsterdam, 2004.

[42] B. Smith, "Against Fantology," in *Experience and Analysis*, J. C. Marek and M. E. Reicher, Eds., Wien, HPT&ÖBV, 2005, pp. 153-170.

[43] J. R. Searle, "Mind, brains and programs. A debate on artificial intelligence" *The Behavioral and Brain Science*, vol. 3, pp. 128-135, 1980.

[44] J. R. Searle, *Intentionality. An essay in the philosophy of mind*, New York: Cambridge UP, 1983.

[45] W. Sellars, "Intentionality and the Mental" in *Minnesota Studies in the Philosophy of Mind. Vol. II: "Concepts, Theories and the Mind-Body Problem"*, H. Feigl, M. Scriven e G. Maxwell, A cura di, Minneapolis, Minnesota UP, 1958, pp. 507-524.

[46] P. F. Strawson, *Individuals. An essay in descriptive metaphysics*, London & New York: Routledge, 2003, 1959.

[47] W. Sellars e R. Rorthry, *Empiricism and the Philosophy of Mind. With an introduction of Richard Rorthry*, Boston Ma.: Harvard UP, 1997.

[48] E. Zalta, *Intensional logic and the metaphysics of intentionality*, Cambridge MA: MIT Press, 1988.

[49] L. Szilard, "On the decrease of entropy content in s thermodynamical system by the intervention of intelligent beings" *Behavioral Science*, vol. 9(4), pp. 301-310, 1964.

MENS, an info-computational model for (neuro-)cognitive systems up to creativity

Andrée C. EHRESMANN[1]

Abstract. MENS is a bio-inspired model for higher level cognitive systems; it is an application of the Memory Evolutive Systems developed with J.-P. Vanbremeersch [12], to model complex multi-scale, multi-agent self-organized systems, such as biological or social systems. Its development resorts from an info-computationalism: first we characterize the properties of the human brain/mind at the origin of higher order cognitive processes up to consciousness and creativity, then we 'abstract' them in a mathematical model MENS for natural or artificial cognitive systems. The model, based on a 'dynamic' Category Theory incorporating Time, emphasizes the computability problems which are raised.

1 INTRODUCTION

One of the aims of this Conference is: "understanding of computational processes in nature and in the human mind". This aim has been central in the development of the *Memory Evolutive Neural Systems* (or MENS), a mathematical model of a cognitive system, such as the brain/mind, allowing for the emergence of higher order cognitive processes, up to thought, consciousness and creativity.

MENS proposes a common frame accounting for the functioning of the neural and of the mental and cognitive system at different levels of description and across different timescales. It does not constitute a logic model of the invariant structure of the neuro-cognitive system; it is intended to give a dynamic model sizing up the system 'in the making', with the variation over time of its configuration and of its information processing. It describes how various brain areas interact as hybrid systems to generate an "algebra of mental objects" (in the terms of Changeux [5]) through iterative 'binding' of more and more complex synchronous assemblies of neurons. In its frame mental objects are treated as 'higher level' neurons (called *category-neurons*) on which to compute how cognitive processes of increasing complexity can emerge.

The bio-inspired development of MENS has followed the two directions proposed by G. Dodig-Crnkovic: analyzing living organisms as info-computational systems/agents, and implementing natural computation strategies [8]. Indeed, first we characterize the properties of the human brain/mind at the root of cognitive processes; then we 'abstract' them in MENS. A third step would be to develop an adequate kind of (probably unconventional) computation to simulate them

MENS is an application of the *Memory Evolutive Systems* (MES), developed with J.-P. Vanbremeersch [12], which give a model, based on Category Theory, for complex multi-scale, multi-agent self-organized systems, such as biological, social or cognitive systems.

In Section 2, we make some recalls on MES, indicating the role of Category Theory in them. Section 3 emphasizes the neural basis of MENS. A description of the structure and of the local/global dynamic of MENS is given in Sections 4 and 5, while Section 6 deals with the emergence of higher cognitive processes. The conclusion proposes an extension of MENS to artificial cognitive systems, and emphasizes the computational problems which it raises.

2 WHY CATEGORY THEORY IN MES?

Category Theory has a unique status at the border between mathematics, logic, and meta-mathematics. Introduced by Eilenberg and Mac Lane [10] in the early forties, its development (e.g. by Kan [21], Ehresmann [15], Lawvere [22]) has provided a setting in which a general concept of structure is possible, and essential mathematical constructions are unified thanks to a capture of their common roots in the ways of thinking of the "working mathematician". As these ways reflect some of the main mental operations at the basis of science, it is natural that categories have begun to be applied to other scientific domains, in particular computer science, physics, complexity theory and biology.

Graphs are extensively used to represent networks of any nature. A *category* is a graph equipped with an internal composition associating to a pair (*a*, *b*) of successive arrows (or *links*) *a*: A → B and *b*: B → C, a *composite* arrow *ab*; A → C; this composition is associative and each object has an *identity*. Each graph generates the *category of its paths*: the objects are the same, the links are the paths (sequences of successive arrows), composition is by convolution. Each category is the quotient of the category of its paths by the equivalence: two paths are *functionally equivalent* if they have the same composite.

If we use categories rather than simple graphs in our study of complex systems, it is because they open the way to important "universal constructions", such as the *colimit* operation which will model the 'binding' of a pattern P of linked objects. A *pattern* (or diagram) P in a category is a family of objects P_i with some distinguished links *f*: $P_i \to Pj$. A *collective link* from P to

[1] Université de Picardie Jules Verne, Faculté des Sciences Mathématiques, Amiens France. Email: ehres@u-picardie.fr
[2] http:// ehres.pagesperso-orange.fr.

an object N is a family of links- s_i from the different P_i to N, such that $fs_j = s_i$ for each distinguished link f: $P_i \rightarrow P_j$ of P. The pattern admits a *colimit* (or inductive limit [21]) M if there is a collective link from P to M which factorizes any other collective link, so that the collective links (s_i) from P to any N are in 1-1 correspondence with the links s: M → N binding them.

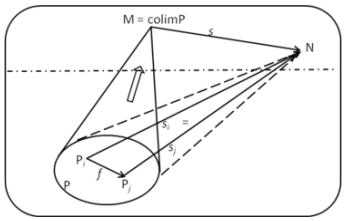


FIGURE 1. Collective link, and colimit of a pattern P

The Memory Evolutive Systems give a model based on a 'dynamic' Category Theory, incorporating time and durations, for complex multi-scale systems, with the following characteristics:

(i) *The system is evolutionary*, its components and their links varying over time. The few models of complex systems using category theory (e.g., inspired by [24]) only consider one category representing the invariant structure of system. On the contrary in MES, the system is not represented by a unique category but by an *Evolutive System* consisting in: a family of categories K_t, representing the successive configurations of the system at each time *t*, and partial transition functors from K_t to $K_{t'}$ accounting for the change from *t* to *t'*.

(ii) *The system is hierarchical*, with a tangled hierarchy of components varying over time. A component C of a certain level 'binds' at least one pattern P of interacting components of lower levels so that C, and P acting collectively, have the same functional role. Modeling this hierarchy raises the *Binding Problem*: how do simple objects bind together to form "a whole that is greater than the sum of its parts" [1] and how can such "wholes" interact? In the categorical setting, the 'whole' C is represented by the colimit of the pattern P of interacting simple objects; and the interactions between wholes are described.

(iii) *There is emergence of complex multiform components*, with development of a *flexible central memory*. Whence the *Emergence Problem*: how to measure the 'real' complexity of an object and what is the condition making possible the emergence over time of increasingly complex structures and processes? We characterize this condition as the *Multiplicity Principle* [12], a kind of 'flexible redundancy' which ensures the existence of multiform components. And we prove that it is necessary for the emergence of increasingly complex objects and processes with multiform presentations, constructed by iterated *complexification* processes [11].

(iv) *The system has a multi-agent self-organization*. Its global dynamic is modulated by the cooperation/competition of a network of internal functional subsystems, the *co-regulators*, with the help of a long-term memory. Each co-regulator operates locally with its own rhythm, logic and complexity, but their different commands can be conflicting and must be harmonized. While the local dynamics are amenable to conventional computations, the problem is different for the global one.

MENS is a MES the level 0 of which represents the 'physical' neural system (neurons and synapses), while its higher level components are 'conceptual' objects (called *category-neurons*) which represent mental objects as the binding of synchronous (hyper-)assemblies of neurons.

3 PROPERTIES OF THE NEURAL SYSTEM

Despite the huge progresses of brain research in the last 20 years, we do not understand the brain's large-scale organizational principles allowing for the emergence of higher order cognitive processes. Interesting mathematical models of a local nature have been developed for particular processes in specialized brain areas; as the different brain areas are heterogeneous both anatomically and functionally, such models cannot be extended to other areas or processes.

However, there are some general properties, and MENS relies on them:

(i) *Graphs of neurons.* The neurons and the synapses existing at an instant *t* form a graph; a neuron has an activity at *t*, a synapse from N to N' has a *propagation delay* and a *strength* depending on how it may transmit the activity of N to N'; the synapse can be active or passive at *t*. The activity of N is a sum of the activities of the neurons connected to N by an active link, pondered by the strength of this link. The graph changes over time: some neurons 'die', new neurons are formed, and the same for synapses; delays and strengths may vary.

(ii) *The structural core.* The graph of neurons has a central sub-graph, called its *structural core*, discovered by Hagmann & *al.* [17] in 2008: "Our data provide evidence for the existence of a structural core in human cerebral cortex […] both spatially and topologically central within the brain […] an important structural basis for shaping large-scale brain dynamics […] linked to self-referential processing and consciousness." Recently (2011) it has been found that this core is a sub-graph with several hubs forming a "rich club" [25].

(iii) *Synchronous assemblies of neurons.* Already in the forties Hebb [18] has noted the formation, persistence and intertwining of more or less complex and distributed assemblies of neurons whose synchronous activation is associated to specific mental processes: "Any frequently repeated, particular stimulation will lead to the slow development of a "cell-assembly" as a close system". And he gives the *Hebb rule* for synaptic plasticity: "When an axon of cell A is near enough to excite B and repeatedly or persistently takes part in firing it […] A's efficiency, as one of the cells firing B, is increased."

(iv) *Degeneracy property of the neural code*. Emphasized by Edelman, it says that: "more than one combination of neuronal groups can yield a particular output, and a given single group can participate in more than one kind of signaling function." [9]. Thus the mental representation of a stimulus should be the common 'binding' of the more or less different neural patterns which it can synchronously activate in different contexts or at different times.

(v) *Modular organization.* The brain has a modular organization, with a variety of 'modules' or areas of the brain with a specific function, from small specialized parts (the "treatment units" of Crick [6]) such as visual centers processing colour, to large areas such as the visual or motor areas, or nuclei of the emotive brain (brain stem and limbic system) or the associative cortex.

The neural system will be represented by the *Evolutive System of Neurons* NEUR: it has for configuration at *t* the *category of neurons* $NEUR_t$: its objects, also called *neurons*, model the neurons N existing at *t* with their activity, the links model the *synaptic paths* between them, labeled by their propagation delay and strength (defined as the sum of those of their factors).

The transition from *t* to a later time *t'* associates to the state at *t* of a neuron N its new state at *t'* provided that N still exists at *t'*, and similarly for the links. The transitions describe what has changed, but they do not indicate the kind of computation (as processing of information) which is internally responsible for the change. A component of NEUR models a neuron through the sequence of its successive states.

NEUR constitutes the level 0 of MENS, from which higher levels are constructed by iterated complexification processes.

4 CATEGORY-NEURONS AND THEIR LINKS

As said above, a mental object (*e.g.*, the mental image of a simple stimulus) synchronously activates an assembly of neurons P, and possibly several ones in different contexts. In simple cases, there is a neuron N 'binding' the assembly, so that it will represent the mental object; for instance there are neurons representing a segment or an angle [19], or more complex but very familiar objects.

However, generally there is no "grand-mother neuron" [3] in NEUR. The mental object activating P will be represented by a conceptual object M, called a *category-neuron* (abbreviated in *cat-neuron*) of level 1, which will become the colimit of P, not in NEUR (where it has no colimit), but in the larger system MENS. The construction (by a *complexification* process) will determine what are the good links between M and other (cat-)neurons, and will guarantee that M also becomes the colimit of the other assemblies of neurons which P can synchronously activate. Thus we can speak of *assemblies of cat-neurons* of level 1, and iterate the construction to obtain a hierarchy of cat-neurons of increasing levels, representing more and more complex mental objects binding together assemblies of simpler ones.

Formally, any assembly of (cat-)neurons is modeled by a *pattern* P in MENS. For the assembly to synchronously activate a (cat-)neuron N, there must exist a collective link (s_i) from P to N, allowing that all the s_i transmit an activation of P_i to N at the same time; in particular this imposes that all the zig-zags of links between P_i and P_j have the same propagation delay.

If such a pattern P is repeatedly activated, its distinguished links are strengthened (via Hebb rule), and there is formation of a mental object. This object will be represented by a higher level cat-neuron M, which becomes the *colimit* of P in MENS (cf. Figure 1). It is important to note that the activation of P precedes that of its colimit M.

The degeneracy property asserts that the mental object can also activate other patterns Q, not necessarily connected to P by a cluster of links. The representing cat-neuron M must also be the colimit of Q, so that it is a *multiform cat-neuron* [12], which can be activated by anyone of its different decompositions P, Q, …, with possibility of switches between them. The existence of multiform cat-neurons signifies that MENS satisfies the *Multiplicity Principle* [12]. Once formed the cat-neuron M preserves its identity up to its 'death' though its lower level decompositions can vary more or less quickly over time. The *stability span* of M at an instant *t* is the longest period during which M admits a decomposition P at *t* whose successive states remain a decomposition of M.

MENS is an Evolutive System. At an instant *t* of the life of the individual, the configuration category $MENS_t$ models the present state of the neural, mental and cognitive system; its objects are the cat-neurons of any level (from the level 0 of neurons up) existing at *t* with their activity, and their links with their propagation delay and strength; a link is active or not at *t.*

The transition from *t* to *t'* points out the structural changes without accounting for the information processing at their origin (to be considered in Section 5). The changes are events of the following kinds: formation (or preservation if it exists) of a new cat-neuron binding some pattern P' of already existing lower level cat-neurons, possibly loss or decomposition of some cat-neurons. In the categorical setting, the new configuration $MENS_{t'}$ at *t'* is obtained as the *complexification* of $MENS_t$ with respect to a procedure Pr having objectives of the preceding kinds (cf. Figure 2). Such a complexification is solution of the "universal problem" of constructing a category in which the objectives of Pr are satisfied in the 'best' way. We have given an explicit construction of the complexification, in particular of the links between cat-neurons; and we have shown in [13] how, using its universal property, the propagation delays and strengths of synaptic paths (at the level 0) can be extended to the links of any level, as well as the *Hebb rule*.

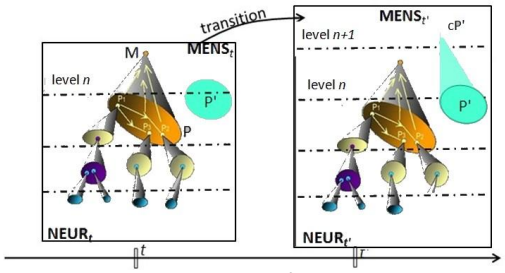


FIGURE 2. Complexification process

The construction distinguishes 2 kinds of links (Figure 3):

(i) *Simple links.* They 'bind' clusters of lower level links as follows. Let M and M' be 2 cat-neurons binding lower level patterns P and P' respectively. If we have a *cluster* G of links from P to P' well correlated by the distinguished links of P and P', this cluster binds into a link from M to M', called a (P, P')-*simple link* (or just a *n-simple link* if P and P' are of level $\leq n$). Such a link just translates at the level *n*+1 the information that P can coherently activate components of P' through the links of G; and this information is computable at the lower levels. A composite of *n*-simple links binding adjacent clusters is *n*-simple.

(ii) *Complex links.* They emerge at a higher level, as composites of *n*-simple links binding non-adjacent clusters. Their existence is possible because of the existence of multiform cat-neurons M. Figure 3 presents a complex link from N to M' composite of a (Q', Q)-simple link with a (P, P')-simple link, where P and Q are non-connected decompositions of M. Such a link represents information *emerging* at the level *n*+1 by integration of the global structure of the lower levels, and not

locally computable through lower level decompositions of N and M'; indeed the fact that the cat-neuron M is multiform imposes global conditions, calling out all its lower decompositions and their collective links; could it be amenable to some kind of unconventional computation?

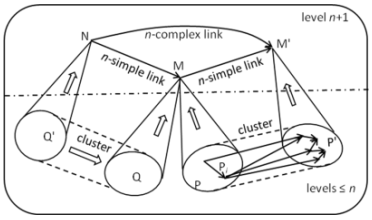


FIGURE 3. Cluster, simple and complex links

Remark. Here we only speak of cat-neurons constructed by colimits. In fact there are also cat-neurons obtained by projective limits [21], which arise for instance in the construction of a semantic memory. When the procedure asks also for the formation of such 'classifying' cat-neurons, we speak of a *mixed complexification*; its construction is more complicated [12].

The construction of cat-neurons of higher levels allows making more precise the brain/mind correlation. The *activation* or the *recall* of the cat-neuron M of level > 0 representing a mental object consists in the unfolding of one of its ramifications down to the neural level (cf. Figure 3): first activation of one of its decompositions P into a synchronous assembly of cat-neurons of lower levels, then a decomposition of each component of P, and so on down to the *physical activation* of synchronous assemblies of neurons. Because of the propagation delays of the links, the unfolding has a certain duration.

At each step, there is a choice between various (possibly non-connected) decompositions, so that the activation of M has several freedom degrees leading to *multiple physical realizabilities* into hyper-assemblies (*i.e.* assemblies of assemblies of… assemblies) of neurons. The ramifications of M have not all the same length. The *complexity order* of M is the smallest length of a ramification; it is less or equal to the level of M. The level indicates the number of steps in which M has been constructed, while the complexity order measures the smallest number of steps sufficient for its later activation.

From general results on complexifications of categories satisfying the Multiplicity Principle [12], we deduce:

THEOREM. *Iterated complexifications preserve the Multiplicity Principle and lead to the emergence in* MENS *of cat-neurons of increasing complexity order, representing more and more complex mental objects or cognitive processes.*

5 LOCAL AND GLOBAL DYNAMIC OF MENS

As any Memory Evolutive System, MENS has a multi-scale self-organization modulated by a network of co-regulators with the help of a central long-term memory.

The *Memory* is a hierarchical sub-system Mem of MENS, which develops over time; it models the innate or acquired knowledge of any modality and the information of any kind which the individual can store and later recognize and/or recall. A cat-neuron M in Mem, called *record*, represents the mental object associated to an item S (external object, signal, past event, internal state, sensory-motor or cognitive processes,..). Initially constructed to bind a particular pattern P of cat-neurons activated by S, it later takes its own identity as a multiform cat-neuron and can even disassociate from P at a later time to adapt to changing situations (as long as the change is progressive enough). S can be recognized and M recalled through the activation of any of the ramifications of M, with possibility of switches between them, so that M is a robust memory but not a rigid one (as in a computer), since it remains flexible and can be constantly revised to account for changes.

Mem contains a sub-system Proc, the *Procedural Memory* in which the records, called *procedures*, have links (or 'commands') toward the pattern of their effectors (*e.g.* motor commands of a specific movement). It also contains a sub-system Sem, the *Semantic Memory*, in which records are classified into invariance classes with respect to some attributes (for the construction of Sem, cf. [12]).

The memory plays an important role in the dynamic of MENS which is modulated by the cooperative/ competitive interactions between functional sub-systems, the *co-regulators*, related to the modular organization of the brain. A co-regulator is based on a specific module of the brain, meaning that its cat-neurons have ramifications down to this module (so that they model hyper-assemblies of neurons of the module). It has its own differential access to Mem, in particular to Proc, to recall its 'admissible procedures' specific of its function.

The dynamic of MENS must account for both the local information processing of each co-regulator, which operates with its own rhythm and logic, and for the global dynamic which results from an 'interplay' among these co-regulators. While the local dynamics are amenable to conventional computations, their merging in the global one raises computational problems.

A co-regulator CR operates stepwise as a hybrid system, A step from t to t' is divided into more or less intermingled phases:

(i) *Formation of the landscape at t.* It is a category L_t which models the partial information accessible to CR through active links: its objects are clusters G from a cat-neuron B to CR with at least one link activating a cat-neuron in CR around t. It plays the role of a working memory during the step.

(ii) *Selection of an admissible procedure* Pr to respond to the situation with adequate structural changes. It is done through the landscape, using the access of CR to Mem to recall how the information has been processed in preceding analogue events. For instance in presence of an object S, a CR treating colours will retain only information on the colour of S, and the objective of Pr could be to bind the pattern P of neurons activated by the colour to memorize the colour or, if already known, recall it.

(iii) *Commands of the procedure* are sent to its effectors in MENS. In the above example, the binding of P into a CR-record of S consists in strengthening the distinguished links of P using Hebb rule. The dynamic by which the effectors realize the commands during the continuous time of the step is computable through conventional computations (e.g., using differential equations implicating the activity of cat-neurons and the propagation delays and strengths of the links [13]).

(iv) *Evaluation at the beginning of the next step*, by comparison of the anticipated landscape (which should be the complexification of L_t with respect to Pr) with the new landscape; then Pr and its result are recorded. If the commands of Pr have not entirely succeeded, we say that there is a *fracture* for CR.

The global dynamic must take account of the different local dynamics. At a given time the commands sent by the various co-regulators should all be realized by the effectors of the system. Since the co-regulators have different functions and rhythms, these commands can be conflicting, and there is need of an equilibration process to ensure the correlation of the different commands, possibly neglecting some of them. For instance to seize an object, the visual and motor commands should fit together. This process, called the *interplay among the co-regulators*, leads to the *operative procedure* Pr° which will be implemented on the system.

The interplay searches for a best compromise between the more or less conflicting commands, keeping as much as possible of them. In particular it takes benefit of the degrees of freedom of a multiform command which can be activated through anyone of its lower level decompositions, with possible switches between them: the decompositions allowing for a better coordination are selected through a kind of Darwinian selection process; for instance, depending on the context, we can seize an object in the right or left hand.

The operative procedure Pr° actually carried out may by-pass the procedures of some co-regulators thus causing dysfunction (temporary fracture or longer *de-synchrony*) to them. A main cause of fractures is the non-respect of the structural temporal constraints (or *synchronicity laws*) imposed on a co-regulator CR by the propagation delays and stability spans in its landscape [12]. Fractures may backfire between co-regulators with heterogeneous complexity levels and temporalities. In the interplay, an important role is played by *evaluating co-regulators*, based on parts of the emotive brain which evaluate the procedures in function of their consequences on the well-being of the person.

A standing problem is to determine what kind of computation could help model the interplay among the co-regulators. Since it makes use of the flexibility of the commands as multiform cat-neurons, it is probably not amenable to conventional computations (cf. Section 4).

6 AC AND HIGHER COGNITIVE PROCESSES

The co-regulators jointly participate in the development over time of an important functional sub-system of the memory Mem, the *Archetypal Core* AC which will act as an internal model, essential for the emergence of higher cognitive processes.

In Section 3 we have said that the brain has a structural core which plays a main role in the shaping of large-scale brain dynamic; it corresponds to the level 0 of AC. A cat-neuron in AC is a higher order cat-neuron, often activated and with ramifications down to the structural core; thanks to the "rich club" organization of this core, the hyper-assemblies of neurons which it binds are largely distributed in different brain areas. Thus an archetypal record integrates and intertwines recurring memories and experiences of different modalities (sensory-motor, proprioceptive, affective, …) as well as notable events with their emotive undertones. Archetypal records are connected by complex links which become stronger and faster (thanks to Hebb rule) along time. These links form *archetypal loops* which propagate very quickly the activation of an archetypal record A back to itself, thus maintaining it for a long time; the activation of A resonates to lower levels via the unfolding of ramifications of A and switches between different decompositions.

It follows that an activation of part of AC extends to a larger domain D of MENS, both in depth (lower level decompositions P are activated) and in duration: if A is activated at *t*, it means that P has been activated earlier (cf. Section 4), and, since the activation of A is self-maintained by the loops, the activation of P will been maintained in the near future: the 'present' of D has some extent, as proposed by Husserl: "Il y a dans le présent une *rétention* du passé (rétention primaire si c'est un passé immédiat, rétention secondaire si c'est un souvenir plus lointain) et une *protention* du futur (de ce qui va immédiatement arriver)." [20]

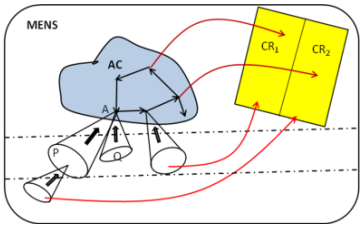


FIGURE 4. Archetypal Core at the basis of GL

AC represents an internal model of the Self, reflecting a personal vision of the world since each ramification of an archetypal record represents a specific association of mental objects dependent on the former experiences of the person. It plays a motor role in the development of higher cognitive processes, through information processing by higher level co-regulators based on associative brain areas and directly linked to AC; these co-regulators, called *intentional co-regulators*, can be compared to the "conscious units" of Crick [6]. An arousing situation or a non-expected event S (such as a fracture in a higher co-regulator) leads to the activation of some archetypal cat-neurons. As explained above, it triggers, through archetypal loops, an extension of the activated domain D. This activation is transmitted back to the intentional co-regulators, which can cooperate to construct a *global landscape* GL uniting their respective landscapes and extending them in depth and duration. GL assembles information related to the present state, reinforces evanescent traces recently accumulated in lower levels of the working memory, and even anticipates some future trends. Successive global spaces partially overlap each other.

The global space GL can be compared to the "global workspace" proposed by different authors (e.g. [7], and to the "theater of consciousness" of Baars [2]. It gives a frame for the development of higher cognitive processes, in particular *conscious processes* characterized by an integration of the time dimension through 2 possibly alternative and/or intermingled processes which extend Husserl's retention and pretention:

(i) A *retrospection process* (toward the past) proceeds by abduction (in the sense of Pierce [23]) to recollect information back in time thanks to its reinforcement in GL. Processing this

information allows for analyzing the event S which has triggered the formation of GL and finding its possible causes, thus "sensemaking" of the present.

(ii) A *prospection process* (toward the future) is then developed in GL, to try and select long term strategies. It is done through the formation, inside GL, of local 'virtual' landscapes (representing "mental spaces"), where successive procedures can be tried by constructing the corresponding complexifications, with evaluation of their benefits and of the risk of dysfunction. A sequence of alternating retrospection and prospection processes thus leads to various 'scenarios'. Once a scenario is selected, the retrospection process allows back-casting to find sequences of procedures (implicating co-regulators of various levels) able to realize this long term program.

The formation of scenarios is at the root of anticipation and creativity. Scenarios directly inspired by the contextual environment and current trends are obtained by simple complexifications of successive virtual landscapes that add, delete and/or combine components (examples: "combinational" and "exploratory" creativity [4]; metaphors and "conceptual blending" [16]). More innovative scenarios ("transformational creativity" [4]) make use of iterated complexifications which (as asserted by the "Iterated complexification Theorem" [12]) are not reducible to a unique complexification and lead to the emergence of mental objects of increasing complexity, so that these scenarios transcend the current situation.

Successive global spaces can 'consciously' process information coming from higher-order cat-neurons, while they 'automatically' keep traces of the operations of lower level co-regulators (recruited by retrospection). It explains how a creative process can go through an 'incubation period' during which the person consciously performs unrelated operations, followed by an "insight" with emergence in the global space of new ideas for the creative scenario.

7 CONCLUSIONS & FUTURE WORK

This paper shows how to model a "theory of mind", in which a hierarchy of mental objects and processes emerges from the functioning of the brain, through the iterative binding of neuronal (hyper-)assemblies. We show that *the degeneracy property of the neural code* is the characteristic which makes this emergence possible, and we explain how it allows the development of a flexible memory, with a central part, the Archetypal Core AC at the basis of the Self and of the formation of higher cognitive processes up to consciousness and creativity. The info-computational model MENS is an application of the Memory Evolutive Systems [12], which are based on a 'dynamic' Category Theory.

The same constructions could lead to the development of an *artificial cognitive system* with higher cognitive processes, provided it is based on a graph satisfying the assumptions of Section 3 (structural core, reinforcement of active assemblies of objects, degeneracy property). Or to the development of Neuro-Bio-ICT systems enlarging MENS, obtained by connecting, to the neural system, an artificial cognitive system acting as an "exocortex" to monitor dysfunctions and/or enhance human abilities [14].

In any case, the local dynamics should be computable, but the problem remains to study how the interplay between them leading to the global dynamic could be amenable to some kind of (hyper-?)computation.

REFERENCES

[1] Aristotle, *Metaphysics: Books Z and H*, (translated with a commentary by D. Bostok), Clarendon Aristotle Series (1994).

[2] B.J. Baars, *In the theatre of consciousness: The workspace of the mind*, Oxford University Press. Oxford (1997).

[3] H.B. Barlow, Single units and sensation: A neuron doctrine for perceptual psychology, *Perception* 1, 371-394 (1972).

[4] M.A. Boden, *The creative mind: myths and mechanisms*, 2nd edition, Routledge, London and New York (2004).

[5] J.-P. Changeux, *L'homme neuronal*, Fayard, Paris (1983).

[6] F. Crick, *The Astonishing Hypothesis*, Macmillan Publishing Company, New York (1994).

[7] S. Dehaene, C. Sergent and J.-P. Changeux, A neuronal network model linking subjective reports and objective physiological data during conscious perception, *Proc. Natl. Acad. Sc. USA* 100, 8520 (2003).

[8] G. Dodig-Crnkovic and V. Müller, A Dialogue Concerning Two World Systems: Info-Computational vs. Mechanistic. In Eds: Dodig-Crnkovic and M Burgin, *Information and Computation*, World Scientific Publishing (2011). http://arxiv.org/abs/0910.5001 2009

[9] G.M. Edelman, *The remembered Present*, Basic Books, New York (1989).

[10] S. Eilenberg and S. Mac Lane, General theory of natural equivalences, *Trans. Am. Math. Soc*. 58, 231-294 (1945)..

[11] A.C. Ehresmann and J.-P. Vanbremeersch, Hierarchical Evolutive Systems: A mathematical model for complex systems, *Bull. of Math. Bio*. 49 (1), 13-50 (1987).

[12] A.C. Ehresmann and J.-P. Vanbremeersch, *Memory Evolutive Systems: Hierarchy, Emergence, Cognition*, Elsevier, Amsterdam (2007).

[13] A.C. Ehresmann and J.-P. Vanbremeersch, A propos des Systèmes Evolutifs à Mémoire et du modèle MENS**,** *Compte-rendus du Séminaire Itinérant des Catégories*, Paris (2009).

[14] A. C. Ehresmann, R. von Ammon, .K. Iakovidis and A. Hunter, Ubiquitous complex events processing in Exocortex applications and mathematical approaches, Submitted to *IEEE CAMAD 2012*, Barcelona (2012).

[15] Ch. Ehresmann, *Charles Ehresmann : Oeuvres complètes et commentées* (Ed. A.C. Ehresmann), Amiens (1981-1983).

[16] G. Fauconnier and M. Turner, *The way we think*, Basic Books (Reprint, 2003).

[17] P. Hagmann, L. Cammoun, X. Gigandet, R. Meuli, C.J. Honey, Van J. Wedeen and O. Sporns, Mapping the Structural Core of Human Cerebral Cortex, *PLoS Biology* 6, Issue 7, 1479-1493 (2008).

[18] D.O. Hebb, *The organization of behaviour*, Wiley, New York (1949).

[19] D.H. Hubel and T.N. Wiesel, Receptive fields..., *J. Physio*. 160 (1), 106-154 (1962).

[20] E. Husserl, *Leçons pour une phénomenologie de la conscience intime du temps*, PUF, Paris (1964).

[21] D.M. Kan, Adjoint Functors, *Trans. Am. Math. Soc.* 89, 294-329 (1958).

[22] F.W. Lawvere, Introduction: Toposes, Algebraic Geometry and Logic, *Lecture Notes in Math.* 274, Springer, 1-12 (1972).

[23] C.S. Pierce, Abduction and induction, in *Philosophical writings of Pierce* (Buchler, J., Ed.), Dover Publications, New York, 150-156 (1903).

[24] R. Rosen, The representation of biological systems from the standpoint of the Theory of Categories, *Bull. Math. Biophys.* 20, 245-260 (1958).

[25] M.P. van den Heuvels & O. Sporns, Rich-Club Organization of the Human Connectome, *J. Neurosci*. 31(44), 15775–15786 (2011).

Representation, Analytic Pragmatism and AI

Raffaela Giovagnoli[1]

Abstract: Our contribution aims at individuating a valid philosophical strategy for a fruitful confrontation between human and artificial representation. The ground for this theoretical option resides in the necessity to find a solution that overcomes, on the one side, strong AI (i.e. Haugeland) and, on the other side, the view that rules out AI as explanation of human capacities (i.e. Dreyfus). We try to argue for Analytic Pragmatism (AP) as a valid strategy to present arguments for a form of weak AI and to explain a notion of representation common to human and artificial agents.

1. Representation in AI

The notion of "representation" is at the basis of a lively debate that crosses philosophy and artificial intelligence. This is because the comparison starts from the analysis of "mental representations". First, we move by adopting a fruitful distinction between the "symbolic" and the "connectionist" paradigms in AI [1]. This distinction is useful to highlight two different ways of explaining the notion of representation in AI.

An important challenge for AI is to simulate not only the "phonemic" and "syntactic" aspects of mental representation but also the "semantic" aspect. Traditionally, philosophers use the notion of "intentionality" to describe the representational nature of mental states namely intentional states are those that "represent" something, because mind is directed toward objects. The challenge for AI is therefore to approximate to human representations i.e. to the semantic content of human mental states. If we think that representation means to connect a symbol to the object of representation we focus on the discreteness of mental representations. On the contrary, it could be plausible to focus on the inter-relation of mental representations. The first corresponds to the symbolic paradigm in AI, according to which mental representations are symbols. The second corresponds to connectionism in AI, according to which mental representations are distributed patterns [2].

The task to consider the similarity between human and artificial representation could involve the risk of skepticism about the possibility of "computing" this mental capacity. If we consider computationalism as defined in purely abstract syntactic terms then we are tempted to abandon it because human representation involves "real world constrains". But, a new view of computationalism could be introduced that takes into consideration the limits of the classical notion and aims at providing a concrete, embodied, interactive and intentional foundation for a more realistic theory of mind [3].

We would like to highlight also an important and recent debate on "digital representation"[4] that focuses on the nature of representations in the computational theory of mind (or computationalism). The starting point is the nature of mental representations, and, particularly, if they are "material". There are authors who maintain that mental representation are material [5] others thing that thought processes use conventional linguistic symbols [6]. The question of digital representation involves the "problem of physical computation [7] as well as the necessity of the notion of representation [8] so that we only have the problem of how to intend the very notion of representation [9]. But, there is also the possibility of understanding computation as a purely syntactic procedure or to include "every natural process" in a "computing universe" [10].

2. What is AP?

The core point of Brandom's original book *Between Saying and Doing* [11] is to describe discursive practices and to introduce norms for deploying an autonomous vocabulary namely a vocabulary of a social practice (science, religion etc.). These norms are logical and are at the basis of an "inferential" notion of representation. But, inference in this sense, recalling Frege, is material [12]. Brandom refuses the explanation of representation in terms of syntactical operations as presented by "functionalism" in "strong" artificial intelligence (AI). He does not even accept weak AI (Searle), rather he aims to present a "logical functionalism" characterizing his analytic pragmatism (AP) [13]. Even though Brandom uses his account of representation to refuse computationalism, his

[1] Pontifical Lateran University

pragmatism is different from the Dreyfus's one, which rests on a non-linguistic know-how (logically and artificially not computable). According to Brandom, we are not only creatures who possess abilities such as to respond to environmental stimuli we share with thermostats and parrots but also "conceptual creatures" i.e. we are logical creatures in a peculiar way.

First, we introduce "practice-vocabulary sufficiency" or "PV-sufficiency" which obtains when exercising a specific set of abilities is sufficient for someone to count as deploying a specified vocabulary [14]. These are for instance "the ability to mean red by the word red" or "the capacity to refer to electrons by the word electrons" (Brandom includes even *intentions* to refer). Together with these basic abilities we must consider the relationship between these and the vocabulary in which we specify them. A second basic meaning-use relation is the "vocabulary-practice sufficiency" or just "VP-sufficiency" namely the relation that holds between a vocabulary and a set of practices-or-abilities when that vocabulary is sufficient to specify those practices-or-abilities.

In order to deploy any autonomous vocabulary we must consider the necessity of certain discursive practices defined as "asserting" and "inferring" that, according to Brandom, rule out computationalism [15]. According to the PV-necessity thesis, there are two abilities that must be had by any system that can deploy an autonomous vocabulary: the ability to respond differentially to some sentence-tokenings as expressing claims the system is disposed to *assert* and the ability to respond differentially to moves relating one set of such sentence-tokenings to another as *inferences* the system is disposed to *endorse*. By hypothesis, the system has the ability to respond differentially to the inference from p (premise) to q (conclusion) by accepting or rejecting it. It also must have the ability to produce tokenings of p and q in the form of asserting.

3. Why AP could be a fruitful strategy to simulate representation?

In this conclusive session I'll try to show that the notion of representation described in AP terms presents aspects that are common to human and artificial intelligence.

The PV- and VP-sufficiency thesis suggest that basic practices can be computationally implemented and this description corresponds to the Brandomian interpretation of the Turing test and, consequently, to the refusal of a classical symbolic interpretation in AI (GOFAI) of the notion of human representation. Brandom introduces a pragmatic conception of artificial intelligence or "pragmatic AI" which means that any practice-or-ability P can be decomposed (pragmatically analyzed) into a set of primitive practices-or-abilities such that:

1. they are PP-sufficient for P, in the sense that P can be algorithmically elaborated from them (that is, that a*ll* you need in principle to be able to engage in or exercise P is to be able to engage in those abilities plus the algorithmic elaborative abilities, when these are all integrated as specified by some algorithm); and
2. one could have the capacity to engage or exercise *each* of those primitive practices-or-abilities without having the capacity to engage in or exercise the target practice-or-ability P [16].

For instance, the capacity to do long division is "substantively" algorithmically decomposable into the primitive capacities to do multiplication and subtraction. Namely, we can learn how to do multiplication and subtraction without yet having learning division.

On the contrary, the capacities to differentially respond to colors are not algorithmically decomposable into more basic capacities. This observation entails that there are human but also animal capacities that represent a challenge for strong AI (GOFAI), but nowadays not for new forms of computationalism. Starting from Sellars, we can call them *reliable differential capacities to respond to environmental stimuli* [17] but these capacities are common to humans, parrots and thermostats so that they do not need a notion of representation as symbol manipulation.

Along the line introduced by Sellars, Brandom intends the notion of representation in an "inferential" sense. It is grounded on the notion of "counterfactual robustness" that is bound to the so-called frame problem [18]. It is a cognitive skill namely the capacity to "ignore" factors that are not relevant for fruitful inferences. The problem for AI is not *how* to ignore but *what* to ignore. In Brandom's words: "Since non-linguistic creatures have no semantic, cognitive, or practical access at all to most of the complex relational properties they would have to distinguish to assess the goodness of many material inferences, there is no reason at all to expect that that sophisticated ability to distinguish ranges of counterfactual robustness involving them could be algorithmically elaborated from sorts of

abilities those creatures do have" [19]. Nevertheless, we could start by studying what "intelligence" really is by starting from the simplest cases.

Brandom introduces the notion of "counterfactual robustness" to overcome strong GOFAI, to avoid the primacy of prelinguistic background capacities and skills in weak AI (Searle) and phenomenology (Dreyfus). The notion of representation he introduces could work only if we embrace a peculiar form of inferentialism. Differently, we could read AP to analyze inferential capacities that are connected with logical laws common to human and artificial agents [20].

REFERENCES

[1] M. Carter, *Minds and Computers*, Edimburgh University Press, Edimburgh, 2007.

[2] Carter (2007), chap. 18.

[3] M. Scheutz (ed.), *Computationalism. New Directions*, MIT, 2002.

[4] V.C. Müller, *Representation in Digital Systems*, in *Current Issues in Computing and Philosophy*, A. Briggle et. Al. (Eds), IOS Press, 2008.

[5] A. Clark, 'Material Symbols', *Philosophical Psychology* 19 (2006), 291-307; S. Sedivy, 'Minds: Contents without vehicles', *Philosophical Psychology* 17 (2004), 149-179.

[6] J. Speaks, 'Is mental content prior to linguistic meaning*?'*, *Nous* 40 (2006), 428-467.

[7] O. Shagrir, 'Why we view the brain as a computer', *Synthese* 153 (2006), 393-416.

[8] J.A. Fodor, 'The mind-body problem', *Scientific American* 244 (1981), 114-123.

[9] G. O'Brien, 'Connectionism, analogicity and mental content', *Acta Analitica* 22 (1998), 111-131.

[10] G. Dodig-Crnkovic, 'Epistemology naturalized: The info-computationalist approach', *APA Newsletter on Philosophy and Computers* 6 (2007), 9-14.

[11] R. Brandom , *Between Saying and Doing*, Oxford University Press, Oxford.

[12] R. Brandom, *Making It Explicit*, Cambridge University press, Cambridge, 1994, chap. 2; R. Giovagnoli, *Razionalità espressiva. Scorekeeping: inferenzialismo, pratiche sociali e autonomia*, Mimesis, Milano, 2004; R. Giovagnoli (ed.), 'Prelinguistic Practices, Social Ontology and Semantics', *Etica & Politica/Ethics & Politics*, vol. XI, n. 1, 2009.

[13] Brandom (2008), chap. 2, chap. 3.

[14] Brandom (2008), pp. 74-77.

[15] Brandom (2008), pp. 77-83.

[16] Brandom (2008), pp. 82-83.

[17] W. Sellars, *Empiricism and the Philosophy of Mind*, Harvard University Press, Cambridge, 1957, 1997.

[18] Brandom (2008), p. 79.

[19], Brandom (2008), p. 83.

[20] H. Boley, *Semantic* 'Web. Knowledge Representation and Inferencing', 2010, http://www.cs.unb.ca/~boley/talks/DistriSemWeb.ppt. ; Carter (2007); R. Evans, 'The Logical Form of Status-Function Declarations' in Giovagnoli (ed.) (2009); R. Evans, 'Introducing Exclusion Logic as Deontic Logic' in Lecture Notes in Computer Science, vol. 6181/2010; R. Giovagnoli, 'Osservazioni sul concetto di "pratica autonoma discorsiva" in Robert Brandom', in *Etica & Politica/Ethics and Politics*, IX, 1, 2008, pp. 223-235; R. Giovagnoli, 'On Brandom's "Logical Functionalism"', *The Reasoner*, 4 (3), (2010), www.thereasoner.org.

Salient Features and Snapshots in Time: an interdisciplinary perspective on object representation

Veronica Arriola-Rios[1] and **Zoe P. Demery**[2]

Abstract. Faced with a vast, dynamic environment, some animals and robots often need to acquire and segregate information about objects. The form of their internal representation depends on how the information is utilised. Sometimes it should be compressed and abstracted from the original, often complex, sensory information, so it can be efficiently stored and manipulated, for deriving interpretations, causal relationships, functions or affordances. We discuss how salient features of objects can be used to generate compact representations, later allowing for relatively accurate reconstructions and reasoning. Particular moments in the course of an object-related process can be selected and stored as 'key frames'. Specifically, we consider the problem of representing and reasoning about a deformable object from the viewpoint of both an artificial and a natural agent.

1 INTRODUCTION

The cognitive architecture of any animal or machine (jointly 'agents') has limits, so it cannot contain a perfect model of the dynamic external and internal world, such as about all matter, processes, affordances, or more abstract concepts, like 'mind' or 'spirit'. Every agent receives a particular amount of data through its sensors. How useful that data is in the short or long term depends on the environmental conditions, how accurately the data might be processed into information, and the agent's behavioural response. Frequently, an agent should maximise the amount of meaningful, relevant information it can obtain about its surroundings, while minimising the energy expended, but this is highly dependent on the nature of the agent [4]. This applies not just to a static snapshot of time, but also to a constantly changing world with a past, present and future, where being able to predict events, or select between alternative actions without actually trying them, may be useful for the agent. So in these circumstances, what are the most useful elements for the agent to store and process in its cognitive architecture and how may they best be coded? Principally, we propose that when an agent gathers information through its senses, often it may form object representations supported by exploration[3].

To date in the field of animal cognition (AC), there has been surprisingly little systematic, quantitative research on exploration, and how it could support learning mechanisms in different agents (see [28] for more discussion). What research there is, has largely been on humans and focussed on Bayesian network learning (e.g. [20]). Among the non-human animal researchers, the focus has been on *what* the different cognitive capacities of different species are, rather than *how* they actually process information to achieve those capacities [25]. For example, the 'trap-tube task' is a typical litmus test for causal understanding of gravity (e.g. [26]). It has revealed a lot about many species, but it is just a binary measure of whether an individual can complete the task or not. No one has fully investigated why one individual can succeed at the task, while another fails – is it something about their different exploratory strategies? Moreover, although quite complex-looking actions can often be performed by agents with simple mechanisms and small neural architectures (e.g. [11]), they may not be able to *generalise* these actions to other similar, but novel circumstances. Thus in this paper, we are concerned with more complex, flexible agents. Another area consistently ignored in AC, but one which may provide answers, is how the senses support exploratory learning (e.g. [7]).

It is a blossoming area in Artificial Intelligence (AI) however. Robots force us to explicitly define the model design, suggesting concrete, testable hypotheses for AC. However, we believe there is not yet a robot/simulation that can flexibly abstract concepts, or generalise knowledge to new situations. AI has looked at different learning mechanisms in isolation with relative success, but few projects have tried combining them into one agent (e.g. [10]). Therefore, AC behavioural experiments can provide realistic biological constraints and inspire more integrative cognitive modelling in AI.

We would like to propose that when exploration of objects occurs for forming representations, it is not always random, but also *structured*, *selective* and *sensitive* to particular features and salient categorical stimuli of the environment. Also that it can follow through three stages of theory formation – the forming, the testing and the refining of hypotheses [6]. Each hypothesis may be specific to a particular group of affordances or processes ('exploratory domains'), but they may also be generalisable to novel contexts. We introduce how studies into artificial agents and into natural agents are complementary [6], by comparing some findings from each field.

First, we will take a top-down approach to explore what some of the general environmental constraints imposed on an agent's system when internalising the world around it may be. Then we will look at some of the possible mechanisms to solve these problems, particularly in the visual domain of object representation. There are several methodological problems in computer vision research, including recognition, tracking and mental imagery [14]. Within robotics, we present an approach where simulations of real objects, calibrated from real-time environmental data, can be used as artificial mental imagery. We have exploited a combination of key features from im-

[1] School of Computer Science, University of Birmingham, Edgbaston, Birmingham, B15 2TT, UK; email: v.arriola-rios@cs.bham.ac.uk
[2] School of Biosciences, B15 2TT; email: zxd878@bham.ac.uk
[3] Cognition does not always rely on internal representations and the degree of detail in any internal representation can vary greatly depending on the situation. For instance, there can be a lack of detail especially when the environment can largely control an agent's behaviour, such as in flocking behaviour or in using pheromone trails. Here alternative, but complementary, mechanisms may be more relevant, such as emergency or embodiment, but in this paper we will not consider these cases[3].

age analysis, computer graphics and animation, as well as aspects of physical models, to generate an internal representation of a deformable object with predictive capabilities. Finally, we will consider the degree of ecological validity of this model by comparing it with AC behavioural findings about parrots, who are notoriously exploratory and playful throughout their lives.

2 REQUIREMENTS FOR THE AGENT-ENVIRONMENT INTERACTION

An agent interacting with its surrounding environment often combines perception and analysis with action. It can also be driven by its goals, which can be quite explicit, like foraging for survival, or particular problem-solving tasks. Or they can be quite implicit, such as to gather information by apparently random exploratory behaviour. Shaw [21] suggests, *"The chief end of an intelligent agent is to understand the world around it."* Here, the word 'understanding' implies the agent's ability to make predictions about the world. For this to take place, the agent should be able to detect the consistent properties or salient features in its environmental niche. These properties allow a link to form between the agent and the environment. We will now consider what some of these primordial properties might be (see also [5]).

2.1 Redundancy

Given the inherent limitations of the agent, it will only be possible for it to gain a partial understanding of its surroundings[4]. This partial understanding may not allow the agent to make perfect predictions for all environmental events, so it cannot always be ready to process useful information. As it detects sensory data, it also may not succeed at processing relevant signals. Therefore, we expect there may be errors and inexactitudes at different levels of the agent's perceptual or analytical processes. It may thus be useful for its system to be able to tolerate this margin-of-error. Some agents often have more than one mechanism to find things, solve problems, or to perform actions. The agent could just react according to different layers of data filtering, or it could use one or a combination of different learning mechanisms [6]. While qualitatively different, all of these mechanisms produce similar, valid results. In this sense, we call these different possible mechanisms 'valid ways', and say they are 'redundant'. Therefore, redundancy allows the agent to ignore irrelevant data, or to reconstruct faulty perceptions from new perceptions that convey the same information.

2.2 Consistency

When multiple methods are used to collect or analyse data, they can act in a complementary way, and contribute by providing different information. Alternatively, they may be superfluous; in which case, they confirm previous findings. For an agent, different methods of perceiving the same thing should be consistent with each other, if there is enough knowledge. An agent that sees a pen while touching it, should gain tactile information in accordance with the position and surface of the image it sees. If there is a fault in the synchronisation between this visual and tactile information, the agent will not be able to properly integrate this information, or accurately describe the object. This principle is present in human mathematics: different methods used to solve the same equation, *must give the same answer*.

[4] An artificial agent (e.g. a virtual automaton) in a very simple environment can make perfect predictions; but we are not concerned with these cases.

2.3 Persistency

For an agent to be able to make relatively accurate predictions about the environment, there should be at least a few unchanging rules in the environment for a significant period of time. These rules are useful for the agent's internal knowledge and learning mechanisms. The strongest examples can be found in mathematics and physics. In order to develop the cosmological theories of physics, it is necessary to assume that the physical laws that rule at the present time on planet Earth, are the same rules that applied during the Big Bang and in galaxies far beyond ours. Agents should respond in the same way to the environment. During complex actions, an agent may change or modify their goals and plans. Even then though, they should make the changes according to a particular, foreseeable pattern, which may be rooted, for example, in their brain structure. If agents do not follow persistent rules, their behaviour is erratic and unpredictable.

2.4 Regularity

This is the predictable presence of previously perceived features or classes of them, due to a fixed relationship between occurrences[5]. There should be persistent patterns in the environment, allowing at least for partial predictions, particularly when an agent is faced with different causal problems. Causality is a manifestation of regularity, where the partaking elements are not always identifiable, but whose manifestation always entails the same consequence. Thus, agents should have mechanisms capable of detecting these patterns to take advantage of them. Then the environment could be categorised using a finite amount of key features linked by predictive relationships, including elements representing continuous features. For example, a small mathematical equation can describe an infinite parabola.

2.4.1 Sequentiality

This is a particular form of regularity, but in a universe with only one temporal dimension, it becomes especially relevant. Sequentiality is the presence of a series of features of two or more elements that are nearly always perceived in the same total or partial order[6]. The first features can be used to identify the sequence and predict either the following features, or the rules set needed to process them. Some examples include: identify a command and know which actions to execute; analyse the first items of a numerical sequence and predict the next; listen to the first notes of a song and remember how to sing the rest (which was memorised in advance); identify the beginning of a question and prepare to understand it to look for the answer; or listen to the sound of a prey and prepare to chase.

2.4.2 Structure: partial order and layers

There could also be a succession of *sub*-sequences. The connections here would only allow a few options to follow, such as beginnings of other sub-sequences. This forms a branching structure, which becomes layered, modular, and, in some cases, hierarchical [1]. The maximum length of an existing sequence, and the maximum number of branches that can be remembered and manipulated, impose strict limitations upon what the agent can understand, and the types of patterns it is capable of detecting. However, this structure may allow more complex agents to make abstractions, as concepts formed at one stage could be re-used and refined to repeatedly form ever more

[5] This can be present in different dimensions, or in a hierarchical structure.

[6] These may not be contiguous and can include cause-and-effect learning.

complex concepts in multiple ways [4]. This allows for progressively specific and parallel processes (e.g. [9]).

2.5 Experience

For small and well-identified tasks, a largely pre-programmed agent may suffice. Little experience may be needed in a relatively static environment, such as where precocial animals, whose behaviour has been almost completely determined by their genome, just need to survive long enough to reproduce. Other agents are often required to adapt to diverse, dynamic environments, where a lot more learning is required (see [4] for greater discussion). The different extractions of relevant information (Section 2.1) would more likely be processed by mechanisms shaped and influenced by experience. The agent should seek out information to reinforce, evolve and, when possible, prove or disprove its current models, particularly if its expectations are violated. Depending on the needs and the competences of the agent, a specific, relevant subset of experiences would allow specific, relevant features of the individual's niche to be captured (e.g. [27]). We believe there is continual extension of these 'branches' or 'blocks of knowledge' throughout the life of a cognitive agent. At different ages or stages of development, an agent should take in different aspects of the same overheard conversation, for instance, or different aspects of the operation of the same tool.

2.6 Where does this leave us?

All of the above described environmental features/constraints together form a structured universe. Parts of this structure may be perceived and understood by artificial and natural agents. The existence of regularities reduces the information needed to describe a part of the environment, as once enough elements and relationships have been identified, the rest can be inferred. Some animals may have the ability to identify 'valid ways' and describe them as 'formalisms'; sets of rules that can warrant good results when sufficient conditions are met [6]. This is essentially how science operates, particularly logic, mathematics and computer science.

Within the field of AI, some formalisms for 'knowledge representation' focus on the association of symbols to entities (i.e. objects, relationships and processes) in a structured way, such as 'Frame Languages' [15]. However others, like 'First Order Logic', incorporate powerful systems of deduction. These symbolic languages are extremely powerful for discrete reasoning, but they may not be particularly appropriate for describing continuous dynamics, or even for making predictions, such as when objects move through an environment. In AI, it is highly relevant to consider the amount and type of knowledge needed before an agent can be capable of processing it. How much does the agent need to know to be able to predict a few movements of different objects? Can that knowledge be learned from experience, or does it need to be programmed beforehand?

In certain contexts, the minimum number of necessary elements to complete a description is known as the number of degrees of freedom. For example, given the generic mathematical formula that describe parabolae, only three points are needed to specify a single, infinite parabola. This principle can be directly applied in computer graphics. By making use of algebraic equations, an infinite amount of shapes can be approximated, represented and reconstructed with just a few polynomials [16]. Furthermore, transformations of these shapes can be encoded with mathematical formulae, thus allowing the representation of physical objects and processes; which can be used to implement a form of mental imagery.

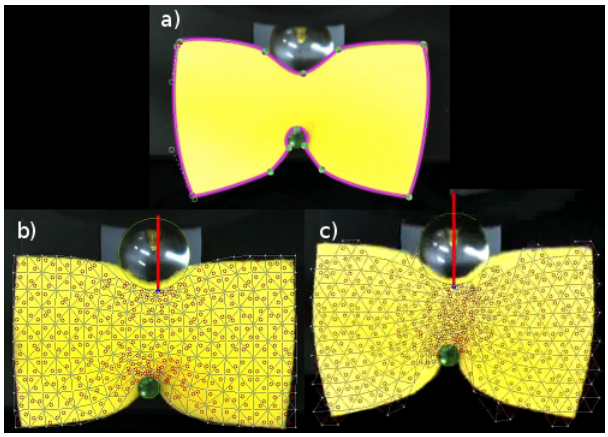


Figure 1. Top view of an experiment where a robotic finger (sphere at the top) pushes a sponge, perpendicular to its widest axis, against a pencil that serves as an obstacle (green cap). **a)** The contour of a deformed sponge approximated by a series of splines, with the control points placed by a human. **b)** The sponge represented by a rectangular mesh, generated in the first frame before deformation; the mesh configuration was predicted by the physics model. **c)** Hexagonal mesh, similar to (b).

Hence, whether the powerful deductive machinery is available in a natural or an artificial agent, it is important to define how we go from representations of continuous transformations, to discrete objects and events. As with the popular phrase, 'a picture is worth a thousand words', predicate logic may not be able to naturally represent 3D graphical information in a consistent, complete and compact description. It may be possible, however, to extract logical information from graphical simulations when required for symbolic reasoning. Here we give an example of how this could be achieved in AI by combining traditional animation techniques, computer graphics and physics, with symbolic representations.

We believe this approach may be more rigorous than the standard mechanism used in human brains. Humans can recognise things without being able to draw them [2], or use mental imagery without making exact simulations [14] (while our AI system requires them). This shows how we need to better understand the underlying mechanisms of natural agents processing and representing the world around them. Observations of natural exploration behaviour do provide realistic biological constraints on the design of AI models for object representation. We will investigate these issues in AC by running behavioural experiments on parrots, as our exemplar exploratory and adaptive species. Is there evidence of each of the environmental requirements/regularities described above being attended to by the parrots? Does their exploration behaviour suggest underlying strategies for processing and representing the environment?

3 DESIGNING A REPRESENTATION

3.1 Using key frames to model deformable objects

The study of the perception and understanding of the affordances of deformable objects is particularly appropriate to illustrate the points outlined in the section above. The problem of representing solid objects, their properties and their related processes has been studied in great detail in computer graphics [8], and there has been attempts to generate representations using semantic information [22]. Within the first field, there are several good representations for many different

types of shapes, most of them based on meshes, splines or quadrics [16]. The motion of objects is simulated with an approach analogous to traditional cartoon animations. There is a set of key frames, where a 'key frame' is a drawn snapshot in time defining the start and end points of any smooth transition, and all of the frames connecting them are called the 'inbetweeners'.

Currently, key frames are identified and drawn by humans; in traditional animation the most skilled cartoonists are responsible for them. Due to the smoothness of the transition between key frames, it is possible for a less-skilled cartoonist to interpolate the inbetweeners. In computer animation, the control points and curves defining the geometry and colours of the scene are set in the key frames. The transitions between key frames are mainly polynomial interpolations, or continuous mathematical transformations of these control elements [18]. To create realistic animations, movements are often captured from real objects. This is a very slow and expensive process [13]. In an attempt to automate the rendering of realistic movements and the inbetweeners, physics engines have been incorporated into the animation packages. They are also present in real-time virtual environments where interaction with a user takes place.

However, the incorporation of physics changes the dynamics of producing an animation slightly. Instead of interpolating between two key frames, the first key frame is given by a human designer and the simulation stops when a given condition is satisfied, thus automatically generating the inbetweeners and the final key frame. Note that predictive capabilities have been attained, and that the simulation is now required to specify the new parameters of the material. This includes mass, young coefficient and spring stiffness, in addition to the method's criteria, such as integration methods or time steps. Correctly estimating these parameters is a difficult problem.

Furthermore, while the simulations may look plausible to the human eye, they may not be physically accurate, so different models are required to simulate different materials and differently shaped objects. A natural agent's brain faces a similar computational problem, yet evolution largely seems to have solved it in a *qualitatively* different way. Humans can reason and make predictions about features of the world, but we probably do not simulate it in the quantitative way a physics engine does. It is still not completely clear how or what exactly the underlying mechanism is in various natural agents. Behavioural experiments can allow us to *infer* what is going on in an animal's mind. However, interpretation of the data is largely based on assumptions and only allows us to make indirect conclusions. Invasive techniques, such as particular neurophysiological or brain imaging methods, only provide partial information about the *content*, or even about the structure or neural representations, in an animal's mind. Thus, if done correctly, AI simulations can be very illuminating. We suggest that an initial list of problems an artificial agent needs to solve are:

1. Generate an internal representation of real deformable objects in the surrounding environment;
2. Identify key frames of the related environmental processes;
3. Interpolate (continuously or discretely) the links between frames;
4. Use previous knowledge to predict future events.

The automation of the animation process provides one solution for the first three points. Traditional animation techniques, however, cannot address the fourth point. The use of physics models and formal logics can address the two last points, but in this case the agent needs to select and calibrate the right model. It is still debatable whether physics models can correctly approximate all the ranges of processes observed in natural environments, given the inherent limitations of mathematical models to model real, complex deformations. Furthermore, there is still no model that integrates all of the points into one agent. Given the huge variety of possible affordances perceived by humans alone, we expect that some form of learning should be used to generate the model(s), which would provide the interpolating link between key frames and aid in making predictions. However, *which* type of learning mechanism is still open to question.

Here we present the advances of a preliminary, physics-based method, where a general (though not completely accurate) model of deformable objects is used [24], and an artificial agent learns to calibrate and use it in the way described above. The next step is to take the key frame representation of the object and extract symbolic ones from it. Then we need to take functions that describe the transformations, associate a symbol to it, and consider that symbol as referring to a categorised process or action. For several cases, this step should be quite straightforward, since the representation has already been discretised, grounded and categorised [23]. Then the already developed, symbolic-level machinery can be applied and the results compared with the natural exploration behaviour (e.g. of parrots) for ecological validity. Is there evidence of similar mechanisms in natural systems? Is our model biologically plausible?

3.2 Representing the object's shape

3.2.1 Kakariki Experiment I: AC implications for AI models

When segregating the world around itself, we believe an agent first needs to identify and represent distinct objects. Then the agent needs to understand what the shape of each object means, i.e. its affordances when it interacts with the rest of the world. What are its physical properties? For instance, if two key elements are connected by a known relationship, anything in between is already implicitly represented. Contact points and segments of continuous curves can be approximated by lines and polynomials, and delimited by key points. Under this light, it is natural that an agent would be more interested in these points of discontinuity. Indeed, in our first AC experiment, we found that this does seem to be the case, at least for the New Zealand red-fronted parakeet or 'kakariki' (*Cyanoramphus novaezelandiae*).

We chose kakariki as our model animal species for investigating how agents gather and represent environmental information, as they are neophilic and have a high exploratory tendency throughout their lives. Moreover, as with many other parrot species, they are relatively intelligent and have an anatomy adapted to dexterity and fine object manipulation. We presented a series of novel objects of a range of different rigid shapes to 21 kakariki individually and recorded their exploratory behaviour in detail over a 25-minute period. They spent most of the time exploring the corners and indents of the objects, then areas of high curvature second, over smooth surfaces. We would like to suggest this may be because corners and areas of high curvature are more likely to cue useful properties/affordances about different objects.

Related to this finding, it is interesting to consider in AI how Shaw uses information theory to apply the principle of maximising information and predictive capabilities to an image analysis task, and the first result he finds is an edge detector [21]. Similarly, related to the AC finding on relative importance of areas of high curvature, Ravishankar [19] found that it is easier for an artificial agent to recognise deformed objects by placing emphasis on the bending around points of high curvature. It is further compelling that a mathematical function is segmented where there is a discontinuity. Thus, in one dimension, corners are discontinuities of the derivative of functions;

in two dimensions, edges are discontinuities of derivatives of functions; while points of high curvature (maxima, minima and inflexion points) are points where the first derivative is zero. It would seem that the same issues that mathematicians deem interesting are playing a major role in both natural and artificial agents, as features for object segmentation, categorisation, tracking and, possibly, prediction. Therefore the use of mathematical curves to approximate deformable objects is highly illustrative.

3.2.2 AI Model I: modelling the sponge

In one dimension, a way of approximating a continuous curve is by a succession of lines. In two or more dimensions, shapes can be approximated by **meshes**, where each element is 'flat' and defined by nodes and edges. Triangular and hexagonal meshes are widely used. Alternatively, quadrics and polynomials of two or three degrees can be used. They are flexible enough for representing most humanly distinguishable continuous shapes. Polynomials have been used to form **splines**, which are defined by a small set of control points. They can be used to interpolate as much detail of a shape as desired, since the polynomials are continuous, while the connections between them can be discontinuous (e.g. [16]). This is why we considered meshes and splines for our model.

As an experimental example, our model analysed the process of deforming a sponge. In general, compliant materials have the potential to change their shape in nearly an infinite amount of unpredictable ways, therefore understanding deformable objects poses a particularly interesting challenge for both artificial and natural agents. Unlike rigid objects, it is not possible to know in advance all of the elements required for representing the deformation. How many splines or elements in a mesh is required, or what are its degrees of freedom? For some specific objects under controlled circumstances, these possibilities can be restricted, as in medical research with human organs [12]. However, an agent that interacts with an environment populated with unrestricted deformable objects, requires a more general solution. One approach is to automatically generate a hierarchical mesh to represent a few objects in a virtual environment, which adapts as an object deforms [17]. However, this has not yet been directly tried in robotics, where an internal representation needs to match objects in the external environment. This continues to remain an open question even in the AC literature – what would the agent do if an object becomes deformed to a shape unforeseen by the initial representation?

As a tentative first step towards solving this problem, we looked at modelling a spheric robotic finger pushing against a sponge. Please note we are not claiming this model replicates animal vision or reasoning, but it may provide a building block from which to work from. The movement of the robotic finger was blocked by a pencil directly opposite. The finger performed a continuous movement against the sponge, while a camera and a force sensor registered the interaction. **Figure 1** illustrates the use of splines and meshes to approximate the contour and surface of the sponge as it became deformed.

3.3 Representing the related processes

3.3.1 Kakariki Experiment II: more implications for models

Once the agent can generate a representation of any object shape it may detect, we believe the next step is for it to understand the related physical processes in the environment. It should identify the key elements and unite them with appropriate functions. How does the object become deformed when interacted with in the world?

We first considered this in the natural dimension in a second AC behavioural experiment. We presented the same kakariki with five cubes of different deformabilities in a random order over five trials over different days. As in the previous experiment, in each trial we allowed them 25 minutes to interact with the objects as they chose and recorded their exploration behaviour in detail.

As we predicted in [6], they initially explored the two extremes the most (i.e. the most rigid and the most deformable cube), but their exploratory 'focus' or 'strategy' changed. So in the second and third trial, the cube of the 'median' or intermediate deformability was explored significantly more than all of the other cubes. Then in the final two trials, the cubes the next interval along (i.e. the second-most deformable cube and the second-most rigid cube) became more of a focus for the kakariki's exploration. In conclusion, the exploration strategy seems to change with time, perhaps as more experience and progressively more specific knowledge is gained about the deformability of objects and different object categories. We would like to suggest that the kakariki had a exploration strategy that allowed them to gain more information about the *process* of deforming an object.

3.3.2 AI Model II: modelling the deformation

Thus as a preliminary, tentative step, we next wanted to consider what the design of this internal strategy/learning mechanism might be. In the AI example of deforming the sponge, the following key frames can be identified:

1. **The finger starts moving.** At this point the force sensor detects only some noise, but the command to move has been given and the vision (camera) begins to detect changes between frames, i.e. that the position of the finger is changing. Thus, the first key frame would contain the finger separated from the sponge and the pencil.
2. **The finger touches the sponge.** At this point the force sensor detects an abrupt increase in one direction. Visually, collision detection routines begin to detect a contact between the circle (i.e. the finger), and one or two triangles in the mesh (i.e. the sponge).
3. **The finger stops moving.** No more changes are detected.

Notice that these coarse key frames are the frames where things change in a very noticeable manner. It is possible to connect frames 1 and 2 by using a function that describes the simple linear translation of the circle (finger). Between frames 2 and 3, the same translation function applies to the circle, but also the physics model gets activated to deform the mesh (sponge) as the circle pushes it. These two functions can predictively describe the observed movements. At frame 3, no function or model is required anymore, because the execution of the command is over and there is no more movement. The scene has ended. From this perspective, the whole process/action can productively be segmented into smaller actions. The internal representation of each frame can be formed by tracing back the activation and deactivation of the required *mechanisms*. Now each segment can be re-represented by a single symbol. The whole sequence can be described as something like: displace finger; push sponge; stop. The agent can then choose between thinking of the command it executed (e.g. translate), or the changes in the sponge (detected through vision or touch), or combinations of both.

There are precedents to doing this type of segmentation, such as in the work by [22]. Here the agent, *Abigail*, analyses a simple circle-and-sticks simulation of ping-pong. Even for this highly simplified world, it was not trivial to unequivocally detect the points of discontinuity that establish the beginning and end of an action. However, Siskind was not quite using our concept of segmentation in

modelling, which is just an extension of the idea of a polynomial connecting two control points. Even though the use of splines to approximate curves is a widely used technique, there is not a general technique that can automatically generate a spline from scratch to approximate any curve. It is a brand new research field; to investigate the use of models for interpolation between frames, segmenting and understanding actions.

4 CONCLUSION

By studying both artificial and natural agents, we can provide a fuller account of how, when necessary, an individual can efficiently represent objects and their related processes in the environment from the huge number of sensory signals they receive. In this light, we can also consider what the requirements posed by the external environment may be upon the finite brain of the agent. Thus, we have briefly discussed two behavioural experiments on parrot exploration of novel objects to give us an insight into what the biological constraints might be on an AI model for representing deformable objects. In considering natural behaviour and the possible underlying exploration strategies for gathering information, we have described how a selection of key elements from the environment could be used as a basis for an object representation. These key elements are connected through functions, which indicate how to obtain the value of other points. The same mechanism could be used to represent processes and actions, by identifying key frames, and finding the correct physics model to interpolate between frames. It is possible to segment a complex interaction between the agent and the environment into individual actions, by detecting: the commands given; discontinuities in the sensory signals; and the intervals of application of each mechanism. Each of these individual actions could then be represented by symbols. These symbols are grounded in the environment through the selected key elements. It is straightforward to use these symbols for traditional problem-solving tasks, as in [1]. We have further provided evidence that natural agents seem to similarly focus their exploration behaviour on key environmental elements, such as corners, edges and areas of high curvature. Likewise, at least with parrots, individuals seem to attend first to extreme exemplars of particular object properties, including deformability/rigidity, but this exploration strategy becomes gradually refined with time. However, we cannot yet confirm if this parrot exploration is due to similar underlying mechanisms as those presented in our AI model. In conclusion, we have presented an interesting *preliminary* analysis of some of the forms of object representation that may be useful to intelligent natural agents in certain contexts, and demonstrated these capabilities in working computer models.

ACKNOWLEDGEMENTS

We would like to thank our supervisors, Jackie Chappell, Aaron Sloman and Jeremy Wyatt, for their comments and unrelenting support. Many thanks also to our colleagues in the Intelligent Robotics Lab and the Cognitive Adaptations Research Group for their help throughout this project. This work was made possible by our funders, the CONACYT, Mexico, and the BBSRC, UK.

REFERENCES

[1] Veronica E. Arriola and Jesus Savage Acquisition, 'Knowledge acquisition and automatic generation of rules for the inference machine clips', in *MICAI 2007: Advances in Artificial Intelligence*, volume 4827, pp. 725–735. Springer, (2007).

[2] Barrington Barber, *How to Draw Everything*, Arcturus, 2009.

[3] *Handbook of cognitive science: An embodied approach*, eds., Paco Calvo and Toni Gomila, Elsevier, London, 2008.

[4] J. Chappell and A Sloman, 'Natural and artificial meta-configured altricial information-processing systems', *International Journal of Unconventional Computing*, **3**(3), 211–239, (2007).

[5] Jackie Chappell, Zoe P Demery, Veronica Arriola-Rios, and Aaron Sloman, 'How to build an information gathering and processing system: Lessons from naturally and artificially intelligent systems.', *Behavioural Processes*, **89**(2), 179–186, (2012).

[6] Zoe Demery, Veronica E. Arriola Rios, Aaron Sloman, Jeremy Wyatt, and Jackie Chappell, 'Construct to understand: learning through exploration', in *Proceedings of the International Symposium on AI-Inspired Biology*, pp. 59–61, (2010).

[7] Zoe P Demery, Jackie Chappell, and Graham R. Martin, 'Vision, touch and object manipulation in Senegal parrots Poicephalus senegalus.', *Proceedings of the Royal Society B*, **278**(1725), 3687–3693, (2011).

[8] Christoph M. Hoffmann and Jaroslaw R. Rossignac, 'A road map to solid modeling', *IEEE Transactions on Visualization and Computer Graphics*, **2**, 3–10, (1996).

[9] Annette Karmiloff-Smith, *Beyond Modularity: a developmental perspective on cognitive science*, MIT Press, Cambridge, MA, 4 edn., 1999.

[10] Henry Markram, 'The blue brain project.', *Nature Reviews Neuroscience*, **7**(2), 153–160, (2006).

[11] J. McCrone, 'Smarter than the average bug', *New Scientist*, **190**(2553), 37–39, (2006).

[12] Tim McInerney and Demetri Terzopoulos, 'Deformable models in medical image analysis: a survey.', *Medical Image Analysis*, **1**(2), 91–108, (1996).

[13] Alberto Menache, *Understanding Motion Capture for Computer Animation*, Elsevier, 2nd edn., 2011.

[14] Marsel Mesulam, 'From sensation to cognition', *Brain*, **121**, 1013–1052, (1998).

[15] Marvin Minsky. A framework for representing knowledge. Memo 306, June 1974.

[16] Johan Montagnat, Hervé Delingette, and Nicholas Ayache, 'A review of deformable surfaces: topology, geometry and deformation', *Image and Vision Computing*, **19**(14), 1023–1040, (2001).

[17] D. Morris and K. Salisbury, 'Automatic preparation, calibration, and simulation of deformable objects', *Computer Methods In Biomechanics And Biomedical Engineering*, **11**(3), 263–279, (2008).

[18] *Computer Animation*, ed., Rick Parent, Morgan Kaufmann, 2009.

[19] Saiprasad Ravishankar, Arpit Jain, and Anurag Mittal, 'Multi-stage contour based detection of deformable objects', *Computer Vision - Eccv, Part I, Proceedings*, **5302**, 483–496, (2008).

[20] LE Schulz, 'God does not play dice: Causal determinism and preschoolers' causal inferences', *Child Development*, (2006).

[21] Jonathan M. Shaw, *Unifiying Perception and Curiosity*, Ph.D. dissertation, University of Rochester, Department of Computer Science, The College Arts and Sciences. Rochester, New York., 2006. Supervised by Professor Dana H. Ballard.

[22] Jeffrey Mark Siskind, 'Grounding language in perception', *Artificial Intelligence Review*, **8**, 371–391, (1994).

[23] L. Steels, 'The symbol grounding problem has been solved. so what's next?', in *Symbols and Embodiment: Debates on Meaning and Cognition*, ed., M. de Vega, chapter 12, Oxford University Press, Oxford, (2008).

[24] Matthias Teschner, Bruno Heidelberg, Matthias Muller, and Markus Gross, 'A versatile and robust model for geometrically complex deformable solids', in *Proceedings of Computer Graphics International (CGI'04)*, pp. 312–319, (2004).

[25] RK Thomas, 'Investigating cognitive abilities in animals: unrealized potential', *Cognitive Brain Research*, **3**(3-4), 157–166, (1996).

[26] E. Visalberghi and M. Tomasello, 'Primate causal understanding in the physical and psychological domains', *Behavioural Processes*, **42**, 189–203, (1997).

[27] Auguste M.P von Bayern, Robert J.P Heathcote, Christian Rutz, and Alex Kacelnik, 'The role of experience in problem solving and innovative tool use in crows.', *Current Biology*, **19**(22), 1965–1968, (2009).

[28] R. W. White, 'Motivation reconsidered: The concept of competence', *Psychological Review*, **66**(5), 297–333, (1959).

Grailog: Mapping Generalized Graphs to Computational Logic

Harold Boley[1]

Abstract. Human intuition is often supported by graph-like knowledge constructs depicting objects as (atomic) nodes and (binary) relationships as directed labeled arcs. Following the AI tradition of simple semantic networks and the Semantic Web use of RDF triple stores, philosophical and domain knowledge could in principle be specified as a single directed labeled graph. However, such graphs cannot directly represent nested structures, non-binary relationships, and relation descriptions; these advanced features require encoded ('contrived') constructs with auxiliary nodes and relationships, which also need to be kept separate from direct ('natural') constructs. Therefore, various extensions of directed labeled graphs have been proposed for knowledge representation, including graph partitionings (possibly interfaced as complex nodes), n-ary relationships as directed labeled hyperarcs, and (hyper)arc labels used as nodes of other (hyper)arcs. Meanwhile, a lot of AI and Semantic Web research and development has gone into extended logics for knowledge representation such as description logics, general modal logics, and higher-order logics. The talk demonstrates how knowledge representation with graphs and logics can be reconciled. It proceeds from simple to extended graphs for logics needed in Philosophy, Cognitive Science, AI, and the Semantic Web. Along with its visual introduction, each graph construct is mapped to its corresponding symbolic logic construct. This has led to the development of the knowledge representation language Grailog as part of the Web-rule industry standard RuleML. By serializing Grailog knowledge in RuleML/XML (http://ruleml.org/#Grailog), it will become interchangeable between Web-based engines for Computational Logic.

[1] University of New Brunswick and National Research Council Canada. Email: Harold.Boley@nrc-cnrc.gc.ca

Toward Turing's A-type unorganised machines in an unconventional substrate: a dynamic representation in compartmentalised excitable chemical media

Larry Bull[1], **Julian Holley**[1], **Ben De Lacy Costello**[1] **and Andrew Adamatzky**[1]

Abstract. Turing presented a general representation scheme by which to achieve artificial intelligence – unorganised machines. Significantly, these were a form of discrete dynamical system and yet dynamic representations remain relatively unexplored. Further, at the same time as also suggesting that natural evolution may provide inspiration for search mechanisms to design machines, he noted that mechanisms inspired by the social aspects of learning may prove useful. This paper presents initial results from consideration of using Turing's dynamical representation within an unconventional substrate - networks of Belousov-Zhabotinsky vesicles - designed by an imitation-based, i.e., cultural, approach.

1 INTRODUCTION

In 1948 Alan Turing produced an internal paper where he presented a formalism he termed "unorganised machines" by which to represent intelligence within computers (eventually published in [37]). These consisted of two main types: A-type unorganised machines, which were composed of two-input NAND gates connected into disorganised networks (Figure 1); and, B-type unorganised machines which included an extra triplet of NAND gates on the arcs between the NAND gates of A-type machines by which to affect their behaviour in a supervised learning-like scheme. In both cases, each NAND gate node updates in parallel on a discrete time step with the output from each node arriving at the input of the node(s) on each connection for the next time step. The structure of unorganised machines is therefore very much like a simple artificial neural network with recurrent connections and hence it is perhaps surprising that Turing made no reference to McCulloch and Pitts' [28] prior seminal paper on networks of binary-thresholded nodes. However, Turing's scheme extended McCulloch and Pitts' work in that he also considered the training of such networks with his B-type architecture. This has led to their also being known as "Turing's connectionism" (e.g., [10]). Moreover, as Teuscher [34] has highlighted, Turing's unorganised machines are (discrete) nonlinear dynamical systems and therefore have the potential to exhibit complex behaviour despite their construction from simple elements. The current work aims to explore the use of Turing's dynamic system representation within networks of small lipid-coated vesicles. The excitable chemical Belousov-Zhabotinsky (BZ) medium is packaged into the vesicles which form the simple/elementary components of a liquid information processing system. The vesicles communicate through chemical "signals" as excitation propagates from vesicle to vesicle. Initial experimental implementations which use micro-fluidics to control vesicle placement have recently been reported [24]. This paper considers implementation of the basic two-input NAND gates using the vesicles and then how to design networks of vesicles to perform a given computation. In particular, a form of collision-based computing (e.g., [1]) is used, along with imitation programming (IP) [8], which was also inspired by Turing's 1948 paper, specifically the comment that "*Further research into intelligence of machinery will probably be very greatly concerned with 'searches' …. [an example] form of search is what I should like to call the 'cultural search' … the search for new techniques must be regarded as carried out by the human community as a whole"* [37].

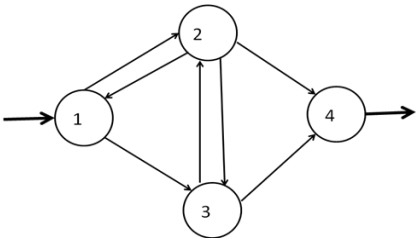


Figure 1. Example A-type unorganised machine consisting of four two-input NAND gate nodes (N=4), with one input (node 1) and one output (node 4) as indicated by the bold arrows.

2 UNORGANISED MACHINES

A-type unorganised machines have a finite number of possible states and they are deterministic, hence such networks eventually fall into a basin of attraction. Turing was aware that his A-type unorganised machines would have periodic behaviour and he stated that since they represent "*about the simplest model of a nervous system with a random arrangement of neurons*" it would be *"of very great interest to find out something about their behaviour"* [37]. Figure 2 shows the fraction of nodes which change state per update cycle for 100 randomly created networks, each started from a random initial configuration, for various numbers of nodes *N*. As can be seen, the time taken to equilibrium is typically around 15 cycles, with all nodes in the larger case changing state on each cycle thereafter, i.e., oscillating (see also [34]). For the smaller networks, some nodes

[1] Unconventional Computing Group, Univ. of the West of England, BS16 1QY, UK. Email: {larry.bull, julian2.holley, ben.delacycostello, andrew.adamatzky}@uwe.ac.uk.

remain unchanging at equilibrium on average; with smaller networks, the probability of nodes being isolated is sufficient that the basin of attraction contains a degree of node stasis. However, there is significant variance in behaviour.

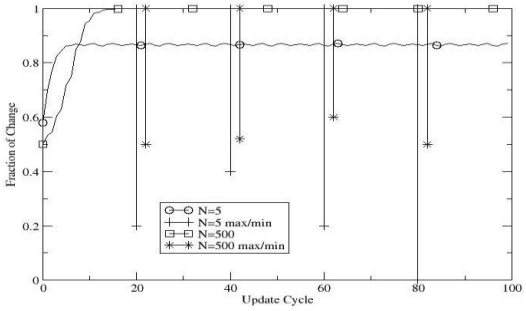


Figure 2. Showing the average fraction of two-input NAND gate nodes which change state per update cycle of random A-type unorganised machines with various numbers of nodes *N*. Error bars show max. and min. values from 100 trials.

Turing [37] envisaged his A-type unorganised machines being used such that they " *... are allowed to continue in their own way for indefinite periods without interference from outside*" and went on to suggest that one way to use them for computation would be to exploit how the application of external inputs would alter the (dynamic) behaviour of the machine. This can be interpreted as his suggesting individual attractors be used to represent distinct (discrete) machine states and the movement between different attractors as a result of different inputs a way to perform computation. Note this hints at some of the ideas later put forward by Ashby [6] on brains as dynamic systems.

Teuscher [34] used a genetic algorithm (GA) [18] to design A-type unorganised machines for bitstream regeneration tasks and simple pattern classification. Bull [8] used IP to design simple logic circuits, such as multiplexers, from them. Here the unorganised machine had an external input applied, was then updated for some number of cycles, e.g., sufficient for an attractor to be typically reached, and then the state of one or more nodes was used to represent the output. More generally, it is well-established that discrete dynamical systems can be robust to faults, can compute, can exhibit memory, etc. (e.g., see [22][39]).

Given their relative architectural simplicity but potential for complex behaviour, A-type unorganised machines appear to be a good candidate (dynamic) representation to use with novel computing substrates. Their use for a chemical computing system is considered here. It can be noted that Turing (e.g., [38]) was also interested in chemical reaction-diffusion systems, for pattern formation not computation.

3 CHEMICAL COMPUTING

Excitable and oscillating chemical systems have been used to solve a number of computational tasks such as implementing logical circuits [32], image processing [25], shortest path problems [31] and memory [29]. In addition chemical diodes [5], coincidence detectors [15] and transformers where a periodic input signal of waves may be modulated by the barrier into a complex output signal depending on the gap width and frequency of the input [30] have all been demonstrated experimentally. See [2] for an overview.

A number of experimental and theoretical constructs utilising networks of chemical reactions to implement computation have been described. These chemical systems act as simple models for networks of coupled oscillators such as neurons, circadian pacemakers and other biological systems [23]. Ross and co-workers [16] produced a theoretical construct suggesting the use of "chemical" reactor systems coupled by mass flow for implementing logic gates neural networks and finite-state machines. In further work Hjelmfelt et al. [17] simulated a pattern recognition device constructed from large networks of mass-coupled chemical reactors containing a bistable iodate-arsenous acid reaction. They encoded arbitrary patterns of low and high iodide concentrations in the network of 36 coupled reactors. When the network is initialized with a pattern similar to the encoded one then errors in the initial pattern are corrected bringing about the regeneration of the stored pattern. However, if the pattern is not similar then the network evolves to a homogenous state signalling non-recognition.

In related experimental work Laplante et al. [26] used a network of eight bistable mass coupled chemical reactors (via 16 tubes) to implement pattern recognition operations. They demonstrated experimentally that stored patterns of high and low iodine concentrations could be recalled (stable output state) if similar patterns were used as input data to the programmed network. This highlights how a programmable parallel processor could be constructed from coupled chemical reactors. This described chemical system has many properties similar to parallel neural networks. In other work Lebender and Schneider [27] described methods of constructing logical gates using a series of flow rate coupled continuous flow stirred tank reactors (CSTR) containing a bistable nonlinear chemical reaction. The minimal bromate reaction involves the oxidation of cerium(III) (Ce^{3+}) ions by bromate in the presence of bromide and sulphuric acid. In the reaction the Ce^{4+} concentration state is considered as "0" "false" ("1""true") if a given steady state is within 10% of the minimal (maximal) value. The reactors were flow rate coupled according to rules given by a feedforward neural network run using a PC. The experiment is started by feeding in two "true" states to the input reactors and then switching the flow rates to generate "true"-"false", "false"-"true" and "false"-"false". In this three coupled reactor system the AND (output "true" if inputs are both high Ce^{4+}, "true"), OR (output "true" if one of the inputs is "true"), NAND (output "true" if one of the inputs is "false") and NOR gates (output "true" if both of the inputs are "false") could be realised. However to construct XOR and XNOR gates two additional reactors (a hidden layer) were required. These composite gates are solved by interlinking AND and OR gates and their negations. In their work coupling was implemented by computer but they suggested that true chemical computing of some Boolean functions may be achieved by using the outflows of reactors as the inflows to other reactors, i.e., serial mass coupling.

As yet no large scale experimental network implementations have been undertaken mainly due to the complexity of analysing and controlling many reactors. That said there have been many experimental studies carried out involving coupled oscillating

and bistable systems (e.g., see [33][11][7][21]). The reactions are coupled together either physically by diffusion or an electrical connection or chemically, by having two oscillators that share a common chemical species. The effects observed include multistability, synchronisation, in-phase and out of phase entrainment, amplitude or "oscillator death", the cessation of oscillation in two coupled oscillating systems, or the converse, "rhythmogenesis", in which coupling two systems at steady state causes them to start oscillating [13].

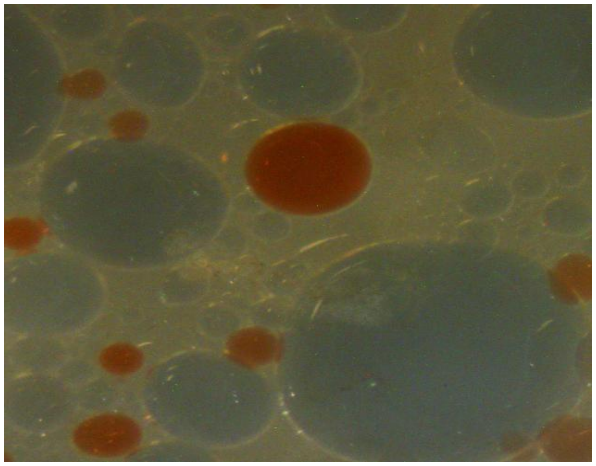


Figure 3. Showing the BZ droplet vesicles.

Vesicles formed from droplets of BZ medium (Figure 3), typically just a few millimetres in diameter, exhibit many properties which may be considered as rudimentary for possible future molecular processing systems: signal transmission, self-repair, signal gain, self-organisation, etc. Their potential use for computation has begun to be explored through collision-based schemes (e.g., [3][4][19][20]). This paper considers their use within a dynamic representation using a collision-based scheme.

Collision-based computing exploits the interaction of moving elements and their mutual effects upon each other's movement wherein the presence or absence of elements at a given point in space and time can be interpreted as computation (e.g., see [2] for chemical systems). Collision-based computing is here envisaged within recurrent networks of BZ vesicles, i.e., based upon the movement and interaction of waves of excitation within and across vesicle membranes. For example, to implement a two-input NAND gate, consider the case shown in Figure 4: when either input is applied, as a stream of waves of excitation, no waves are seen at the output location in the top vesicle - only when two waves coincide is a wave subsequently seen at the output location giving logical AND. A NOT gate can be constructed through the disruption of a constant Truth input in another vesicle, as shown.

A-type unorganised machines can therefore be envisaged within networks of BZ vesicles using the three-vesicle construct for the NAND gate nodes, together with chains of vesicles to form the connections between them. Creation of such chains is reported in the initial experimentation with micro-fluidics noted above [24]. As also noted above, it has recently been shown that IP is an effective design approach with the dynamic representation.

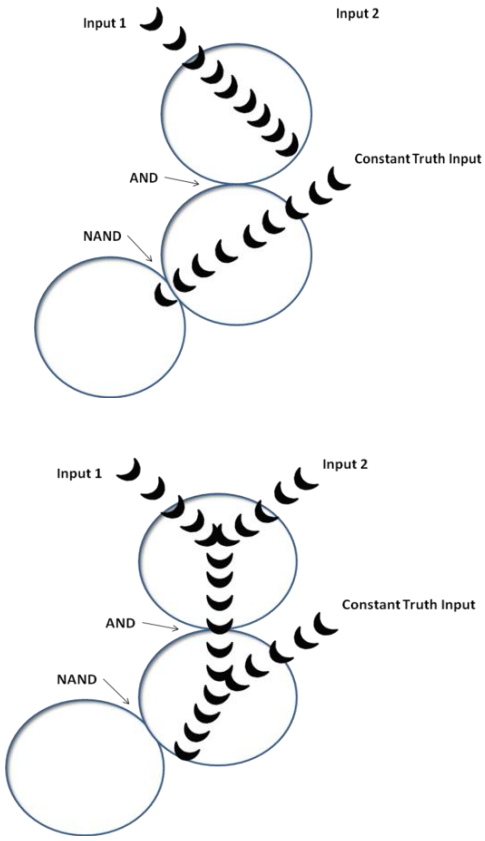


Figure 4. Showing the construction of a two-input NAND gate under a collision-based scheme using three BZ vesicles. The cases of inputs True-False (top) and True-True (bottom) are shown. Techniques such as micro-fluidics are envisaged as being used to influence/control vesicle position.

4 IMITATION PROGRAMMING

For A-type design, IP utilizes a variable-length representation of pairs of integers defining node inputs, each with an accompanying single bit defining the node's start state. There are three imitation operators - copy a node connection, copy a node start state, and change size through copying. In this paper, each operator can occur with or without error, with equal probability, such that an individual performs one of the six during the imitation process as follows.

To copy a node connection, a randomly chosen node has one of its randomly chosen connections set to the same value as the corresponding node and its same connection in the individual it is imitating. When an error occurs, the connection is set to the next or previous node (equal probability, bounded by solution size). Imitation can also copy the start state for a randomly chosen node from the corresponding node, or do it with error (bit flip here). Size is altered by adding or deleting nodes and depends upon whether the two individuals are the same size. If the individual being imitated is larger than the copier, the connections and node start state of the first extra node are copied

to the imitator, a randomly chosen node being connected to it. If the individual being imitated is smaller than the copied, the last added node is cut from the imitator and all connections to it re-assigned. If the two individuals are the same size, either event can occur (with equal probability). Node addition adds a randomly chosen node from the individual being imitated onto the end of the copier and it is randomly connected into the network. The operation can also occur with errors such that copied connections are either incremented or decremented. For a problem with a given number of binary inputs I and a given number of binary outputs O, the node deletion operator has no effect if the parent consists of only $O + I + 2$ nodes. The extra two inputs are constant True and False lines. Similarly, there is a maximum size (100) defined beyond which the growth operator has no effect.

In this paper, each individual in the population P creates one variant of itself and it is adopted if better per iteration. In the case of ties, the solution with the fewest number of nodes is kept to reduce size, otherwise the decision is random. The individual to imitate is chosen using a roulette-wheel scheme based on proportional solution utility, i.e., the traditional reproduction selection scheme used in GAs. Other forms of updating, imitation processes, and imitation selection are, of course, possible [8]. In this form IP may be seen as combining ideas from memetics [12] with Evolutionary Programming [14]. It can be noted GAs have previously been used to design chemical computing systems in various ways (e.g., [9][35][36]).

5 EXPERIMENTATION

In the following, three well-known logic problems are used to begin to explore the characteristics and capabilities of the general approach. The multiplexer task is used since they can be used to build many other logic circuits, including larger multiplexers. These Boolean functions are defined for binary strings of length $l = k + 2^k$ under which the k bits index into the remaining 2^k bits, returning the value of the indexed bit. Hence the multiplexer has multiple inputs and a single output. The demultiplexer and adders have multiple inputs and multiple outputs. As such, simple examples of each are also used here. In all cases, the correct response to a given input results in a quality increment of 1, with all possible binary inputs being presented per solution evaluation. Upon each presentation of an input, each node in an unorganised machine has its state set to its specified start state. The input is applied to the first connection of each corresponding I input node. The A-type is then executed for 15 cycles. The value on the output node(s) is then taken as the response. All results presented are the average of 20 runs, with P=20. Experience found giving initial random solutions N=O+I+2+30 nodes was useful across all the problems explored here, i.e., with the other parameter/algorithmic settings.

Figure 5 shows the performance of IP to design A-type unorganised machines on k=2 versions of the three tasks: the 6-bit multiplexer (opt. 64), 2-bit adder (opt. 16) and 6-bit demultiplexer (opt. 8). As can be seen, optimal performance is reached in all cases, well within the allowed time, and that the solution sizes are adjusted to the given task. That is, discrete dynamical circuits capable of the given logic functions have been designed. As discussed elsewhere [8], the relative robustness of such circuits to faults, their energy usage, etc. remains to be explored.

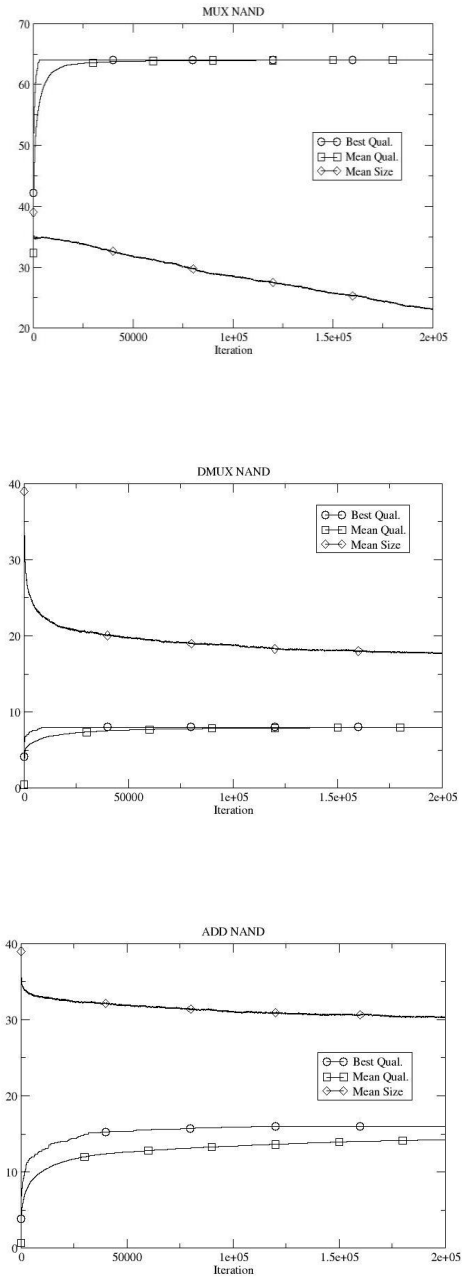


Figure 5. Showing the performance of IP in designing A-type unorganised machines for the three logic tasks.

However, to begin to consider implementing such designs within BZ vesicles, the time taken for signal propagation between NAND gate nodes needs to included. That is, in Figure 5, as in all previous work with such dynamic representations, any changes in node state are immediately conveyed to any other

connected nodes since a traditional computational substrate is assumed. Within the vesicles, changes in NAND gate node state will propagate through chains and hence there will be a time delay proportional to the distance between nodes.

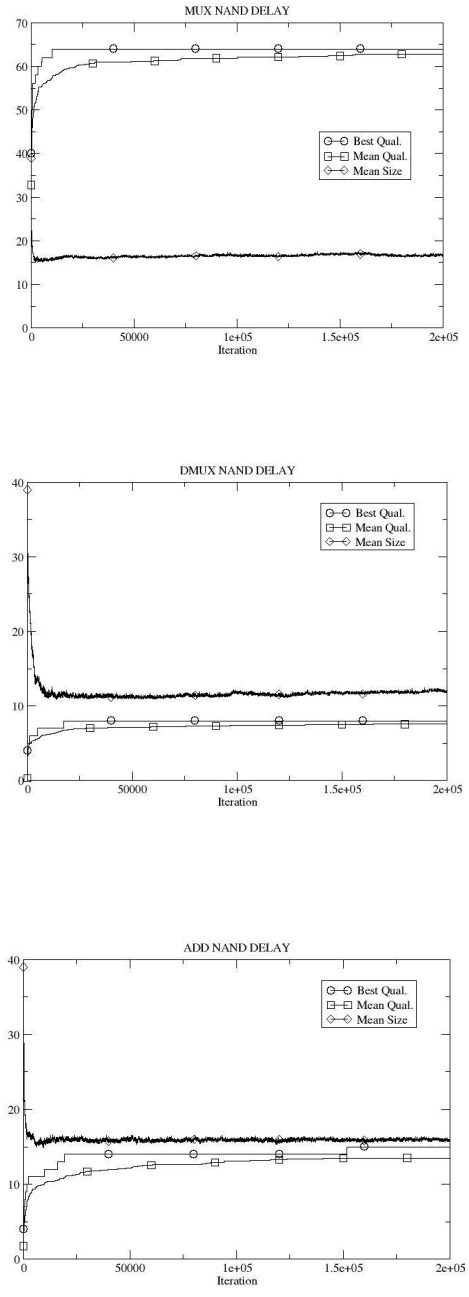


Figure 6. Showing the performance of IP in designing A-type unorganised machines for the three logic tasks with signal propagation times added.

Figure 6 shows results for the same experiments and parameters as before but with a form of time delay added to begin to consider the physical implementation in an elementary way. Here NAND gate node states take the same number of update cycles to propagate between nodes as the absolute difference in node number. For example, the state of node 11 at time *t* would take 8 update cycles to reach node 3. Hence at update cycle *t*+8, node 3 would use the state of node 11 as at time *t* as one of its inputs. The number of overall update cycles for the A-types was increased to 50.

As Figure 6 shows, it takes longer to reach optimal solutions (T-test, $p<0.05$) and they are perhaps surprisingly smaller (T-test, $p<0.05$) than before, but suitable dynamic designs are again found in the allotted time, except for the adder which takes longer to reach optimality (not shown).

6 CONCLUSIONS

Over sixty years ago, Alan Turing presented a simple representation scheme for machine intelligence – a discrete dynamical system network of two-input NAND gates. Since then only a few other explorations of these unorganized machines are known. As noted above, it has long been argued that dynamic representations provide numerous useful features, such as an inherent robustness to faults and memory capabilities by exploiting the structure of their basins of attraction. For example, unique attractors can be assigned to individual system states/outputs and the map of internal states to those attractors can be constructed such that multiple paths of similar states lead to the same attractor. In this way, some variance in the actual path taken through states can be varied, e.g., due to errors, with the system still responding appropriately. Turing appears to have been thinking along these lines also.

Given the relative simplicity of A-types but their potential for complex behaviour, this paper suggests they may provide a useful representation scheme for unconventional computing substrates. Unconventional computing aims to go beyond traditional architectures and formalisms, much of which is based upon Turing's work on computability, by exploiting the inherent properties of systems to perform computation. A number of experimental systems have been presented in biological, chemical and physical media. Where NAND gate function can be realised, whilst also leaving open the potential utilisation of other aspects of the chosen medium, A-types could be explored. In particular, a substrate of BZ vesicles recently presented as a step towards molecular information processing, e.g., for future smart drugs, was considered and a form of two-input NAND gate designed for it through collision-based computing.

It was then shown how a number of well-known benchmark logic circuits can be designed from A-type unorganised machines using an approach inspired by a comment from Turing on cultural search. Further consideration of the physical implementation within networks of BZ vesicles meant that signal propagation times were also included into the A-types. Results indicate that the design process was slowed relatively but still effective.

Current work is increasing the level of detail of the simulated chemical system both in terms of the vesicle structure and of the BZ therein. Future use within the real substrate is expected to open the potential to further exploit emergent properties such as structural self-organisation and non-linear behaviour more fully.

Acknowledgement

The research was supported by the NEUNEU project sponsored by the European Community within FP7-ICT-2009-4 ICT-4-8.3 - FET Proactive 3: Bio-chemistry-based Information Technology (CHEM-IT) program.

REFERENCES

[1] Adamatzky, A. (Ed.) *Collision-based Computing*. Springer, London (2002).
[2] Adamatzky, A., De Lacy Costello, B. & Asai, T. *Reaction-Diffusion Computers*. Elsevier. (2005)
[3] Adamatzky, A., Holley, J., Bull, L. & De Lacy Costello, B. On Computing in Fine-grained Compartmentalised Belousov–Zhabotinsky Medium. *Chaos, Solitons & Fractals*, 44(10):779-790 (2011).
[4] Adamatzky, A., De Lacy Costello, B., Holley, J., Gorecki, J. & Bull, L. Vesicle computers: Approximating a Voronoi diagram using Voronoi automata. *Chaos Solitons and Fractals 44*:480-489 (2011)
[5] Agladze K, Aliev RR, Yamaguhi T & Yoshikawa K. Chemical diode. *Journal of Physical Chemistry*, 100:13895-13897 (1996)
[6] Ashby, W.R. *Design for a Brain*. Wiley, New York (1954).
[7] Bar-Eli, K. & Reuveni, S. (1985). Stable stationary-states of coupled chemical oscillators: Experimental evidence. *Journal of Physical Chemistry*, 89, 1329-1330
[8] Bull, L. Using Genetical and Cultural Search to Design Unorganised Machines. *Evolutionary Intelligence*, 5(1): (2012).
[9] Bull, L., Budd, A., Stone, C., Uroukov, I., De Lacy Costello, B. & Adamatzky, A. (2008) Towards Unconventional Computing Through Simulated Evolution: Learning Classifier System Control of Non-Linear Media. *Artificial Life* 14(2): 203-222
[10] Copeland, J. & Proudfoot, D. On Alan Turing's Anticipation of Connectionism. *Synthese* 108:361-377 (1996)
[11] Crowley, M.F. & Field, R.J. Electrically coupled Belousov-Zhabotinskii oscillators 1: Experiments and simulations. *Journal of Physical Chemistry*, 90:1907-1915 (1986)
[12] Dawkins, R. *The Selfish Gene*. Oxford Press, Oxford (1976)
[13] Dolnik, M. & Epstein, I.R. Coupled chaotic oscillators. *Physical Review E*, 54:3361-3368 (1996)
[14] Fogel, L. J., Owens, A.J. & Walsh, M.J. Artificial Intelligence Through A Simulation of Evolution. In M. Maxfield et al. (Eds) *Biophysics and Cybernetic Systems: Proceedings of the 2nd Cybernetic Sciences Symposium*. Spartan Books, pp131-155 (1965)
[15] Gorecki, J., Yoshikawa, K. & Igarashi, Y. On chemical reactors that can count. *Journal of Physical Chemistry A*, 107:1664-1669 (2003)
[16] Hjelmfelt, A. & Ross, J. Mass-coupled chemical systems with computational properties. *Journal of Physical Chemistry*, 97:7988-7992 (1993)
[17] Hjelmfelt, A., Weinberger, E.D. & Ross, J. Chemical implementation of neural networks and Turing machines. *PNAS* 88:10983-10987 (1991)
[18] Holland, J.H. *Adaptation in Natural and Artificial Systems*. Univ. of Mich. Press. (1975)
[19] Holley, J., Adamatzky, A., Bull, L., De Lacy Costello, B. & Jahan, I. Computational modalities of Belousov-Zhabotinsky encapsulated vesicles, *Nano Communication Networks*, *2*: 50-61 (2011)
[20] Holley, J., Jahan, I., De Lacy Costello, B., Bull, L. & Adamatzky, A. Logical and Arithmetic Circuits in Belousov Zhabotinsky Encapsulated Discs, *Physical Review E 84*: 056110 (2011)
[21] Holz, R. & Schneider, F.W. (1993). Control of dynamic states with time-delay between 2 mutually flow-rate coupled reactors. *Journal of Physical Chemistry*, 97, 12239
[22] Kauffman, S. A. *The Origins of Order*. Oxford Press, Oxford (1993)
[23] Kawato, M. & Suzuki, R. Two coupled neural oscillators as a model of the circadian pacemaker. *Journal of Theoretical Biology*, 86:547-575 (1980)
[24] King, P. H., Corsi, J. C., Pan, B.-H., Morgan, H., de Planque, M. R. & Zauner, K.-P. Towards molecular computing: Co-development of microfluidic devices and chemical reaction media. *Biosystems* (2012)
[25] Kuhnert, L., Agladze, K.I. & Krinsky, V.I. Image processing using light sensitive chemical waves. *Nature*, 337:244-247 (1989)
[26] Laplante, J.P., Pemberton, M., Hjelmfelt, A. & Ross, J. Experiments on pattern recognition by chemical kinetics. *Journal of Physical Chemistry*, 99:10063-10065 (1995)
[27] Lebender, D. & Schneider, F.W. Logical gates using a nonlinear chemical reaction. *Journal of Physical Chemistry*, 98:7533-7537 (1994)
[28] McCulloch, W.S. & Pitts, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics* 5: 115-133 (1943)
[29] Motoike, I.N., Yoshikawa, K., Iguchi, Y. & Nakata, S. Real time memory on an excitable field. *Physical Review E*, 63:1-4 (2001)
[30] Sielewiesiuk, J. & Gorecki, J. Passive barrier as a transformer of chemical frequency. *Journal of Physical Chemistry A*, 106:4068-4076 (2002)
[31] Steinbock, O., Toth, A. & Showalter, K. Navigating complex labyrinths: Optimal paths from chemical waves. *Science*, 267:868-871 (1995)
[32] Steinbock, O., Kettunen, P. & Showalter, K. Chemical wave logic gates. *Journal of Physical Chemistry*, 100:18970-18975 (1996)
[33] Stuchl, I. & Marek, M. Dissipative structures in coupled cells: Experiments. *Journal of Physical Chemistry*, 77:2956-63 (1982)
[34] Teuscher, C. *Turing's Connectionism*. Springer, London (2002)
[35] Toth, R., Stone, C., De Lacy Costello, B., Adamatzky, A. & Bull, L. (2008) Dynamic Control and Information Processing in the Belousov-Zhabotinsky Reaction using a Co-evolutionary Algorithm. *Journal of Chemical Physics* 129: 184708
[36] Toth, R., Stone, C., De Lacy Costello, B., Adamatzky, A. & Bull, L. (2009) Simple Collision-based Chemical Logic Gates with Adaptive Computing. *Journal of Nanotechnology and Molecular Computation* 1(3): 1-16
[37] Turing, A. Intelligent Machinery. In C.R. Evans & A. Robertson (Eds) *Key Papers: Cybernetics*. Butterworths, pp91-102 (1968)
[38] Turing, A. The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237 (641):37-72 (1952)
[39] Wolfram, S. *A New Kind of Science*. Wolfram Media. (2002)

Some constraints on the physical realizability of a mathematical construction

Francisco Hernández-Quiroz[1] and **Pablo Padilla**[2]

Abstract. Mathematical constructions of abstract entities are normally done disregarding their actual physical realizability. The definition and limits of the physical realizability of these constructions are controversial issues at the moment and the subject of intense debate.

In this paper, we consider a simple and particular case, namely, the physical realizability of the enumeration of rational numbers by Cantor's diagonalization by means of an Ising system.

We contend that uncertainty in determining a particular state in an Ising system renders impossible to have a reliable implementation of Cantor's diagonal method and therefore a stronger physical system is required. We also point out what are the particular limitations of this system from the perspective physical realizability.

1 Introduction

"There is no quantum world. There is only an abstract quantum description. It is wrong to think physics' task is to discover how Nature is. Physics deals with what is possible to say about Nature."

This quote is attributed to Niels Bohr, when he was asked whether the quantum formalism reflected the underlying physical reality. Bohr's, other philosophers' and scientists' opinions aside, a good deal of paper has been used to analyse the possibility of describing and understanding reality by means of formal mathematical tools. Barrow, Chaitin, Hawking and Penrose (among others) have advanced some ideas with varying degrees of formality.

Here we address a reciprocal question: given a mathematical construction and a particular physical system, is the latter adequate to "implement" the former? By implementation we mean an actual physical device that (a) has structural properties that correspond to components of the mathematical entity (some have talked about an isomorphism between physical and mathematical structures [3], but a weaker notion may also do); (b) a physical procedure that can produce experimental results which reflect accurately corresponding properties of the mathematical construction.

These are very intricate and hard questions to be answered definitely in a general case. Our aim is more modest, namely to explore a specific instance of this problem: we take the classical Cantor's diagonalization for the enumeration of the rational numbers [2] and how it can be implemented by an Ising system. We provide a specific implementation and show its limitations deriving from properties of the physical system itself.

This leads us to think that some clearly defined mathematical questions cannot always be posed and answered within the context of a particular physical system. Of course, the more general question of the existence of a physical system realizing a particular mathematical construction is beyond the limits of this work but we hope our example helps to stimulate discussions on this line of thought. The standard interpretation of quantum mechanics regarding physically meaningful questions is that it should be possible to pose them in such a way that they can be answered experimentally.

The reciprocal question is also interesting: to what extent mathematical constructions should be considered valid? One possible approach, would imply that only those mathematical constructions that can actually be implemented by means of a physical system can in fact be used, at least in terms of computation.

In the next section we present—as a reminder—Cantor's diagonalization method for enumerating the rational numbers. The third section deals with Ising systems and its properties.The fourth section presents our implementation of Cantor's method and how to find a specific rational number. In the final section, which is the central part of this paper, we show how our system is unable to perform the task for which it was designed due to intrinsic limitations of Ising systems and other physical principles, and we also discuss some implications.

2 Cantor's diagonalization

In 1878 Cantor defined rigorously when two sets have the same cardinality. Let A and B be two sets. They have the same number of elements if and only if there exists a bijection between them, i.e., a function $f : A \to B$ which is both injective and surjective.

He also proved that the set of natural numbers and the set of rational numbers are equinumerous, even though the former is a proper subset of the latter. His argument introduced an ingenious device to construct a one-to-one correspondence between the two sets. The idea is that rational numbers are not arranged according to the traditional $<$ relation, but rather, by taking advantage of the fact that a rational number (in accordance with the etimology of the name) can be regarded as the ratio of two integers. For example, the number 0.5 is also represented by the fraction $1/2$.

The fractional representation of a number, let us say m/n, can be transformed into the convention that the pair *(m,n)* represents this very number. Now consider the list

$$(1,1),(1,2),(2,1),(1,3),(2,2),(3,1),(1,4),\ldots$$

where pairs are arranged so that the sum of the two components is increasing; pairs whose sum produces the same value are ordered by the traditional $<$ order applied to the first coordinate of the pairs. By omitting pairs representing the same number (which can always be calculated in a finite number of steps as the list is being produced), this is a bijection between natural and rational numbers, and thus both sets have the same cardinality.

[1] School of Science, UNAM, email: fhq@ciencias.unam.mx
[2] Institute for Applied Mathematics, IIMAS, UNAM, email: pablo@mym.iimas.unam.mx

If we set aside the traditional objections posed by mathematical constructivists to the idea of actual infinite sets, Cantor's argument seems very straightforward and has been regarded as such ever since. However we could take a mathematical constructive perspective and reject Cantor's device (and his whole set theory, for that matter).

But we can also take a different constructive perspective, which we may name *physical constructivism*: What requirements should a particular physical system meet in order to serve as a basis for implementing Cantor's device? At first sight there must be physical systems on which this may not be possible (although the symmetrical question does not seem easy to answer). Specifically, we will analyse the feasibility of Ising models for this task in the next section.

3 Ising models

In the last decades, some models in physics have played a central role in understanding specific connections between mathematical aspects of the theory and experiments. One of such is precisely the Ising model. We use it here for different purposes. We suggest that it can be taken as a real system in which Cantor's diagonal procedure could be implemented and therefore as a starting point from which conclusions can be drawn regarding the limitations that mathematical constructions could have in the physical world. This is due to the fact that, in principle, the physical configurations of the system can be put in correspondence with rational numbers. Moreover, for the Ising model a direct relationship between the physical entropy and the informational entropy can be established, allowing a quantitative comparison .

We briefly recall what the Ising model is about and later on we make a few remarks on the entropy of a discrete physical system. What follows is basically adapted from [5].

We consider a magnetic material in which the electrons determining the magnetic behaviour are localized near the atoms of a lattice and can have only two magnetization states (spin up or down). The spin for a given site in this lattice will be identified with the of the 0's or 1's used in the mathematical construction of the previous section to write down the binary expansion of the rational numbers. Notice that we need only a finite number or 0's or 1's since these expansions will be either finite or periodic. For instance, we might put in a row all numbers (m, n) of a fixed height one after the other with a conventional sequence to denote beginning and end of a number. As mentioned before, the magnetization S_i can take only two values ± 1 that we identify with 0 and 1 respectively. There is a Hamiltonian associated in the presence of an external magnetic force depending on the site, h_i which is given by:

$$H = -J \sum_{i,k} S_i S_j - \sum_i h_i S_i,$$

where the sum over i and k runs over all possible nearest-neighbour pairs of the lattice and J is the so called exchange constant.

The fact that is important to stress is that a possible enumeration of the rationals correspond to a particular physical configuration. Notice that we are desregarding the obvious limitation of size. That is, in Cantor's procedure we need an infinite number of rows and columns, that is an ideal lattice, whereas a physical material will necessarily have finite size. Nevertheless, we will see that even then, there are physical constraints that are imposed by the quantum nature of the system to the entropy, which can be interpreted as informational restrictions on the physical realizability of the mathematical construction.

For a continuous system whose configuration is denoted by C, where the configuration space is assumed to be endowed with a measure μ (for simplicity one may think of $\mathbb{R}^d$, the entropy associated with a specific probability distribution P is given by

$$S[P] = -\int d\mu(C) P(C) ln P(C),$$

that is, the expected value of $-lnP(C)$ with respect to μ.

By dividing the space into cells of size ε^d the entropy of the continuous system can be well approximated by the entropy of the discrete system resulting from the partition:

$$S_{disc} = S_{cont} - d ln(\varepsilon).$$

As a matter of fact, the ε can be taken to be the Planck constant for a quantum system. This observation will be important later on.

4 Implementing Cantor's method

As we mentioned before, we can in principle use the Ising system to physically array and enumerate the rational numbers and locate any of them in this array. In fact the question: "How to find a rational number in the list?" is well defined and would only need a finite number of steps.

In the section devoted to the Ising model, we recalled equation 3 for the entropy of a quantum system. Notice that the second term is positive and independent of the details of the system, only due to the quantum nature of the same. This has an important implication in terms of the possibility of actually determining the state in which the Ising model is. If we relate the information content with the entropy of the system we see that, in order for the state of the system to be completely determined, we would need zero entropy [4]. This is physically impossible. Moreover, a lower bound for the entropy is related not only to the discrete (quantum) nature of the system, but it also depends on the temperature and other parameters. The conclusion is that even when the counting and locating procedure is well defined, there is always an intrinsic error. Of course one might argue that this is probably due to the chosen system, but the reasoning is general enough as to suggest that no matter what physical implementation we choose, there will always exist this limitation.

5 Conclusion: Uncertainty comes in the way or how real is reality?

We have argued that uncertainty in determining a particular state in an Ising system renders impossible to have a reliable implementation of Cantor's diagonal method. There are also other related mathematical constructions that could be analysed in a similar way. For instance, Cantor's proof of the uncountability of the real numbers relies on similar ideas. As a matter of fact, in the usual argument, a contradiction is obtained by producing a real number that cannot be included in a proposed enumeration. This is done by considering the diagonal sequence and taking its negation. Once this is done, it can be shown that if t is the truth value of the element of this sequence intersecting the diagonal, then it would have to satisfy the relation

$$t = 1 - t,$$

which leads to a contradiction *if one assumes the only possible truth values are 0 or 1* (see for instance chapter 2 on diagonalization in[1]). However, this equation does not pose any problem if t is interpreted in a probabilistic way and assigned a value of $1/2$. This opens up a

series of even subtler questions such as whether we can actually have a physical model of the real numbers and many others, that from our perspective, are worth addressing.

Many other people have previously addressed this questions either in general terms or for particular mathematical concepts. A pioneering work is [6], which posed the question of realizing an abstract mapping process within the constraints of a physical version of Church's thesis. A very recent case study in the field of control and quantum systems can be found in [7].

REFERENCES

[1] Boolos, G.S., Burgess, J.P. and Jeffrey, R.C. *Computability and Logic*, Cambridge Univ. Press, New York (2007).

[2] Cantor, G., *Contributions to the Founding of the Theory of Transfinite Number*, Dover, New York (1915).

[3] Chalmers, D., "On Implementing a Computation", *Mind and Machines*, 4, 391–402 (1994).

[4] Kintchin, A., *Mathematical Foundations of Information Theory*, Dover, London (1963).

[5] Parisi, G., *Statistical Field Theory*. Frontiers in Physics. Addison-Wesley Publishing Company, (1988).

[6] R Rosen. Church?s thesis and its relation to the concept of realizability in biology and physics. Bulletin of Mathematical Biology 24, 375–393 (1962).

[7] I. Petersen and A. Shaiju. A Frequency Domain Condition for the Physical Realizability of Linear Quantum Systems. *IEEE Transactions on Automatic Control*, issue 9, (2012).

Axiomatic Tools versus Constructive approach to Unconventional Algorithms

Gordana Dodig-Crnkovic[1] and **Mark Burgin**[2]

Abstract. In this paper, we analyze axiomatic issues of unconventional computations from a methodological and philosophical point of view. We explain how the new models of algorithms changed the algorithmic universe, making it open and allowing increased flexibility and creativity. However, the greater power of new types of algorithms also brought the greater complexity of the algorithmic universe, demanding new tools for its study. That is why we analyze new powerful tools brought forth by the axiomatic theory of algorithms, automata and computation.

1 INTRODUCTION

Tradition in computation is represented by conventional computations. The conventional types and models of algorithms make the algorithmic universe, i.e., the world of all existing and possible algorithms, closed because there is a rigid boundary in this universe formed by recursive algorithms such as Turing machines.

Super-recursive algorithms controlling and directing unconventional computations break this boundary bringing people to an *open algorithmic universe* – a world of unbounded creativity. As the growth of possibilities involves much higher complexity of the new open world of super-recursive algorithms, innovative hardware and unconventional organization, we discuss means of navigation in this new open algorithmic world.

The paper is organized as follows. First in Section 2 we compare local and global mathematics. Section 3 addresses local logics and logical varieties, while Section 4 offers the discussion of projective mathematics versus reverse mathematics versus classical mathematics. Section 5 answers the question how to navigate in the algorithmic multiverse. Finally Section 6 presents our conclusions and provides directions for future work.

2 LOCAL MATHEMATICS VERSUS GLOBAL MATHEMATICS

Mathematics exists as an aggregate of various mathematical fields. If at the beginning, there were only two fields – arithmetic and geometry, now there are hundreds of mathematical fields and subfields. However, mathematicians always believed in mathematics as a unified system striving to build common and in some sense absolute foundations for all mathematical fields and subfields. At the end of the 19th century, mathematicians came very close to achieving this goal as the emerging set theory allowed building all mathematical structures using only sets and operations with sets. However, in the 20th century, it was discovered that there are different set theories. This brought some confusion and attempts to find the "true" set theory.

To overcome this confusion, Bell [1] introduced the concept of local mathematics in 1986. The fundamental idea was to abandon the unique absolute universe of sets central to the orthodox set-theoretic account of the foundations of mathematics, replacing it by a plurality of local mathematical frameworks. Bell suggested taking elementary toposes as such frameworks, which would serve as local replacements for the classical universe of sets. Having sufficient means for developing logic and mathematics, elementary toposes possess a sufficiently rich internal structure to enable a variety of mathematical concepts and assertions to be interpreted and manipulated. Mathematics interpreted in any such framework is called *local mathematics* and admissible transformation between frameworks amounts to a (definable) *change of local mathematics*. With the abandonment of the absolute universe of sets, mathematical concepts in general lose absolute meaning, while mathematical assertions liberate themselves from absolute truth values. Instead they possess such meanings or truth values only *locally*, i.e., *relative* to local frameworks. It means that the *reference* of any mathematical concept is accordingly not fixed, but *changes* with the choice of local mathematics.

It is possible to extend the approach of Bell in two directions. First, we can use an arbitrary category as a framework for developing mathematics. When an internal structure of such a framework is meager, the corresponding mathematics will be also indigent. Second, it is possible to take a theory of some structures instead of the classical universe of sets and develop mathematics in this framework.

A similar situation emerged in computer science.

Usually to study properties of computers and to develop more efficient applications, mathematicians and computer scientists use mathematical models. There is a variety of such models: Turing machines of different kinds (with one tape and one head, with several tapes, with several heads, with *n*-dimensional tapes, nondeterministic, probabilistic, and alternating Turing machines, Turing machines that take advice and Turing machines with oracle, etc.), Post productions, partial recursive functions, neural networks, finite automata of different kinds (automata without memory, autonomous automata, accepting automata, probabilistic automata, etc.), Minsky machines, normal Markov algorithms, Kolmogorov algorithms, formal grammars of different kinds (regular, context free, context sensitive, phrase-structure, etc.), Storage Modification Machines or simply, Shönhage machines, Random Access Machines (RAM), Petri nets, which like Turing machines have several forms – ordinary, regular, free, colored, self-modifying, etc.), and so on. All these models are constructive, i.e., they have a tractable explicit descriptions and simple rules for operation. Thus, the constructive approach is dominating in computer science.

[1] School of Innovation, Design and Engineering, Mälardalen University, Sweden. Email: gordana.dodig-crnkovic@mdh.se

[2] Dept. of Mathematics, UCLA, Los Angeles, USA. Email: mburgin@math.ucla.edu

This diversity of models is natural and useful because each of these classes is suited for some kind of problems. In other words, the diversity of problems that are solved by computers involves a corresponding diversity of models. For example, general problems of computability involve such models as Turing machines and partial recursive functions. Finite automata are used for text search, lexical analysis, and construction of semantics for programming languages. In addition, different computing devices demand corresponding mathematical models. For example, universal Turing machines and inductive Turing machines allows one to investigate characteristics of conventional computers [7]. Petri nets are useful for modeling and analysis of computer networks, distributed computation, and communication processes [31]. Finite automata model computer arithmetic. Neural networks reflect properties of the brain. Abstract vector and array machines model vector and array computers [7].

To utilize some kind of models that are related to a specific type of problems, we need to know their properties. In many cases, different classes have the same or similar properties. As a rule, such properties are proved for each class separately. Thus, alike proofs are repeated many times in similar situations involving various models and classes of algorithms.

In contrast to this, the *projective* (also called *multiglobal*) *axiomatic theory* of algorithms, automata and computation suggests a different approach [9][30]. Assuming some simple basic conditions (in the form of postulates, axioms and conditions), we derive in this theory many profound properties of algorithms. This allows one, when dealing with a specific model not to prove this property, but only to check the conditions from the assumption, which is much easier than to prove the property under consideration. In such a way, we can derive various characteristics of types of computers and software systems from the initial postulates, axioms and conditions.

Breaking the barrier of the Church-Turing Thesis drastically increased the variety of algorithmic model classes and changed the algorithmic universe of recursive algorithms to the multiverse of super-recursive algorithms, which consists of a plurality of local algorithmic universes. Each class of algorithmic models forms a local algorithmic universe, providing means for the development of local computer science in general and a local theory of algorithms in particular.

Local mathematics brings forth local logics because each local mathematical framework has its own logic and it is possible that different frameworks have different local logics.

3 LOCAL LOGICS AND LOGICAL VARIETIES

Barwise and Seligman (1997) developed a theory of information flow. In it, the concept of local logic plays a fundamental role in the modeling commonsense reasoning. The basic concept of this theory is a classification, which can be interpreted as a representation of some domain in the physical or abstract world. Each local logic corresponds to a definite classification. This implies a natural condition that each domain has its own local logic and different domains may have different local logics.

In the multiverse of super-recursive algorithms, each class of super-recursive algorithms forms a local algorithmic universe, which has a corresponding local logic. These logics may be essentially different. For instant, taking two local algorithmic universes formed by such classes as the class $\boldsymbol{T}$ of all Turing machines and the class $\boldsymbol{TT}$ of all total, i.e., everywhere defined, Turing machines, we can find that the first class satisfies the axiom of universality, which affirms existence of a universal algorithm, i.e., a universal Turing machine in this class. However, the class $\boldsymbol{TT}$ does not satisfy this axiom [9].

Analyzing the system of local logics, it is possible to see that there are different relations between them and it would be useful to combine these logics in a common structure. As it is explained in [9], local logics form a deductive logical variety or a deductive logical prevariety, which were introduced and studied in [4] as a tool to work with inconsistent systems of knowledge.

Minsky [24] was one of the first researchers in AI who attracted attention to the problem of inconsistent knowledge. He wrote that consistency is a delicate concept that assumes the absence of contradictions in systems of axioms. Minsky also suggested that in artificial intelligence (AI) systems this assumption was superfluous because there were no completely consistent AI systems. In his opinion, it is important to understand how people solve paradoxes, find a way out of a critical situation, learn from their own or others' mistakes or how they recognize and exclude different inconsistencies. In addition, Minsky [25] suggested that consistency and effectiveness may well be incompatible. He also writes [26]: "An entire generation of logical philosophers has thus wrongly tried to force their theories of mind to fit the rigid frames of formal logic. In doing that, they cut themselves off from the powerful new discoveries of computer science. Yes, it is true that we can describe the operation of a computer's hardware in terms of simple logical expressions. But no, we cannot use the same expressions to describe the meanings of that computer's output -- because that would require us to formalize those descriptions inside the same logical system. And this, I claim, is something we cannot do without violating that assumption of consistency." Then Minsky [26] continues, "In summary, there is no basis for assuming that humans are consistent - not is there any basic obstacle to making machines use inconsistent forms of reasoning". Moreover, it has been discovered that not only human knowledge but also representations/models of human knowledge (e.g., large knowledge bases) are inherently inconsistent [11]. Logical varieties or prevarieties provide powerful tools for working with inconsistent knowledge.

There are different types and kinds of logical varieties and prevarieties: *deductive* or *syntactic varieties* and *prevarieties*, *functional* or *semantic varieties* and *prevarieties* and *model* or *pragmatic varieties* and *prevarieties*. Syntactic varieties, prevarieties, and quasi-varieties (introduced in [10]) are built from logical calculi as buildings are built from blocks.

Let us consider a logical language L, an inference language R, a class $\mathbf{K}$ of syntactic logical calculi, a set Q of inference rules ($Q \subseteq R$), and a class $\mathbf{F}$ of partial mappings from L to L.

A triad $\mathbf{M} = (A, H, M)$, where A and M are sets of expressions that belong to L (A consists of axioms and M consists of theorems) and H is a set of inference rules, which belong to the set R, is called:

(1) a *projective syntactic* (**K**,**F**)-*prevariety* if there exists a set of logical calculi $\boldsymbol{C}_i = (A_i , H_i , T_i)$ from $\mathbf{K}$ and a system of mappings $f_i : A_i \to L$ and $g_i : M_i \to L$ ($i \in I$) from $\mathbf{F}$ in which A_i consists of all axioms and M_i consists of all theorems of the logical calculus $\boldsymbol{C}_i$, and for which the equalities $A = \cup_{i \in I} f_i(A_i)$, H

$= \bigcup_{i \in I} H_i$ and $M = \bigcup_{i \in I} g_i(M_i)$ are valid (it is possible that $\boldsymbol{C}_i = \boldsymbol{C}_j$ for some $i \neq j$).

(2) a *projective syntactic* (**K,F**)-*variety* with the depth k if it is a projective syntactic (**K,F**)-quasi-prevariety and for any i_1 , i_2 , i_3 , … , $i_k \in I$ either the intersections $\bigcap_{j=1}^{k} f_{ij}(A_{ij})$ and $\bigcap_{j=1}^{k} g_{ij}(T_{ij})$ are empty or there exists a calculus $\boldsymbol{C} = (A, H, T)$ from **K** and projections f: $A \to \bigcap_{j=1}^{k} f_{ij}(A_{ij})$ and g: $N \to \bigcap_{j=1}^{k} g_{ij}(M_{ij})$ from **F** where $N \subseteq T$;

(3) a *syntactic* **K**-prevariety if it is a projective syntactic (**K,F**)-prevariety in which $M_i = T_i$ for all $i \in I$ and all mappings f_i and g_i that define **M** are bijections on the sets A_i and M_i , correspondingly;

(4) a *syntactic* **K**-*variety* if it is a projective syntactic (**K,F**)-variety in which $M_i = T_i$ for all $i \in I$ and all mappings f_i and g_i that define **M** are bijections on the sets A_i and M_i , correspondingly.

The calculi $\boldsymbol{C}_i$ used in the formation of the prevariety (variety) **M** are called *components* of **M**.

We see that the collection of mappings f_i and g_i makes a unified system called a prevariety or quasi-prevariety out of separate logical calculi $\boldsymbol{C}_i$, while the collection of the intersections $\bigcap_{j=1}^{k} f_{ij}(A_{ij})$ and $\bigcap_{j=1}^{k} g_{ij}(T_{ij})$ makes a unified system called a variety out of separate logical calculi $\boldsymbol{C}_i$. For instance, mappings f_i and g_i allow one to establish a correspondence between norms/laws that were used in one country during different periods of time or between norms/laws used in different countries.

The main goal of syntactic logical varieties is in presenting sets of formulas as a structured logical system using logical calculi, which have means for inference and other logical operations. Semantically, it allows one to describe a domain of interest, e.g., a database, knowledge of an individual or the text of a novel, by a syntactic logical variety dividing the domain in parts that allow representation by calculi.

In comparison with varieties and prevarieties, logical quasi-varieties and quasi-prevarieties studied in [5] are not necessarily closed under logical inference. This trait allows better flexibility in knowledge representation.

While syntactic logical varieties and prevarietis synthesize local logics in a unified system, semantic logical varieties and prevarieties studied in [5] unify local mathematics forming a holistic realm of mathematical knowledge.

In addition, syntactic logical varieties and prevarieties found diverse applications to databases and network technology (cf., for example, [6]).

4 PROJECTIVE MATHEMATICS VERSUS REVERSE MATHEMATICS VERSUS CLASSICAL MATHEMATICS

Mathematics suggests an approach for knowledge unification, namely, it is necessary to find axioms that characterize all theories in a specific area and to develop the theory in an axiomatic context. This approach worked well in a variety of mathematical fields.

Axiomatization has been often used in physics (Hilbert's sixth problem refers to axiomatization of branches of physics in which mathematics is prevalent), biology (The most enthusiastic proponent of this approach, the British biologist and logician Joseph Woodger, attempted to formalize the principles of biology—to derive them by deduction from a limited number of basic axioms and primitive terms—using the logical apparatus of the Principia Mathematica by Whitehead and Bertrand Russell, according to Britannica), and some other areas, such as philosophy or technology. It is interesting that the axiomatic approach was also used in areas that are very far from mathematics. For instance, Spinoza used this approach in philosophy, developing his ethical theories and writing his book Ethics in the axiomatic form. More recently, Kunii [20] developed an axiomatic system for cyberworlds.

With the advent of computers, deductive reasoning and axiomatic exposition have been delegated to computers, which performed theorem-proving, while the axiomatic approach has come to software technology and computer science. logical tools and axiomatic description has been used in computer science for different purposes. For instance, Manna [21] built an axiomatic theory of programs, while Milner [23] developed an axiomatic theory of communicating processes. An axiomatic description of programming languages was constructed by Meyer and Halpern [22]. Many researchers have developed different kinds of axiomatic recursion theories (cf., for example [15,19,14,13,29,28]).

However, in classical mathematics, axiomatization has the global character. Mathematicians tried to build a unique axiomatics for the foundations of mathematics. Logicians working in the theory of algorithms tried to find axioms comprising all models of algorithms.

This is the classical approach – axiomatizing the studied domain and then to deduce theorems from axioms. All classical mathematics is based on deduction as a method of logical reasoning and inference. *Deduction* is a type of reasoning processes that construct and/or evaluate *deductive arguments* and where the conclusion follows from the premises with logical necessity. In logic, an argument is called deductive when the truth of the conclusion is purported to follow necessarily or be a logical consequence of the assumptions. Deductive arguments are said to be valid or invalid, but never true or false. A deductive argument is valid if and only if the truth of the conclusion actually does follow necessarily from the assumptions. A valid deductive argument with true assumptions is called sound; a deductive argument which is invalid or has one or more false assumptions or both is called unsound. Thus, we may call classical mathematics by the name *deductive mathematics*.

The goal of deductive mathematics is to deduce theorems from axioms. Deduction of a theorem is also called proving the theorem. When mathematicians cannot prove some interesting and/or important conjecture, creative explorers invent new structures and methods, introducing new axioms to solve the problem. Researchers with a standard thinking try to prove that the problem is unsolvable.

Some consider deductive mathematics as a part of axiomatic mathematics, assuming that deduction (in a strict sense) is possible only in an axiomatic system. Others treat axiomatic mathematics as a part of deductive mathematics, assuming that there are other inference rules besides deduction.

While deductive mathematics is present in and actually dominates all fields of contemporary mathematics, reverse mathematics is the branch of mathematical logic that seeks to determine what are the minimal axioms (formalized conditions) needed to prove the particular theorem [17,18]. This direction in

mathematical logic was founded by [15,16]. The method can briefly be described as going backwards from theorems to the axioms necessary to prove these theorems in some logical system [27]. It turns out that over a weak base theory, many mathematical statements are equivalent to the particular additional axiom needed to prove them. This methodology contrasts with the ordinary mathematical practice where theorems are deduced from a priori assumed axioms.

Reverse mathematics was prefigured by some results in set theory, such as the classical theorem that states that the axiom of choice, well-ordering principle of Zermelo, maximal chain priciple of Hausdorff, and statements of the vector basis theorem, Tychonov product theorem, and Zorn's lemma are equivalent over ZF set theory. The goal of reverse mathematics, however, is to study ordinary theorems of mathematics rather than possible axioms for set theory. A sufficiently weak base theory is adopted (usually, it is a subsystem of second-order arithmetic) and the search is for minimal additional axioms needed to prove some interesting/important mathematical statements. It has been found that in many cases these minimal additional axioms are equivalent to the particular statements they are used to prove.

Projective mathematics is a branch of mathematics similar to reverse mathematics, which aims to determine what are simple conditions needed to prove the particular theorem or to develop a particular theory. However, there are essential differences between these two directions: reverse mathematics is aimed at a logical analysis of mathematical statements, while projective mathematics is directed to making the scope of theoretical statements in general and mathematical statements in particular much larger and extending their applications. As a result, instead of proving similar results in various situations, it becomes possible to prove a corresponding general result in the axiomatic setting and to ascertain validity of this result for a particular case by demonstrating that all axioms (conditions) used in the proof are true for this case. In such a way the general result is projected on different situations. This direction in mathematics was founded by Burgin [9]. This approach contrasts with the conventional (deductive) mathematics where axioms describe some area or type of mathematical structures, while theorems are deduced from a priori assumed axioms.

Projective mathematics has its precursor in such results as extension of many theorems initially proved for numerical functions to functions in metric spaces or generalizations of properties of number systems to properties of groups, rings and other algebraic structures.

Here we use projective mathematics to study algorithms and automata. Our goal is to find some simple properties of algorithms and automata in general, to present these properties in a form of axioms, and to deduce from these axioms theorems that describe much more profound and sophisticated properties of algorithms. This allows one, taking some class **A** of algorithms, not to prove these theorems but only to check if the initial axioms are valid in **A**. If this is the case, then it makes possible to conclude that all corresponding theorems are true for the class **A**. As we know, computer scientists and mathematicians study and utilize a huge variety of different classes and types of algorithms, automata, and abstract machines. Consequently, such an axiomatic approach allows them to obtain many properties of studied algorithms and automata in a simple and easy way.

It is possible to explain goals of classical (deductive) mathematics, reverse mathematics and projective mathematics by means of relations between axioms and theorems.

A set *A* of axioms can be:

1. *Consistent* with some result (theorem) *T*, i.e., when the theorem *T* is added as a new axiom, the new system remains consistent, allowing in some cases to deduce (prove) this theorem.

2. *Sufficient* for some result (theorem) *T*, i.e., it is possible to deduce (prove) the theorem *T* using axioms from *A*.

3. *Irreducible* with respect to some result (theorem) *T*, i.e., the system *A* is a minimal set of axiom that allows one to deduce (prove) the theorem *T*.

After the discovery of non-Euclidean geometries, creation of modern algebra and construction of set theory, classical mathematics main interest has been in finding whether a statement *T* has been consistent with a given axiomatic system *A* (the logical goal) and then in proving this statement in the context of *A*. Thus, classical mathematics is concerned with the first relation. Reverse mathematics, as we can see, deals with the third relation.

In contrast to this, projective mathematics is oriented at the second relation. The goal is to find some simple properties of algorithms or automata in general, to present these properties in a form of a system *U* of axioms, and from these axioms, to deduce theorems that describe much more profound properties of algorithms and automata. This allows one, taking some class **A** of algorithms or automata, not to prove these theorems but only to check if all axioms from the system *U* are valid in **A**. If this is the case, then it is possible to conclude that all corresponding theorems are true for the class **A**. As we know, computer scientists and mathematicians study and utilize a huge variety of different classes and types of algorithms, automata, and abstract machines. Consequently, the projective axiomatic approach allows them to obtain many properties of studied algorithms in a simple and easy way. In such a way, the axiom system *U* provides a definite perspective on different classes and types of algorithms, automata, and abstract machines.

It is interesting that Bernays had a similar intuition with respect to axioms in mathematics, regarding them not as a system of statements about a subject matter but as a system of conditions for what might be called a relational structure. He wrote [2]:

“A main feature of Hilbert’s axiomatization of geometry is that the axiomatic method is presented and practiced in the spirit of the abstract conception of mathematics that arose at the end of the nineteenth century and which has generally been adopted in modern mathematics. It consists in abstracting from the intuitive meaning of the terms . . . and in understanding the assertions (theorems) of the axiomatized theory in a hypothetical sense, that is, as holding true for any interpretation . . . for which the axioms are satisfied. Thus, an axiom system is regarded not as a system of statements about a subject matter but as a system of conditions for what might be called a relational structure . . . [On] this conception of axiomatics, . . . logical reasoning on the basis of the axioms is used not merely as a means of assisting intuition in the study of spatial figures; rather, logical dependencies are considered for their own sake, and it is insisted that in reasoning we should rely only on those properties of a figure that either are explicitly assumed or follow logically from the assumptions and axioms.”

It is possible to formalize the approach of projective mathematics using logical varieties. Indeed, let us take a collection C of postulates, axioms and conditions, which are formalized in a logical language as axioms. This allows us to assume that we have a logical variety M that represents a given domain D in a formal mathematical setting and contains the set C. For instance, the domain D consists of a system of algorithmic models so that the logic of each model D_i is a component M_i of M. Then we deduce a theorem T from the statements from C. Then instead of proving the theorem T for each domain D_i, we check whether $C \subseteq M_i$. When this is true, we conclude that the theorem T belongs to the component M_i because M_i is a calculus and thus, the theorem T is valid for the model D_i. Because C usually consists of simple statements, to check the inclusion $C \subseteq M_i$ is simpler than to prove T in M_i.

5 HOW TO NAVIGATE IN THE ALGORITHMIC MULTIVERSE

It is possible to see that for a conformist, it is much easier to live in the closed algorithmic universe because all possible and impossible actions, as well as all solvable and insolvable problems can be measured against one of the most powerful and universal in the algorithmic universe classes of algorithms. Usually it has been done utilizing Turing machines.

Open world provides much more opportunities for actions and problem solving, but at the same time, it demands more work, more efforts and even more imagination for solving problems insolvable in the closed algorithmic universe. Even the closed algorithmic universe contains many classes and types of algorithms, which have been studied with a reference to a universal class of recursive algorithms. In some cases, partial recursive functions have been used. In other cases, unrestricted grammars have been employed. The most popular have been utilization of Turing machines. A big diversity of new and old classes of algorithms exist that demand specific tools for exploration.

Mathematics has invented such tools and one of the most efficient for dealing with diversity is the axiomatic method. This method has been also applied to the theory of algorithms, automata and computation when the axiomatic theory of algorithms, automata and computation was created [9]. In it, many profound properties of algorithms are derived based on some simple basic conditions (in the form of postulates, axioms and conditions). Namely, instead of proving similar results in various situations, it becomes possible to prove a necessary general result in the axiomatic setting and then to ascertain validity of this result for a particular case by demonstrating that all axioms (conditions) used in the proof are true for this case. In such a way the general result is projected on different situations. For instance, the theorem on undecidability of the Fixed Output Problem proved in [9] has more than 30 corollaries for various classes of algorithms, including the famous theorem about undecidability of the halting problem for Turing machines. Another theorem on recognizability of the Fixed Output Problem proved in [9] has more than 20 corollaries for various classes of algorithms, such as Turing machines, random access machines, Kolmogorov algorithms, Minsky machines, partial recursive functions, inductive Turing machines of the first order, periodic evolutionary Turing machines and limiting partial recursive functions.

The axiomatic context allows a researcher to explore not only individual algorithms and separate classes of algorithms and automata but also classes of classes of algorithms, automata, and computational processes. As a result, axiomatic approach goes higher in the hierarchy of computer and network models, reducing in such a way complexity of their study. The suggested axiomatic methodology is applied to evaluation of possibilities of computers, their software and their networks with the main emphasis on such properties as computability, decidability, and acceptability. In such a way, it became possible to derive various characteristics of types of computers and software systems from the initial postulates, axioms and conditions.

It is also worth mentioning that the axiomatic approach allowed researchers to prove the Church-Turing Thesis for an algorithmic class that satisfies very simple initial axioms [3,12]. These axioms form a system C considered in the previous section and this system provides a definite perspective on different classes of algorithms, ensuring that in these classes the Church-Turing Thesis is true, i.e., it is a theorem.

Moreover, the axiomatic approach is efficient in exploring features of innovative hardware and unconventional organization.

It is interesting to remark that algorithms are used in mathematics and beyond as constructive tools of cognition. Algorithms are often opposed to non-constructive, e.g., descriptive, methods used in mathematics. Axiomatic approach is essentially descriptive because axioms describe properties of the studied objects in a formalized way.

Constructive mathematics is distinguished from its traditional counterpart, axiomatic classical mathematics, by the strict interpretation of the expression “there exists” (called in logic the *existential quantifier*) as “we can construct” and show the way how to do this. Assertions of existence should be backed up by constructions, and the properties of mathematical objects should be decidable in finitely many steps.

However, in some situations, descriptive methods can be more efficient than constructive tools. That is why descriptive methods in the form of the axiomatic approach came back to the theory of algorithms and computation, becoming efficient tool in computer science.

6 CONCLUSIONS

This paper demonstrated the role of the axiomatic methods for different paradigms of mathematics.

Classical mathematics utilizes global axiomatization and classical logic.

Local mathematics utilizes local axiomatization, diverse logics and logical varieties.

Reverse mathematics utilizes axiomatic properties decomposition and backward inference.

Projective mathematics utilizes view axiomatization, logical varieties and properties proliferation.

Here we considered only some consequences of new trends in the axiomatic approach to human cognition in general and mathematical cognition in particular. It would be interesting to study other consequences.

An important direction for future work is to study hardware systems and information processing architectures by applying the axiomatic methods of the mathematical theory of information technology [8].

REFERENCES

[1] J. L. Bell, From absolute to local mathematics, *Synthese*, Volume 69, Number 3, 409-426 (1986)

[2] P. Bernays, D. Hilbert, in *The encyclopedia of philosophy*, v. 3, New York, Macmillan publishing company and The Free Press, 496-504 (1967)

[3] U. Boker, and N. Dershowitz, A. Formalization of the Church-Turing Thesis for State-Transition Models. (2004)

[4] M. Burgin, Knowledge in Intelligent Systems, in *Proceedings of the Conference on Intelligent Management Systems*, Varna, Bulgaria, pp. 281-286 (1989)

[5] M. Burgin, Logical varieties and covarieties, in *Theoretical Problems of Mathematics and Information Sciences*, Ukrainian Academy of Information Sciences, Kiev, pp. 18-34 (1997) (in Russian)

[6] M. Burgin, Logical Tools for Program Integration and Interoperability, in Proceedings of the IASTED International Conference on Software Engineering and Applications, pp.743--748, MIT, Cambridge (2004)

[7] M. Burgin, *Super-recursive Algorithms*, Springer, New York/Heidelberg/Berlin (2005)

[8] M. Burgin, Mathematical Theory of Information Technology, in *Proc*. 8th *WSEAS International Conference on Data Networks, Communications, Computers* (DNCOCO'09), Baltimore, Maryland, pp. 42 - 47 (2009)

[9] M. Burgin, *Measuring Power of Algorithms, Computer Programs, and Information Automata*, Nova Science Publishers, New York, (2010)

[10] M. Burgin, and C.N.J. de Vey Mestdagh, *The Representation of Inconsistent Knowledge in Advanced Knowledge Based Systems*, Lecture Notes in Computer Science, Knowlege-Based and Intelligent Information and Engineering Systems, v. 6882, pp. 524-537 (2011).

[11] J.P. Delgrande, and J. Mylopoulos, Knowledge Representation: Features of Knowledge. In: "Fundamentals of Artificial Intelligence". Springer Verlag, Berlin-New York–Tokyo, pp. 3-38 (1986)

[12] N. Dershowitz, and Y. A. Gurevich, Natural Axiomatization of Computability and Proof of Church's Thesis, *Bulletin of Symbolic Logic*, v. 14, No. 3, pp. 299-350 (2008)

[13] Ershov, A.P. (1981) Abstract computability on algebraic structures, in *Algorithms in modern mathematics and computer science*, Springer, Berlin, pp. 397–420

[14] J.E. Fenstad, Computation theories: An axiomatic approach to recursion on general structures, *Lecture Notes in Mathematics*, Springer Berlin / Heidelberg, pp. 143-168 (1975)

[15] H. Friedman, Axiomatic recursive function theory, in *Logic Colloquium* '69, Studies in Logic, North Holland, Amsterdam, pp. 113-137 (1971)

[16] H. Friedman, Systems of second order arithmetic with restricted induction, I, II (Abstracts), *Journal of Symbolic Logic*, v. 41, pp. 557–559 (1976)

[17] H. Friedman and J. Hirst, Weak comparability of well orderings and reverse mathematics, *Annals of Pure and Applied Logic*, v. 47, pp. 11-29 (1990)

[18] M. Giusto, and S.G. Simpson, Located Sets and Reverse Mathematics, *Journal of Symbolic Logic*, v. 65, No. 3, pp. 1451-1480 (2000)

[19] Gritliot, T. J. (1974) Dissecting abstract recursion, in *Generalized Recursion Theory*, North-Holland, Amsterdam, pp. 405-420

[20] T. L. Kunii, The Potentials of Cyberworlds - An Axiomatic Approach, *Third International Conference on Cyberworlds* (CW'04), Tokyo, Japan, pp. 2-7 (2004)

[21] Z. Manna, *Mathematical Theory of Computation*, McGraw Hill, (1974)

[22] A.R. Meyer and J.Y. Halpern, Axiomatic definitions of programming languages: a theoretical assessment (preliminary report), in *Proceedings of the 7th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, Las Vegas, Nevada, pp. 203-212 (1980)

[23] M. Milner, *Communication and concurrency*, Prentice Hall, New York/London/Toronto, (1989)

[24] M. Minsky, A Framework for Representing Knowledge, MIT, Cambridge (1974)

[25] M. Minsky, Society of Mind: A Response to Four Reviews. Artificial Intelligence, v. 48, pp. 371-396 (1991)

[26] M. Minsky, Conscious Machines, in "Machinery of Consciousness", Proceedings, 75th Anniversary Symposium on Science in Society, National Research Council of Canada (1991a)

[27] S.G. Simpson, *Subsystems of Second Order Arithmetic*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin (1999)

[28] D. Skordev, *Computability in Combinatory Spaces*: *An Algebraic Generalization of Abstract First Order Computability*, Kluwer Academic Publishers, Dordrecht-Boston-London (1992)

[29] S. Thompson, Axiomatic Recursion Theory and the Continuous Functionals, *J. Symbolic Logic*, v. 50, No. 2, pp. 442-450 (1985)

[30] C. Ehresmann, "Cahiers de Topologie et Geom. Dif." VIII, 1966: http://archive.numdam.org/ARCHIVE/CTGDC/CTGDC_1966__8_/CTGDC_1966__8__A1_0/CTGDC_1966__8__A1_0.pdf

[31] J. L. Peterson, *Petri Net Theory and the Modeling of Systems.* Prentice Hall. (1981)

All the links accessed at 08 06 2012

From the Closed Universe to an Open World

Mark Burgin[1] **and Gordana Dodig-Crnkovic**[2]

Abstract. There are different aspects and spheres of unconventional computations. In this paper, we analyze philosophical and methodological implications of algorithmic issues of unconventional computations. At first, we describe how the algorithmic universe was developed and analyze why it became closed in the conventional approach to computation. Then we explain how the new models of algorithms changed the algorithmic universe, making it open and allowing higher flexibility and superior creativity. As Gödel undecidability theorems imply, the closed algorithmic universe restricts essential forms of human cognition, while the open algorithmic universe eliminates such restrictions.

1 INTRODUCTION

Development of society is characterized by a tension between tradition and innovation. Tradition sustains society, while innovation moves society forward. Efficient functioning depends on the equilibrium between tradition and innovation. When there is no equilibrium, society declines: too much tradition brings stagnation and often collapse under the pressure of inner or/and outer forces, while too much innovation results in instability and often in rapture.

The same is true for different areas and aspects of society. Here we are interested in computation, which becomes more and more important for society. Tradition in computation is represented by conventional computations, while unconventional computation characterizes the boldest and far-reaching innovations.

It is possible to demarcate three areas in which computation can be unconventional:

1. Novel hardware, e.g., quantum systems, provides material realization for unconventional computation.

2. Novel algorithms, e.g., super-recursive algorithms, provide operational realization for unconventional computation.

3. Novel organization, e.g., evolutionary computation or self-optimizing computation, provides structural realization for unconventional computation.

Here we discuss algorithmic issue of unconventional computation and analyze philosophical and methodological problems related to it, making a distinction between three classes of algorithms: *recursive*, *subrecursive*, and *super-recursive algorithms*.

Each type of *recursive algorithms* form a class in which it is possible to compute exactly the same functions that are computable by Turing machines. Examples of recursive algorithms are partial recursive functions, RAM, von Neumann automata, Kolmogorov algorithms, and Minsky machines.

Each type of *subrecursive algorithms* forms a class that has less computational power than all Turing machines. Examples of subrecursive algorithms are finite automata, primitive recursive functions and recursive functions.

Each type of *super-recursive algorithms* forms a class that has more computational power than all Turing machines. Examples of super-recursive algorithms are inductive and limit Turing machines, limit partial recursive functions and limit recursive functions.

The main problem is that conventional types and models of algorithms make the algorithmic universe, i.e., the world of all existing and possible algorithms, closed because there is a rigid boundary in this universe formed by recursive algorithms, such as Turing machines, and described by the Church-Turing Thesis. This closed system has been overtly dominated by depressing incompleteness results, such as Gödel incompleteness theorems.

Contrary to this, super-recursive algorithms controlling and directing unconventional computations break this boundary bringing people to an open algorithmic universe – world of unbounded creativity, development, and inspiration, putting no limits on human endeavor.

The paper is organized as follows. First, we summarize how the *closed algorithmic universe* was created and what are advantages and disadvantages of living inside such a closed universe. Next, we describe the breakthrough brought about by the creation of super-recursive algorithms. In Section 4, we analyze implications for people's cognition brought forth by super-recursive algorithms. The main effect is the immense growth of cognitive possibilities. …

2 THE CLOSED UNIVERSE OF TURING MACHINES AND OTHER RECURSIVE ALGORITHMS

Having an extensive experience of problem solving, mathematicians understood that solutions were based on various algorithms. That is why when they more and more encountered problems that they were not able to solve, mathematicians and especially experts in mathematical logic came to the conclusion that it was necessary to develop a rigorous mathematical concept of algorithm. Being, as always, very creative, mathematicians have suggested a diversity of exact mathematical models of algorithm as a general concept. The first models were λ-*calculus* developed by Church in 1931 – 1933, *general recursive functions* introduced by Gödel in 1934, ordinary *Turing machines* constructed by Turing in 1936 and in a less explicit form by Post in 1936, and *partial recursive functions* built by Kleene in 1936. Creating λ-calculus, Church was developing a logical theory of functions and suggested a formalization of the notion of computability by means of λ-definability. In 1936, Kleene demonstrated that λ-definability is computationally equivalent to general recursive functions. In 1937, Turing showed that λ-definability is computationally equivalent to Turing machines. Church was so impressed by these results that

[1] Dept. of Mathematics, UCLA, Los Angeles, USA. Email: mburgin@math.ucla.edu

[2] School of Innovation, Design and Engineering, Mälardalen University, Sweden. Email: gordana.dodig-crnkovic@mdh.se

he suggested what was later called the Church-Turing thesis. Turing formulated a similar conjecture in the Ph.D. thesis that he wrote under Church's supervision.

It is interesting to know that the theory of Frege [8] actually contains λ-calculus. So, there were chances to develop a theory of algorithms and computability in the 19th century. However, at that time the mathematical community did not feel a need in such a theory and probably, would not accept it if somebody created it.

The Church-Turing thesis explicitly engineered a rigid boundary for the algorithmic universe, making this universe closed by Turing machines. Any algorithm from this universe was inside that boundary.

After the first breakthrough, other mathematical models of algorithms were suggested. They include a variety of Turing machines: *multihead, multitape Turing machines, Turing machines with n-dimensional tapes, nondeterministic, probabilistic*, *alternating* and *reflexive Turing machines*, *Turing machines with oracles*, *Las Vegas Turing machines*, etc.; *neural networks* of various types – *fixed-weights, unsupervised, supervised, feedforward,* and *recurrent neural networks*; *von Neumann automata* and general *cellular automata*; *Kolmogorov algorithms finite automata* of different forms – *automata without memory, autonomous automata, automata without output or accepting automata, deterministic, nondeterministic*, *probabilistic automata*, etc.; *Minsky machines*; *Storage Modification Machines* or simply, *Shönhage machines*; *Random Access Machines* (RAM) and their modifications - *Random Access Machines with the Stored Program* (RASP), *Parallel Random Access Machines* (PRAM); *Petri nets* of various types – *ordinary* and ordinary *with restrictions*, *regular, free, colored*, and *self-modifying Petri nets*, etc.; *vector machines*; *array machines*; *multidimensional structured model of computation and computing systems*; *systolic arrays*; *hardware modification machines*; *Post productions*; *normal Markov algorithms*; *formal grammars* of many forms – *regular, context-free, context-sensitive, phrase-structure*, etc.; and so on. As a result, the theory of algorithms, automata and computation has become one of the foundations for computer science.

In spite of all differences between and diversity of algorithms, there is a unity in the system of algorithms. While new models of algorithm appeared, it was proved that any of them could not compute more functions than the simplest Turing machine with a one-dimensional tape. All this give more and more evidence to validity of the Church-Turing Thesis.

Even more, all attempts to find mathematical models of algorithms that were stronger than Turing machines were fruitless. Equivalence to Turing machines has been proved for many models of algorithms. That is why the majority of mathematicians and computer scientists have believed that the Church-Turing Thesis was true. Many logicians assume that the Thesis is an axiom that does not need any proof. Few believe that it is possible to prove this Thesis utilizing some evident axioms. More accurate researchers consider this conjecture as a law of the theory of algorithms, which is similar to the laws of nature that might be supported by more and more evidence or refuted by a counter-example but cannot be proved.

Besides, the Church-Turing Thesis is extensively utilized in the theory of algorithms, as well as in the methodological context of computer science. It has become almost an axiom. Some researchers even consider this Thesis as a unique absolute law of science, or more exactly, computer science.

Thus, we can see that the initial aim of mathematicians was to build a closed algorithmic universe, in which a universal model of algorithm provided a firm foundation and as it was found later, a rigid boundary for this universe.

It is possible to see the following advantages and disadvantages of the closed algorithmic universe.

Advantages:

1. Turing machines and partial recursive functions are feasible mathematical models.
2. These and other recursive models of algorithms provide an efficient possibility to apply mathematical technique.
3. The closed algorithmic universe allowed mathematicians to build beautiful theories of Turing machines, partial recursive functions and some other recursive and subrecursive algorithms.
4. The closed algorithmic universe provides sufficiently exact boundaries for knowing what is possible to achieve with algorithms and what is impossible.
5. The closed algorithmic universe provides a common formal language for researchers.
6. For computer science and its applications, the closed algorithmic universe provides a diversity of mathematical models with the same computing power.

Disadvantages:

1. The main disadvantage of this universe is that its main principle - the Church-Turing Thesis - is not true.
2. The closed algorithmic universe restricts applications and in particular, mathematical models of cognition.
3. The closed algorithmic universe does not correctly reflect computing practice.

3 THE OPEN WORLD OF SUPER-RECURSIVE ALGORITHMS

In opposition to the general opinion, some researchers expressed their concern for the Church-Turing Thesis. As Nelson writes [13], "*Although Church-Turing Thesis has been central to the theory of effective decidability for fifty years, the question of its epistemological status is still an open one*." There were also researchers who directly suggested arguments against validity of the Church-Turing Thesis. For instance, Kalmar [11] raised intuitionistic objections, while Lucas and Benacerraf discussed objections to mechanism based on theorems of Gödel that indirectly threaten the Church-Turing Thesis. In 1972, Gödel's observation entitled "A philosophical error in Turing's work" was published where he declared that: "Turing in his 1937, p. 250 (1965, p. 136), gives an argument which is supposed to show that mental proce- dures cannot go beyond mechanical procedures. However, this argument is inconclusive. What Turing disregards completely is the fact that mind, in its use, is not static, but constantly developing, i.e., that we understand abstract terms more and more precisely as we go on using them, and that more and more abstract terms enter the sphere of our understanding. There may exist systematic methods of actualizing this development, which could form part of the procedure. Therefore, although at each stage the number and precision of the abstract terms at our disposal may be finite, both (and, therefore, also Turing's number of distinguishable states of mind) may converge toward infinity in the course of the application of the procedure." [10]

Thus, pointing that Turing disregarded completely the fact that mind, in its use, is not static, but constantly developing, Gödel predicted necessity for super-recursive algorithms that realize inductive and topological computations [5]. Recently, Sloman [6] explained why recursive models of algorithms, such as Turing machines, are irrelevant for artificial intelligence.

Even if we abandon theoretical considerations and ask the practical question whether recursive algorithms provide an adequate model of modern computers, we will find that people do not see correctly how computers are functioning. An analysis demonstrates that while recursive algorithms gave a correct theoretical representation for computers at the beginning of "computer era", super-recursive algorithms are more adequate for modern computers. Indeed, at the beginning, when computers appeared and were utilized for some time, it was necessary to print out data produced by computer to get a result. After printing, the computer stopped functioning or began to solve another problem. Now people are working with displays and computers produce their results mostly on the screen of a monitor. These results on the screen exist there only if the computer functions. If this computer halts, then the result on its screen disappears. This is opposite to the basic condition on ordinary (recursive) algorithms that implies halting for giving a result.

Such big networks as Internet give another important example of a situation in which conventional algorithms are not adequate. Algorithms embodied in a multiplicity of different programs organize network functions. It is generally assumed that any computer program is a conventional, that is, recursive algorithm. However, a recursive algorithm has to stop to give a result, but if a network shuts down, then something is wrong and it gives no results. Consequently, recursive algorithms turn out to be too weak for the network representation, modeling and study.

Even more, no computer works without an operating system. Any operating system is a program and any computer program is an algorithm according to the general understanding. While a recursive algorithm has to halt to give a result, we cannot say that a result of functioning of operating system is obtained when computer stops functioning. To the contrary, when the operating system does not work, it does not give an expected result.

Looking at the history of unconventional computations and super-recursive algorithms we see that Turing was the first who went beyond the "Turing" computation that is bounded by the Church-Turing Thesis. In his 1938 doctoral dissertation, Turing introduced the concept of a *Turing machine with an oracle*. This, work was subsequently published in 1939. Another approach that went beyond the Turing-Church Thesis was developed by Shannon [17], who introduced the *differential analyzer*, a device that was able to perform continuous operations with real numbers, and namely, such as operation of differentiation. However, mathematical community did not accept operations with real numbers as tractable because irrational numbers do not have finite numerical representations.

In 1957, Grzegorczyk introduced a number of equivalent definitions of computable real functions. Three of Grzegorczyk's constructions have been extended and elaborated independently to super-recursive methodologies: the, so-called, *domain approach* [18,19], *type* 2 *theory of effectivity* or *type* 2 *recursion theory* [20,21], and the *polynomial approximation approach* [22]. In 1963, Scarpellini introduced the class $\mathbf{M}_1$ of functions that are built with the help of five operations. The first three are elementary: substitutions, sums and products of functions. The two remaining operations are performed with real numbers: integration over finite intervals and taking solutions of Fredholm integral equations of the second kind.

Another type of super-recursive algorithms was introduced in 1965 by Gold and Putnam, who brought in concepts of *limiting recursive function* and *limiting partial recursive function*. In 1967, Gold produced a new version of limiting recursion, also called *inductive inference*, and applied it to problems of learning. Now inductive inference is a fruitful direction in machine learning and artificial intelligence. One more direction in the theory of super-recursive algorithms emerged in 1967 when Zadeh introduced *fuzzy algorithms*. It is interesting that limiting recursive function and limiting partial recursive function were not considered as valid models of algorithms even by their authors. A proof that fuzzy algorithms are more powerful than Turing machines was obtained much later (Wiedermann, 2004). Thus, in spite of existence of super-recursive algorithms, researchers continued to believe in the Church-Turing Thesis as an absolute law of computer science.

After the first types of super-recursive models had been studied, a lot of other super-recursive algorithmic models have been created: *inductive Turing machines*, *limit Turing machines*, *infinite time Turing machines*, *general Turing machines*, *accelerating Turing machines*, *type* 2 *Turing machines*, *mathematical machines*, δ-Q-*machines*, *general dynamical systems*, *hybrid systems*, *finite dimensional machines* over real numbers, ***R****-recursive functions* and so on.

However, the first publication where it was explicitly stated and proved that there are algorithms more powerful than Turing machines was [2].

The closest to conventional algorithms are inductive Turing machines of the first order because they work with constructive objects, all steps of their computation are the same as the steps of conventional Turing machines and the result is obtained in a finite time. In spite of these similarities, inductive Turing machines of the first order can compute much more than conventional Turing machines.

Inductive Turing machines of the first order form only the lowest level of super-recursive algorithms. There infinitely more levels and as a result, the algorithmic universe becomes open. Taking into consideration algorithmic schemas, which go beyond super-recursive algorithms, we come to an open world of information processing, which includes the algorithmic universe. Openness of this world has many implications for human cognition in general and mathematical cognition in particular. For instance, it is possible to demonstrate that not only computers but also the brain can work not only in the recursive mode but also in the inductive mode, which is essentially more powerful and efficient. Some of them are considered in the next section.

4 ABSOLUTE PROHIBITION IN THE CLOSED UNIVERSE AND INFINITE OPPORTUNITIES IN THE OPEN WORLD

To provide sound and secure foundations for mathematics, David Hilbert proposed an ambitious and wide-ranging program in the philosophy and foundations of mathematics. His approach formulated in 1921 stipulated two stages. At first, it was

necessary to formalize classical mathematics as an axiomatic system. Then, using only restricted, "finitary" means, it was necessary to give proofs of the consistency of this axiomatic system.

Achieving a definite progress in this direction, Hilbert became very optimistic. In his speech in Königsberg in 1930, he made a very famous statement:

Wir müssen wissen. Wir werden wissen.
(*We must know*. *We will know*.)

Next year the Gödel undecidability theorems were published [9]. They undermined Hilbert's statement and his whole program. Indeed, the first Gödel undecidability theorem states that it is impossible to validate truth for all true statements about objects in an axiomatic theory that includes formal arithmetic. This is a consequence of the fact that it is impossible to build all sets from the arithmetical hierarchy by Turing machines. In such a way, the closed Algorithmic Universe imposed restriction on the mathematical exploration. Indeed, rigorous mathematical proofs are done in formal mathematical systems. As it is demonstrated (cf., for example, [7]), such systems are equivalent to Turing machines as they are built by means of Post productions. Thus, as Turing machines can model proofs in formal systems, it is possible to assume that proofs are performed by Turing machines.

The second Gödel undecidability theorem states that for an effectively generated consistent axiomatic theory *T* that includes formal arithmetic and has means for formal deduction, it is impossible to prove consistency of *T* using these means.

From the very beginning, Gödel undecidability theorems have been comprehended as absolute restrictions for scientific cognition. That is why Gödel undecidability theorems were so discouraging that many mathematicians consciously or unconsciously disregarded them. For instance, the influential group of mostly French mathematicians who wrote under the name Bourbaki completely ignored results of Gödel [12]. It is possible to suggest that the reason was not essentially rational but mostly psychological.

However, later researchers came to the conclusion that these theorems have such drastic implications only for formalized cognition based on rigorous mathematical tools. For instance, in the 1964 postscript, Gödel wrote that undecidability theorems "do not establish any bounds for the powers of human reason, but rather for the potentialities of pure formalism in mathematics."

Discovery of super-recursive algorithms and acquisition of the knowledge of their abilities drastically changed understanding of the Gödel's results. Being a consequence of the closed nature of the closed algorithmic universe, these undecidability results loose their fatality in the open algorithmic universe. They become relativistic being dependent on the tools used for cognition. For instance, the first undecidability theorem is equivalent to the statement that it is impossible to compute by Turing machines or other recursive algorithms all levels of the Arithmetical Hierarchy [15]. However, as it was stated in [3], there was a hierarchy of inductive Turing machines so that all levels of the Arithmetical Hierarchy were computable and even decidable by these inductive Turing machines. Complete proofs of these results were published only in 2003 due to the active opposition of the proponents of the Church-Turing Thesis [4].

This makes the Gödel's results relative to the means used for proving mathematical statements because decidability of the Arithmetical Hierarchy implies decidability of the formal arithmetic. For instance, the first Gödel undecidability theorem is true when recursive algorithms are used for proofs but it becomes false when inductive algorithms are utilized. It was demonstrated, for example, in 1936 by Gentzen, who in contrast to the second Gödel undecidability theorem, proved consistency of the formal arithmetic using ordinal induction.

5 THE OPEN WORLD AND THE INTERNET

As we all know, the *open world*, or more exactly, the *open world of knowledge*, is an important concept for the knowledge economy. According to Rossini [16], it emerges from a world of pre-Internet political systems, but it has come to encompass an entire worldview based on the transformative potential of open, shared, and connected technological systems. The idea of an open world synthesizes much of the social and political discourse around modern education and scientific endeavor and is at the core of the *Open Access* (OA) and *Open Educational Resources* (OER) movements. While the term *open society* comes from international relations, where it was developed to describe the transition from political oppression into a more democratic society, it is now being appropriated into a broader concept of an open world connected via technology [16]. The idea of openness in access to knowledge and education is a reaction to the potential afforded by the global networks, but is inspired by the sociopolitical concept of the open society.

Open Access (OA) is a knowledge-distribution model by which scholarly, peer-reviewed journal articles and other scientific publications are made freely available to anyone, anywhere over the Internet. It is the foundation for the open world of scientific knowledge, and thus, a principal component of the open world of knowledge as a whole. In the era of print, open access was economically and physically impossible. Indeed, the lack of physical access implied the lack of knowledge access - if one did not have physical access to a well-stocked library, knowledge access was impossible. The Internet has changed all of that, and OA is a movement that recognizes the full potential of an open world metaphor for the network.

In OA, the old tradition of publishing for the sake of inquiry, knowledge, and peer acclaim and the new technology of the Internet have converged to make possible an unprecedented public good: "the world-wide electronic distribution of the peer-reviewed journal literature" [1].

The open world of knowledge is based on the Internet, while the Internet is based on computations that go beyond Turing machines. One of the basic principles of the Internet is that it is always on, always available. Without these features, the Internet cannot provide the necessary support for the open world of knowledge because ubiquitous availability of knowledge resources demands non-stopping work of the Internet. However, as it is proved in [5], if an automatic (computer) system works without halting, gives results in this mode and can simulate any operation of a universal Turing machine, then this automatic (computer) system is more powerful than any Turing machine. This means that this automatic (computer) system, in particular, the Internet, performs unconventional computations.

6 CONCLUSIONS

This paper shows how the universe (world) of algorithms became open with the discovery of super-recursive algorithms, providing more tools for human cognition and artificial intelligence.

Here we considered only some consequences of the open world environment for human cognition in general and mathematical cognition in particular. It would be interesting to study other consequences of coming to an open world of algorithms and computation.

It is known that not all quantum mechanical events are Turing-computable. So, it would be interesting to find a class of super-recursive algorithms that compute all such events or to prove that such a class does not exist.

It might be interesting to contemplate relations between the Open Algorithmic Universe and the Open Science in the sense of Nielsen [14]. For instance, one of the pivotal features of the Open Science is accessibility of research results on the Internet. At the same time, as it is demonstrated in [5], the Internet and other big networks of computers are always working in the inductive mode or some other super-recursive mode. Moreover, actual accessibility depends on such modes of functioning.

REFERENCES

[1] Budapest Open Access Initiative: <http://www.soros.org/openaccess/read.shtml>.

[2] M. Burgin, The Notion of Algorithm and the Turing-Church Thesis, In *Proceedings of the VIII International Congress on Logic, Methodology and Philosophy of Science*, Moscow, v. 5, part 1, pp. 138-140 (1987)

[3] M. Burgin, Arithmetic Hierarchy and Inductive Turing Machines, *Notices of the Russian Academy of Sciences*, v. 299, No. 3, pp. 390-393 (1988)

[4] M. Burgin, Nonlinear Phenomena in Spaces of Algorithms, *International Journal of Computer Mathematics*, v. 80, No. 12, pp. 1449-1476 (2003)

[5] Burgin, M. *Super-recursive Algorithms*, Springer, New York/Heidelberg/Berlin (2005)

[6] A. Sloman, The Irrelevance of Turing machines to AI http://www.cs.bham.ac.uk/~axs/ (2002)

[7] R.M. Smullian, *Theory of Formal Systems*, Princeton University Press (1962)

[8] G. Frege, *Grundgesetze der Arithmetik*, Begriffschriftlich Abgeleitet, Viena (1893/1903)

[9] K. Gödel, Über formal unentscheidbare Sätze der Principia Mathematics und verwandter System I, *Monatshefte für Mathematik und Physik*, b. 38, s.173-198 (1931)

[10] K. Gödel, Some Remarks on the Undecidability Results, in G¨odel, K. (1986–1995), Collected Works, v. II, Oxford University Press, Oxford, pp. 305–306 (1972)

[11] L. Kalmar, An argument against the plausibility of Church's thesis, in *Constructivity in mathematics*, North-Holland Publishing Co., Amsterdam, pp. 72-80 (1959)

[12] A.R.D. Mathias, The Ignorance of Bourbaki, Physis Riv. Internaz. Storia Sci (N.S.) 28, pp. 887-904 (1991)

[13] R. J. Nelson, Church's thesis and cognitive science, *Notre Dame J. of Formal Logic*, v. 28, no. 4, 581—614 (1987)

[14] M. Nielsen, Reinventing Discovery: The New Era of Networked Science, Princeton University Press, Princeton and Oxford (2012)

[15] H. Rogers, Theory of Recursive Functions and Effective Computability, MIT Press, Cambridge Massachusetts (1987)

[16] C. Rossini, Access to Knowledge as a Foundation for an Open World, *EDUCAUSE Review*, v. 45, No. 4, pp. 60–68 (2010)

[17] C. Shannon, Mathematical Theory of the Differential Analyzer, *J. Math. Physics*, MIT, v. 20, 337-354 (1941)

[18] S. Abramsky, A. Jung, Domain theory. In S. Abramsky, D. M. Gabbay, T. S. E. Maibaum, editors, (PDF). Handbook of Logic in Computer Science. III. Oxford University Press. (1994).

[19] A. Edalat, Domains for computation in mathematics, physics and exact real arithmetic, Bulletin Of Symbolic Logic, Vol:3, 401-452 (1997)

[20] K. Ko, Computational Complexity of Real Functions, Birkhauser Boston, Boston, MA (1991)

[21] K. Weihrauch, *Computable Analysis. An Introduction*. Springer-Verlag Berlin/ Heidelberg (2000)

[22] M. B. Pour-El, and J. I. Richards, *Computability in Analysis and Physics. Perspectives in Mathematical Logic*, Vol. 1. Berlin: Springer. (1989)

All the links accessed at 08 06 2012