Minds & Machines (2011) 21:301–322
DOI 10.1007/s11023-011-9235-1

Significance of Models of Computation, from Turing Model to Natural Computation

Gordana Dodig-Crnkovic

Received: 19 December 2009 / Accepted: 18 October 2010 / Published online: 2 February 2011

Abstract The increased interactivity and connectivity of computational devices along with the spreading of computational tools and computational thinking across the fields, has changed our understanding of the nature of computing. In the course of this development computing models have been extended from the initial abstract symbol manipulating mechanisms of stand-alone, discrete sequential machines, to the models of natural computing in the physical world, generally concurrent asynchronous processes capable of modelling living systems, their informational structures and dynamics on both symbolic and sub-symbolic information processing levels. Present account of models of computation highlights several topics of importance for the development of new understanding of computing and its role: natural computation and the relationship between the model and physical implementation, interactivity as fundamental for computational modelling of concurrent information processing systems such as living organisms and their networks, and the new developments in logic needed to support this generalized framework. Computing understood as information processing is closely related to natural sciences; it helps us recognize connections between sciences, and provides a unified approach for modeling and simulating of both living and non-living systems.

Keywords Philosophy of computer science · Philosophy of computing · Theory of computation · Hypercomputing · Philosophy of information · Models of computation

---

G. Dodig-Crnkovic (✉)
Computer Science Laboratory, School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden
e-mail: gordana.dodig-crnkovic@mdh.se

Introduction

> “Computer Science has something more to offer to the other sciences than the computer. In particular, in the mathematical and logical understanding of fundamental transdisciplinary scientific concepts such as interaction, concurrency and causality, synchrony and asynchrony, compositional modelling and reasoning, open versus closed systems, qualitative versus quantitative reasoning, operational methodologies, continuous versus discrete, hybrid systems and more, Computer Science is far ahead of many other sciences, due in part to the challenges arising from the amazing rapidity of the technology change and development it is constantly being confronted with. One could claim that computer scientists (maybe without realizing it) constitute an avant-garde for the sciences in terms of providing fresh paradigms and methods.” (Abramsky and Coecke 2007).

Present day computers are very different from the early stand-alone calculators designed for mechanizing mathematical operations. They are largely used for communication in world-wide networks and variety of information processing and knowledge management. Moreover, they play an important role in the control of physical processes and thus connect to the physical world, especially in automation and robotics. Apart from classical engineering and hard-scientific domains, computing has in recent decades entered new fields such as biology and cognitive sciences,[1] humanities and arts—all previously considered as typical non-mechanical domains.

Computational processes are nowadays distributed, reactive, agent-based and concurrent. The main criterion of success of the computation is not its termination, but its response to the outside world, its speed, generality and flexibility; adaptability, and tolerance to noise, error, faults, and damage.

One of the aims of the paper is to highlight the conceptual importance of new models of computation with interaction as the fundamental mechanism,[2] together with the developments in the fields of natural computation and applications of novel approaches in logic—all of which can be seen as generalizations and enrichment of the repertoire of classical computation models.

Generalizing the traditional algorithmic Turing Machine model of computation, in which the computer was an isolated box provided with a suitable algorithm and an input, left alone to compute until the algorithm terminated, interactive computation (Turing 1939,[3] Wegner 1988; Goldin et al. 2006) allows interaction

---

[1] “Cognitive sciences” here refer to “the interdisciplinary study of mind and intelligence, embracing philosophy, psychology, artificial intelligence, neuroscience, linguistics, and anthropology.”, according to Thagard, Paul, “Cognitive Science”, The Stanford Encyclopedia of Philosophy (Summer 2010 Edition), Edward N. Zalta (ed.), URL = <http://plato.stanford.edu/archives/sum2010/entries/cognitive-science/>.

[2] 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.

[3] An extension of the basic Turing machine model introduced by (Turing 1939) which is the extension by oracles which enables new, external, and possibly noncomputable information to enter a computation. Such machines could compute an arbitrary non-recursive function from naturals to naturals. Turing

i.e. communication of the computing process with the environment during the computation. Interaction consequently provides a new conceptualization of computational phenomena which involve communication and information processing.

The important novelty that interactive computing brings about is its articulation of the difference between an open and a closed system, the distinction relevant for physics, biology, mathematics as well as for computing itself. The conventional theories are typically about closed, isolated systems with the environment represented by some average behavior, and treated as a perturbation. An observer is external to the system. In the interactive framework the system is in general communicating with the explicitly expressed environment that also allows for the integration of the observer into the model. (Dodig-Crnkovic 2010).

Even though practical implementations of interactive computing are several decades old, a foundational theory, and in the first place semantics and logic of interactive computing is only in its beginning.[4] A theoretical base analogous to what Turing machines are for algorithmic computing, is under development for interactive computing. (Wegner 1998; Abramsky 2003; Japaridze 2006).

As (Sloman 1996) points out, concurrent and synchronized machines are equivalent to sequential machines, but some concurrent machines are asynchronous. TMs can in principle approximate machines with continuous changes, but cannot implement them exactly. This argument may also be found in (Copeland and Sylvan 1999). A continuous machine with non-linear feedback loops may be chaotic and impossible to approximate discretely, even over short time scales. If a machine is composed of asynchronous concurrently running subsystems and their relative frequencies vary randomly then such a machine cannot be adequately modeled by Turing machine. See also (Dodig-Crnkovic 2006). In general, in place of idealized, symbol-manipulating models, more and more physics-inspired modeling is taking place.

> “In particular, the quantum informatic endeavor is not just a matter of feeding physical theory into the general field of natural computation, but also one of using high-level methods developed in Computer Science to improve on the quantum physical formalism itself, and the understanding thereof. We highlight a seemingly contradictory phenomenon: passing to an abstract, categorical quantum informatics formalism leads directly to a simple and elegant graphical formulation of quantum theory itself, which for example makes the design of some important quantum informatic protocols completely transparent. It turns out that essentially all of the quantum informatic machinery can be recovered from this graphical calculus. But in turn, this

---

Footnote 3 continued
showed that even in those more powerful systems, undecidability still appears. Oracle machines were only mathematical models, and were not thought of as physically realizable. The central ideas which (Cooper 2008) brings to this project are the concepts of definability and invariance in a context of the real-world computation. He is modeling causal relationships based on Turing’s 1939 concept of interactive computation, which Cooper defines over reals.

[4] “Focusing on interaction without representation, concentrating on computation beyond the ’Turing barrier’, without climbing the higher levels which become the subject in a mathematical analysis of the structures of interaction (see e.g. Cooper)”—I am thankful for the anonymous reviewer for this remark.

> graphical formalism provides a bridge between methods of logic and computer science, and some of the most exciting developments in the mathematics of the past two decades "(Abramsky and Coecke 2007).

The same two-way process of learning is visible in biocomputing according to (Rozenberg and Kari 2008).

Living systems are essentially opened and in constant communication with the environment. New computational models must be interactive in order to be applicable to biological and social phenomena and approach richness of their information processing repertoire. This openness and interactivity has direct consequence for the logic. (Goldin and Wegner 2002) argue e.g. that computational logic must be able to model interactive computation, that classical logic does not suffice and that logic must be paraconsistent due to the incompleteness of interaction.

> "Consider a computer which stores a large amount of information. While the computer stores the information, it is also used to operate on it, and, crucially, to infer from it. Now it is quite common for the computer to contain inconsistent information, because of mistakes by the data entry operators or because of multiple sourcing. This is certainly a problem for database operations with theorem-provers, and so has drawn much attention from computer scientists. Techniques for removing inconsistent information have been investigated. Yet all have limited applicability, and, in any case, are not guaranteed to produce consistency. (There is no algorithm for logical falsehood.) Hence, even if steps are taken to get rid of contradictions when they are found, an underlying paraconsistent logic is desirable if hidden contradictions are not to generate spurious answers to queries." (Priest and Tanaka 2004).

There are several new developments in logic that will be mentioned in this account. In general, it is argued that new models of computing presuppose not only a generalized understanding of what computing is and what it might be, but at the same time calls for novel approaches in several related fields. New view of computing goes hand in hand with new understanding of information and logic, physics, and a number of other related fields. What is then the place and role of semantics in this new emerging informational—computational—communicational world? Here is a suggestion for an answer:

> "Computability logic believes that logic is meant to be the most basic, general-purpose formal tool potentially usable by intelligent agents in successfully navigating real life. And it is semantics that establishes that ultimate real-life meaning of logic." (Japaridze 2006).

Information Processing versus Turing Model

The definition of computation is still the subject of a debate. The special issue of the journal Minds and Machines (1994, 4, 4) was devoted to the question "What is Computation?" The most general and closest to the common acceptance is the view of computation as information processing, found in number of mathematical accounts of computing; see (Burgin 1999, 2005) for exposition. Understanding of computation as information processing is also widespread in Biology, Neuroscience, Cognitive science and number of other fields.

For a process to qualify as computation a model must exist such as algorithm, network topology, physical process or in general any mechanism which ensures definability of its behavior.

The characterization of computing can be made in several dimensions by classification into orthogonal types: digital/analog, symbolic/subsymbolic, interactive/batch and sequential/parallel. Nowadays digital computers are used to simulate all sorts of natural processes, including those that in physics are understood as continuous. However, it is important to distinguish between the mechanism of computation and the simulation model.

No matter if the data form any symbols; computation is a process of change of the data/structure. On a fundamental quantum–mechanical level, the universe performs computation on its own (Lloyd 2006). Symbols appear on a much higher level of organization, and always in relation with living organisms. Symbols represent something for a living organism; have a function as carriers of meaning.

The notion of computation as formal (mechanical) symbol manipulation originates from discussions in mathematics in the early twentieth century. The most influential program for formalization was initiated by Hilbert, who in order to reduce mathematics to a finitary formal system, treated formalized reasoning as a symbol game in which the rules of derivation are expressed in terms of the syntactic properties of the symbols. As a result of Hilbert's program large areas of mathematics have been formalized. Formalization means the establishment of the basic language which is used to formulate the system of axioms and derivation rules defined such that the important semantic relationships are preserved by inferences defined only by the syntactic form of the expressions. Hilbert's Grundlagen der Mathematik, and Whitehead and Russell's Principia Mathematica are examples of such formalization. However, there are limits to what can be formalized, as demonstrated by Gödel's incompleteness theorems.

LCMs (Logical Computing Machines, (Turing 1948, p. 7) expression for Turing machines) were an attempt to give a mathematically precise definition of "algorithm" or "mechanical procedure". In Turing's words: "We may compare a man in the process of computing a… number to a machine." (Turing 1936, p. 231.) A thesis concerning the extent of effective procedures that a human being unaided by machinery is capable of carrying out has no implication concerning the extent of the procedures that other computing systems are capable of carrying out. Among a "machine's" (computing physical system's) repertoire of atomic operations there may be those that no human being unaided by "computing machinery" can perform. (Copeland 1997, 1998, 2000, 2002).

The Church-Turing thesis in its original formulation (Church 1935, 1936) says that real-world calculation can be performed using the lambda calculus, which is equivalent to using general recursive functions. The thesis addresses several kinds of computation, such as cellular automata, register machines, and substitution systems. As a matter of fact, the Church-Turing thesis has long served as a definition for computation. There has never been a proof, but the evidence for its validity comes from the equivalence of mentioned computational models.

Copeland and Shagrir (2007) analyze the claim of Turing's student Gandy who argued that any machine satisfying four idealized physical principles is equivalent to some Turing machine. Gandy's four principles thus define a class of computing machines and authors show that the relationship of this class to the class of all (ideal) physical computing machines is not identity as Gandy would have it. They provide examples of (ideal) physical machines that fall outside the class of Gandy machines and compute functions that are not TM computable. Sieg (2007) revising Gandy's approach, formulates conditions of boundedness (a fixed bound on the number of configurations a computer can immediately recognize) and locality (only immediately recognizable sub-configurations can be modified) for two types of calculators—humans acting as computing agents and mechanical computing devices. Sieg argues that the distinctive feature of the machines is that they *can carry out parallel computations*, the idea found even in the work of Copeland.

(Kampis 1991) claims that the Church-Turing thesis applies only to simple systems. According to Kampis, complex biological systems must be modeled as self-referential, self-organizing systems called "component-systems" (self-generating systems), whose behavior, though computational in a generalized sense, goes far beyond the simple Turing machine model.

> "a component system is a computer which, when executing its operations (software) builds a new hardware… [W]e have a computer that re-wires itself in a hardware-software interplay: the hardware defines the software and the software defines new hardware. Then the circle starts again." (Kampis 1991, p. 223.)

I would add an obvious remark. The Turing machine is supposed to be given from the outset—its logic, its physical resources, and the meanings ascribed to its actions. *The Turing Machine essentially presupposes a human as a part of a system*—the human is the one who poses the questions, provides material resources and interprets the answers.

Naturalist Computationalism and Computing Universe

Zuse[5] 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 natural computationalist (pancomputationalist).

> "And how about the entire universe, can it be considered to be a computer? Yes, it certainly can, it is constantly computing its future state from its current state, it's constantly computing its own time-evolution! And as I believe Tom

---

[5] http://www.idsia.ch/~juergen/digitalphysics.htm.

> Toffoli pointed out, actual computers like your PC just hitch a ride on this universal computation!" (Chaitin 2006).

Even (Wolfram 2002) advocates pancomputationalist view, in which complexity of behaviors and structures found in nature are derived from a few basic mechanisms. Natural phenomena are thus the products of computation. In a computational universe new and unpredictable phenomena emerge as a result of simple algorithms operating on simple computing elements—cellular automata, and complexity originates from the bottom-up emergent processes. Cellular automata are equivalent to a universal Turing Machine (Wolframs Rule 110). However, in living organisms not only bottom-up but even top-down phenomena occur. Emergent properties act in the top-down direction constraining constituent elements. Wolfram's critics also remark that cellular automata do not evolve beyond a certain level of complexity. The mechanisms involved do not necessarily demand evolutionary development. Actual physical mechanisms at work in the physical universe appear to be quite different from cellular automata which are very simple explanatory models and as such surprisingly successful ones for a range of phenomena. Ed Fredkin, in Digital Philosophy, suggests that particle physics can emerge from cellular automata. The universe is digital, time and space are discrete, and humans are software running on a universal computer. Wolfram and Fredkin assume that the universe is a discrete system and suitably understood as an all-encompassing digital computer. However, the hypothesis about the discreteness of the physical world is not the decisive one for natural computationalism. As is well known, there are digital as well as analogue computers. There are interesting philosophical connections between digital and analog processes. For example, DNA code (digital) is closely related to protein folding (analog) for its functioning in biological systems. In physics some natural phenomena are discrete, some are continuous. Especially quantum mechanics has dual character where both particles and waves are necessary (Lloyd 2006). Natural computationalism is expected to be able to express both continuous and discrete phenomena.

Pancomputationalism has been criticized as explanatory vacuous, for example by (Piccinini 2007) based on the idea that computation is equivalent to Turing Machine, which is not the general case of computational mechanism as argued in this paper. Some would allege that "even if the pancomputationalism is both non-trivial and true, it doesn't provide any new theory compared to what we already know from physics, so it might be claimed that pancomputationalism is just 'bombastic redescription', a phrase borrowed from Dennett." (I have to thank for this accurate formulation of criticism to one of my anonymous reviewers). Contrary to what many critics claim, there are good reasons to expect increased understanding of the universe in terms of information and computation as basic elements (Info-computationalism), as argued in (Dodig-Crnkovic and Müller 2010) supporting the view that info-computational framework provides a new research programme which gives us 'eine Tieferlegung der Fundamente' (a deepening of the foundations) in words of Hilbert. (Denning 2007) argues that computing today is a natural science and this new generalized computing understood as information processing is closely related to classical natural sciences. It helps us understand connections between them, and provides a unified framework for modeling and simulating of both living and non-living systems. (Dodig-Crnkovic 2010).

Computation as Information Processing

In his programmatic paper Open Problems in the Philosophy of Information Floridi (2004) lists the five most interesting areas of research for the field of Philosophy of Information (and Computation), containing eighteen fundamental questions. Information dynamics is of special interest here, as information processing (computation).

If we accept the natural computationalist stance as a point of departure, and if all physical processes may be expressed as computations, meaning the whole universe might be represented as a network of computing processes at different scales or levels of granularity, then we may see information in the first place as a result of (natural) computation.

Information and computation are two complementary ideas in a similar way to continuum and a discrete set. In its turn continuum—discrete set dichotomy may be seen in a variety of disguises such as: time—space; wave—particle; geometry—arithmetic; interaction—algorithm; computation—information. Two elements in each pair presuppose each other, and are inseparably related to each other. (Dodig-Crnkovic 2006; Dodig-Crnkovic and Stuart 2007; Dodig-Crnkovic and Müller 2010) The field of Philosophy of Information is so closely interconnected with the Philosophy of Computation that it would be appropriate to call it Philosophy of Information and Computation, having in mind the dual character of information-computation. Burgin (2005) puts it in the following way:

> "It is necessary to remark that there is an ongoing synthesis of computation and communication into a unified process of information processing. Practical and theoretical advances are aimed at this synthesis and also use it as a tool for further development. Thus, we use the word computation in the sense of information processing as a whole. Better theoretical understanding of computers, networks, and other information processing systems will allow us to develop such systems to a higher level".

The traditional mathematical theory of computation is the theory of algorithms. Ideal, theoretical computers are mathematical objects and they are equivalent to algorithms, or abstract automata, (Turing machines), or effective procedures, or recursive functions, or formal languages. New envisaged future computers are information processing devices. That is what makes the difference. Syntactic mechanical symbol manipulation is replaced by information (both syntactic and semantic) processing. Compared to 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 is a special case of Einstein's theory, or the Euclidean geometry is a limit case of non-Euclidean geometries.

According to Burgin (2005), there are three distinct components of information processing systems: hardware (physical devices), software (programs that regulate its functioning) and infoware which represents information processed by the system. Infoware is a shell built around the software-hardware core which was the traditional domain of automata and algorithm theory. Communication of information and knowledge takes place on the level of infoware.

Both computation and communication imply the transition, transformation and preservation of information. (Bohan Broderick 2004) compares notions of communication and computation which leads him to the conclusion that the two are often not conceptually distinguishable. He shows how computation and communication may be distinguished if computation is limited to actions within a system and communication is an interaction between a system and its environment. The interesting problem of distinction arises when the computer is conceived as an open system in communication with the environment, where the boundary is dynamic, as in biological computing.

Complexity, Computing and Hypercomputation

As computers are used not only for calculation and modeling but also for interaction (in real-time) with the physical world, computation process must match and directly connect to its environments. According to (Ashby 1964) it is therefore necessary for a computation to correspond to the complexity of the environment. Ashby's "Law of Requisite Variety" states namely, that to control a situation and to perform up to requirements, the variety of system responses must at least match the variety of behaviors of an environment that it controls. This amounts to the claim that in order for a computer to achieve an adequate control of a complex system, the complexity of the repertoire of its responses must match the complexity of the environment.

Our present day information and communication technology is based on algorithms. The Church-Turing thesis is the basis of the algorithmic model that claims that all of computation can be expressed by recursive algorithms (Turing machines).

Generally speaking, the semantics of a model are relative to a domain of application and they are usually not well-defined outside that domain (Kuipers 2006) gives some interesting examples of the domain dependence of theory). Even the traditional model of computation has its domain, and the discussion of the presuppositions and the context of the Turing machine model are therefore in order. In spite of its validity within a given domain, the Turing machine model does not suffice for certain important present applications such as parallel asynchronous computing.

As already mentioned, it has been argued that Turing computation is what we mean by computation, but MacLennan proposes a broader definition of computation that includes both Turing computation and alternative (in Burgin's terminology super-recursive) hypercomputing models.

From the perspective of natural computationalism, if we compare Turing machines with the physical universe, including quantum physics, the latter exhibits much higher order of complexity. That would imply that we would need more powerful computers than what is represented by Turing machines to efficiently represent, simulate and control the real world phenomena.

In order to surpass Turing models limitations, the theory of hypercomputation (such as super-recursive algorithms) addresses two distinct problems (Burgin 2005): the nature of the computing mechanism and the nature of the halting problem.

The first problem could be answered by natural computation, see next chapter. Computing is used to not only calculate but also simulate phenomena, which is best done by natural computation in the case of natural phenomena.

The second question is answered by the insight that computing process in general has no need of halting. The Internet neither computes any function nor is it expected to halt. Another way to see the stop problem is thinking of the original question of *uncomputability as the internalized problem of induction*, (Kelly 2004). Induction in a sense of learning process is stopped at a certain point, decided on semantic (pragmatic) grounds. The result will not be a perfect answer, but a good enough one. From idealized logical machines providing perfectly correct general solutions to idealized problems, computing devices develop towards embodied real-world machinery providing good enough solutions to specific problems.

The field of 'hypercomputation', i.e. non-Turing computation, was introduced by (Copeland and Proudfoot 1999),[6] who mention two Turing's 'forgotten ideas'—a 1948 proposal for developing *networks of unorganised elements* (the forerunners of the neural networks) and interactive oracle machines which introduce a *non-mechanical* logical element into computing. Hypercomputation has developed in several directions (for an overview, see Copeland 2002) and a special issue of the journal Applied Mathematics and Computation 178 (2006). Bringsjord and Zenzen (2002) claim:

> "In fact, just as there are an infinite number of mathematical devices equivalent to Turing machines (machines running programs from the language L visited above, Register machines, the λ-calculus, abaci,..; these are all discussed in the context of an attempt to define computation in Bringsjord (1994)), there are an infinite number of devices beyond the Turing Limit. As you might also guess, a small proper subset of these devices dominates the literature. In fact, three kinds of hypercomputational devices—analog chaotic neural nets, trial-and error machines, and Zeus machines—are generally featured in the literature."

According to (Burgin 2005), the most important types of hypercomputation listed in chronological order are: inductive computations and inference; computations and recursive functions with real numbers; interactive and concurrent computations; topological computations; infinite time computations, and neural networks with real number parameters (Siegelman 1999). Each of these computational models brings with it a new logic of computation.

---

[6] (Hodges 2009) criticizes proposal by (Copeland and Proudfoot 1999) to use a physical device as an oracle, as Turing in (Turing 1939, p. 173) said: 'We shall not go any further into the nature of this oracle apart from saying that it cannot be a machine.'

Natural Computation

Natural computation is a study of computational systems including the following:

1) *Computing techniques that take inspiration from nature for the development of novel problem-solving methods* (artificial neural networks, swarm intelligence, artificial immune systems, computing on continuous data, membrane computing, artificial life, evolvable hardware, self-organizing systems, emergent behaviors, machine perception.)
2) *Use of computers to simulate natural phenomena*; and
3) *Computing in nature (by natural materials)* (e.g., information processing in evolution by natural selection, in the brain, in the immune system, in the self-organized collective behavior of groups of animals such as ant colonies, and particle swarms, quantum computing, molecular computing, DNA computing, biocomputing, neural computation, evolutionary computation, biological computing/organic computing)

Computational paradigms studied by natural computing are abstracted from natural phenomena such as self-x attributes of living (organic) systems (including -replication, -repair, -definition and -assembly), the functioning of the brain, evolution, the immune systems, cell membranes, and morphogenesis. These computational paradigms can be implemented not only in the electronic hardware, but also in materials such as biomolecules (DNA, RNA), or quantum computing systems.[7]

According to natural computationalism, one can view the time development (dynamics) in nature as information processing, and learn about its computational characteristics. Such processes include self-assembly, developmental processes, gene regulation networks, gene assembly in unicellular organisms, protein–protein interaction networks, biological transport networks, and similar.

Natural computing has specific criteria for success of a computation. Unlike the case of Turing model, the halting problem is not a central issue, but instead the adequacy of the computational response. Organic computing system e. g. 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 not being algorithmic which makes it interesting to investigate computational characteristics of non-algorithmic natural computation (sub-symbolic, analog).

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. The research in theoretical foundations of Natural computing is needed to improve understanding on the fundamental level of computation as information processing which underlie all of computing in nature. Answering those focal research questions will increase understanding of the potential and the limits of the emerging computational

---

[7] For more details about new computational paradigms, and how "classical computability has expanded beyond its original scope to address issues related to computability and complexity in algebra, analysis, and physics", see (Cooper et al. 2008).

paradigm which will have significant impact on the research in both computing and natural sciences.

Much like the research in other disciplines of Computing such as AI, SE, and Robotics, Natural computing is interdisciplinary research, and has a synthetic approach, unifying knowledge from a variety of related fields. Research questions, theories, methods and approaches are used from Computer Science (such as Theory of automata and formal languages, Interactive computing), Information Science (e.g. Shannon's theory of communication), ICT studies, Mathematics (such as randomness, Algorithmic theory of information), Logic (e.g. pluralist logic, game logic), Epistemology (especially naturalized epistemologies), evolution and Cognitive Science (mechanisms of information processing in living organisms) in order to investigate foundational and conceptual issues of Natural computation and information processing in nature.

Significant for Natural computing is a bidirectional research (Rozenberg); as the natural sciences are rapidly absorbing ideas of information processing, computing concurrently assimilates ideas from natural sciences.

MacLennan (2004) defines natural computation as "computation occurring in nature or inspired by that in nature", (the "inspired by nature" includes simulation of natural phenomena we mentioned earlier) which might be represented by either discrete or continuous models. Natural computational models are most relevant in applications that resemble natural systems, as for example real-time control systems, autonomous robots, and distributed intelligent systems in general. There is an interesting synergy gain in the relating of human designed computing with the computing going on in nature.

If computation is to be able to simulate the observable natural phenomena, relevant characteristics in natural computation should be incorporated in new models of computation. Natural computational systems have the following important features (MacLennan 2004):

- Adequacy of real-time response—deliver usable results in prescribed real-time bounds. The speed of the basic operations is critical, as well as the absolute number of steps from input to output.
- Generality of response—with real-time response fixed, a natural computation may be improved by increasing the range of inputs to which it responds adequately.
- Flexibility in response to novelty—respond appropriately to novel inputs (which the system was not designed to handle).
- Adaptability—adapt to a changing environment, as quickly as possible, while retaining existing competence and stability. Natural computation systems can be compared with respect to the quality and speed of their adaptation and the stability of their learning.
- Robustness in the presence of perturbations, noise, faults, errors and damage, or even the ability to exploit perturbations and similar to the advantage of the system in developing new features.

That is why in natural computation, the same features are becoming important characteristics of computation. In this context it would be instructive to apply insights from Colburn and Shute (2008) in case of natural computing as a new metaphor of computing, with its different roles (pedagogical, design-oriented and scientific) and establishment of a new relationship between language and reality. In sum: A promising new approach to the complex world of modern autonomous, intelligent, adaptive, networked computing has successively emerged. Natural computing is a new paradigm of computing which deals with computability in the physical world, which has brought a fundamentally new understanding of computation.

Computation as Interaction and Concurrent Interactive Computing

Interactive computation (Wegner 1998) involves interaction, or communication, with the environment during computation, contrary to traditional algorithmic computation which goes on in an isolated system. The interactive paradigm includes concurrent and reactive computations, agent-oriented, distributed and component-based computations, (Goldin and Wegner 2002).

The paradigm shift from algorithms to interactive computation follows the technology shift from mainframes to networks, and intelligent systems, from calculating to communicating, distributed and often even mobile devices. A majority of the computers today are embedded in other systems and they are continuously communicating with each other and with the environment. The communicative role has definitely outweighed the original role of a computer as an isolated, fast calculating machine.

The following characteristics distinguish this new, interactive notion of computation (Goldin and Smolka 2006):

- Computational problem is defined as *performing a task*, rather than (algorithmically) producing an answer to a question.
- Dynamic input and output modeled by dynamic streams which are interleaved; later values of the input stream may depend on earlier values in the output stream and vice versa.
- The environment of the computation is a part of the model, playing an active role in the computation by dynamically supplying the computational system with the inputs, and consuming the output values from the system.
- Concurrency: the computing system (agent) computes in parallel with its environment, and with other agents that may be in it.
- Effective non-computability: the environment cannot be assumed to be static or effectively computable; for example, it may include humans, or other elements of the real world. We cannot always pre-compute input values or predict the effect of the system's output on the environment.

Even though practical implementations of interactive computing are several decades old, a foundational theory, and primarily the semantics and logic of interactive computing is only in its beginnings. A theoretical foundations analogous to what Turing machines are for algorithmic computing, is under development (Wegner 1998; Abramsky 2003).

Computational logic is a tool that both supports computation modeling and reasoning about computation. (Goldin and Wegner 2002) argue e.g. that computational logic must be able to model interactive computation, that classical logic does not suffice and that logic must be paraconsistent, able to model both a fact and its negation, due to the role of the environment and incompleteness of interaction.

If the semantics for the behavior of a concurrent system is defined by the functional relationship between inputs and outputs, as within the Church-Turing framework, then the concurrent system can be simulated by a Turing machine. The Turing machine is a special case of a more general computation concept.

The added expressiveness of a concurrent interactive computing may be seen as a consequence of the introduction of time within the perspective. Time seen from a system is defined through the occurrence of external events, i.e. through interaction with the environment. In a similar way, spatial distribution, (between an inside and an outside of the system, also between different systems) gets its full expression through interaction. Different distributed agents, with different behaviors, interact with different parts of the environment. In interactive computing, time distribution and generally also (time-dependent) spatial distribution are modeled in the same formalism (Milner 1989) and (Wegner 1998).

The advantages of concurrency theory in the toolbox of formal models used to simulate observable natural phenomena are according to (Schachter 1999) that:

> "it is possible to express much richer notions of time and space in the concurrent interactive framework than in a sequential one. In the case of time, for example, instead of a unique total order, we now have interplay between many partial orders of events–the local times of concurrent agents–with potential synchronizations, and the possibility to add global constraints on the set of possible scheduling. This requires a much more complex algebraic structure of representation if one wants to "situate" a given agent in time, i.e., relatively to the occurrence of events originated by herself or by other agents."

Theories of concurrency are partially integrating the observer into the model by permitting limited shifting of the inside-outside boundary. By this integration, theories of concurrency might bring major enhancements to the computational expressive toolbox.

> "An important quality of Petri's conception of concurrency, as compared with "linguistic" approaches such as process calculi, is that it seeks to explain fundamental concepts: causality, concurrency, process, etc. in a syntax-independent, "geometric" fashion. Another important point, which may originally have seemed merely eccentric, but now looks rather ahead of its time, is the extent to which Petri's thinking was explicitly influenced by physics (…).
>
> To a large extent, and by design, Net Theory can be seen as a kind of discrete physics: lines are time-like causal flows, cuts are space-like regions, process unfoldings of a marked net are like the solution trajectories of a differential equation. This acquires new significance today, when the consequences of the idea that "Information is Physical" [17] are being explored in the rapidly

> developing field of quantum informatics. Moreover, the need to recognize the spatial structure of distributed systems has become apparent, and is made explicit in formalisms such as the Ambient calculus [10], and Milner's bigraphs [23]." (Abramsky 2008).

It is important to point out that all mentioned non-standard computation models describe a new generation of computers: intelligent, adaptive, learning, with typical self-x properties as in organic computing and they definitely do not require infinite time or infinite resolution of measurement as they find inspiration in the information processing going on in organic systems, which perform their computational tasks in a finite time and with finite resources.

New Logical Approaches: Logical Pluralism

One can see the development of computer science in the light of historical experiences. Historically science was forced to leave absolutes, one by one. We were shifted from the absolute center of the Universe with a unique and privileged coordinate system, and placed in the outskirts of our galaxy which in no way is special among galaxies, only to later on be forced to leave the idea of absolute space altogether and what is even worse to give up absolute time. (Dodig-Crnkovic 2003) Now it is time to leave the absolute truth, which is connected to leaving the idea of one and only true logic and accept logical pluralism.

How does the change in logic relate to computing, computers and information? Those elements influence each other and the development within one field induces the development in the others, which in its turn, influences the original field, and so on.

There are several points of departure one can take in order to explore the alternatives of logical monism in the context of Philosophy of Information and Computation.

Focusing on information instead of knowledge can be a smooth way to logical pluralism (Beall and Restall 2000, 2005) motivated by an analysis of disagreement within the classical first-order logic, relevant logic and intuitionistic logic in the account of logical consequence (and hence of logical truth). Allo (2007) is arguing that logical pluralism could also entail semantic informational pluralism as informational content depends upon the underlying logic. Furthermore:

> "An elementary consequence of this point of view is that, when a formal account of semantic information is elaborated, the absolute validity of logic cannot be taken for granted. Some further—external—evidence for its applicability is needed."

Allo presents an interesting, case of communication between agents adhering to different logics in a multi-agent system. Information pluralism (as a consequence of logical pluralism) is not only interesting theoretical problem, but has relevant practical consequences. Understanding of contexts where it appears may help us computationally articulate fields outside the domain of traditional computing.

Information is characteristic of a dynamical system while knowledge presupposes static, steady states. Knowledge is not something you receive today and discard tomorrow. Information is.

The new interactive (communicative) role of computing is apparent in the Internet, the phenomenon that allows global communication and data transfer, making information easily available for people in different fields, establishing completely new preconditions for interdisciplinary learning, communication and collaboration. Related to the question of influence from other fields on computing, let us mention the work of Cantwell Smith (1996).

In his book On the Origin of Objects, Cantwell Smith gives an outline of the foundations for Philosophy of Computing, which may be understood as a philosophy of the phenomena that produce, transfer, or preserve information. The book ascertains that the old digital, mechanical computing paradigm is not enough; there is only a vague intuition of something new that will result from the opening up of computing (as defined by Hilbert's mathematical research agenda, i.e. algorithms) to the arts, humanities and other non-scientific practices. For the detailed discussion see (Dodig-Crnkovic 2010).

Some years later, the positive side of what is going on have become more visible—computing is bringing together sciences, humanities and arts, in a development parallel to that of the Renaissance, (Dodig-Crnkovic 2003), now with the computer in the place of the printing press:

> "All modern Sciences are strongly connected to Technology. This is very much the case for Biology, Chemistry and Physics, and even more the case for Computing. The engineering parts in Computing have connections both with the hardware (physical) aspects of the computer and software. The important difference is that the computer (the physical object that is directly related to the theory) is not a focus of investigation (not even in the sense of being the cause of a certain algorithm proceeding in a certain way) but it is rather theory materialized, a tool always capable of changing in order to accommodate even more powerful theoretical concepts."

New technological developments are exposing new sides of our relations with the world—with each other, as articulated in the arts and humanities, as well as with nature, as expressed in sciences. These changes have of course feedback mechanisms. Technology changing culture which in its turn changes technology.

Computers are as much theoretical (conceptual) devices as the material ones. The next step is to make computers capable of accommodating natural computation, as the most expressive way of computation able to simulate natural phenomena, including biological, cognitive and social ones. The traditional computing is not enough; computing is expanding its domains.

Closely related to the new expectations on computing, which is to be able to tackle open systems interacting with the environment, is the question of logic in such open interactive even heterogeneous systems, often such that they do not share the same logic. This actualizes the need for new logical approaches (Benthem van 2001, 2003), including logical pluralism. Pluralist logics are developing within the theory of computing (Allo 2007) and they will soon show as tools we need to reconceptualize the semantics of networks (or at least their computational theory). In terms of the new interaction paradigm computational processes are conceived as distributed, reactive, agent-based and concurrent. Agents, in general, may use different logics. Interaction provides a new conceptualization of computational phenomena which involve communication by information exchange, which makes way for logical pluralism.

> “Where is logic heading today? There is a general feeling that the discipline is broadening its scope and agenda beyond classical foundational issues, and maybe even a concern that, like Stephen Leacock’s famous horseman, it is ‘riding off madly in all directions’. So, what is the resultant vector? There seem to be two broad answers in circulation today. One is logical pluralism, locating the new scope of logic in charting a wide variety of reasoning styles, often marked by non-classical structural rules of inference. This is the new program that I subscribed to in my work on sub-structural logics around 1990, and it is a powerful movement today. But gradually, I have changed my mind about the crux of what logic should become. I would now say that the main issue is not variety of reasoning styles and notions of consequence, but the variety of informational tasks performed by intelligent interacting agents, of which inference is only one among many, involving observation, memory, questions and answers, dialogue, or general communication. And logical systems should deal with a wide variety of these, making information-carrying events first-class citizens in their set-up. This program of logical dynamics was proposed in van Benthem 1996. The purpose of this brief paper is to contrast and compare the two approaches, drawing freely on some insights from earlier published papers. In particular, I will argue that logical dynamics sets itself the more ambitious diagnostic goal of explaining why sub-structural phenomena occur, by ‘deconstructing’ them into classical logic plus an explicit account of the relevant informational events. I see this as a still more challenging departure from traditional logic. Diehard mathematicians still feel at ease with logical pluralism since it is all still a ‘science of formal systems’ describing ‘inference’, while to me, inference is just one way of producing information, at best on a par, even for logic itself, with others. But eventually—that is how my brain is wired—I move from confrontation to cooperation, suggesting ways in which the two views can pose new research questions for the other. In particular, inference and consequence relations pose challenges to logical dynamics, while dynamic logics in my style generate new consequence relations for pluralists to study.” (Benthem van 2008).

Logical Games and Interactive Computing

Games with their distributed, reactive, agent-based concurrency present a very suitable formalism for the modeling of interactive computing, i.e. of information flow and multi-agent interaction.

> "One difficulty in extending logic from the sentence level to a discourse level has been the scarcity of mathematical paradigms satisfying the standards that one has become used to at the sentence level. In recent years, a congenial mid-size level has been found in game theory. Games are typically a model for a group of agents trying to achieve certain goals through interaction. They involve two new notions compared with what we had before: agents' preferences among possible outcome states, and their longer-term strategies providing successive responses to the others' actions over time. In particular, strategies take us from the micro-level to a description of longer-term behaviour." (Benthem van 2003).

Up till now games in logic have been used to find models and semantic justifications for syntactically introduced intuitionist logic or linear logic. The recently initiated computability logic views games as foundational entities in their own right. The computability logic is motivated by the belief that syntax (axiomatization or other deductive constructions) should serve a meaningful and motivated semantics rather than vice versa. (Japaridze 2006).

> "The concept of games that computability logic is based on appears to be an adequate formal counterpart of our broadest intuition of interactive computational tasks,—tasks performed by a machine for a user/environment. What is a task for a machine is a resource for the environment and vice versa, so computability-logic games, at the same time, formalize our intuition of computational resources. Logical operators are understood as operations on such tasks/resources/games, atoms as variables ranging over tasks/resources/games, and validity of a logical formula as existence of a machine that always (under every particular interpretation of atoms and against any possible behavior by the environment) successfully accomplishes/provides/wins the task/resource/game represented by the formula. With this semantics, computability logic is a formal theory of computability in the same sense as classical logic is a formal theory of truth. Furthermore, the classical concept of truth turns out to be nothing but computability restricted to a special, so called elementary sort of games, which translates into classical logic's being nothing but a special elementary fragment of computability logic." (Japaridze 2006).

Hintikka (1982) has extended games to natural language semantics and to games with imperfect information.

According to Abramsky (1997) the key feature of games, compared to other models of computation, is that they make possible an explicit representation of the environment, and thus model interaction in an intrinsic way.

It is interesting to note that even Aristotle regarded logic as being closely related with the rules of dialogue, a form of information exchange via verbal interaction. The common medieval name for logic was dialectics. Charles Hamblin retrieved the link between dialogue and the sound reasoning, while Paul Lorenzen had connected dialogue to constructive foundations of logic in the mid twentieth century, (Hodges 2004). Hintikka (1973) raised the Dawkins question of purpose or intention for semantic games. His proposal was that one should look to Wittgenstein's language games, and the language games for understanding quantifiers are those about seeking and finding. In the corresponding logical games one should think of as Myself and as a Nature (which can never be relied on); so to be sure of finding the object I want, I need a winning strategy. Hodges (2004) criticizes this interpretation as not being very convincing with the claim that the motivation of Nature is irrelevant, but in the light of biological computing, Hintikka's suggestion seems to be an ingenious and fruitful metaphor.

Multi-agent interactions may be expressed in terms of two-person games. A one-person game is simply a transition system. Wegner's (1998) gives a detailed account of the interaction as a two-player game:

> "We consider the following questions: What kind of logic has a natural semantics in multi-player (rather than 2-player) games? How can we express branching quantifiers, and other partial-information constructs, with a properly compositional syntax and semantics? We develop a logic in answer to these questions, with a formal semantics based on multiple concurrent strategies, formalized as closure operators on Kahn-Plotkin concrete domains. Partial information constraints are represented as co-closure operators. We address the syntactic issues by treating syntactic constituents, including quantifiers, as arrows in a category, with arities and co-arities. This enables a fully compositional account of a wide range of features in a multi-agent, concurrent setting, including IF-style quantifiers." (Abramsky 2007).

In sum: new approaches in logic are developing which support modeling of interactive computation, which turn out to be useful in several ways for the advance of understanding of computational systems and processes.

Conclusions

> "In these times brimming with excitement, our task is nothing less than to discover a new, broader, notion of computation, and to understand the world around us in terms of information processing." (Rozenberg and Kari 2008).

This new, broader understanding of computation as information processing performed by a definable process is a generalization of Turing Machine model of algorithmic computing which is basically symbol manipulation according to a set of rules. The choice of the model for computation has wide consequences not only for computing and closely related fields of mathematics and logics but also for sciences and humanities. If we take the computing with a wide scope as natural computationalism suggests, computing becomes as (Denning 2007) already noticed—natural science. Used as exploratory instrument computing in this sense provides new foundation for natural philosophy—study of the world (including biological beings) within the common framework. The gain of this generalization is that it enables modeling of living systems as informational structures whose dynamics is information processing (computing). In this general case computation (in the world) is concurrent and asynchronous. The Turing model which presupposes closed system, sequential process and discrete variables (symbol system) is a special case of a more general concept of natural computation with open systems, parallel processes, both continuous and discrete variables, both symbolic and sub-symbolic, depending on level of organization (Cooper et al. 2008). From the perspective of natural computationalism (Zuse, Fredkin, Wolfram, Chaitin, Lloyd), the physical universe is a computer,[8] a network of distributed communicating computing processes going on in an informational structure.

This essay presents several ideas that combined result in a new view of natural, interactive computing as information processing, with broad consequences relevant for not only computer science and closely related fields of physics, mathematics and logic but even for traditionally non-mechanizable fields of biology, cognitive sciences and neuroscience. We have still much to learn from nature and especially from naturally intelligent information processing systems such as humans and animals which will bring about new models of computation and intelligence:

> “We can only see a short distance ahead, but we can see plenty there that needs to be done.” (Turing 1950).

Acknowledgments The author would like to thank Björn Lisper, Lars-Göran Johansson and Kaj Börje Hansen for reviewing the manuscript and offering valuable suggestions. Further credit is extended to Richard Bonner and George Masterton for their interesting comments and discussions. I am most indebted to Vincent Müller with whom I newly wrote a dialogue article based on several years of discussions on topics of foundation of information and computation. Last but not list I would like to acknowledge the constructive criticisms and helpful suggestions of two anonymous reviewers on an earlier version of this paper.

References

Abramsky, S. (1997). Semantics of interaction: An introduction to game semantics. In P. Dybjer & A. Pitts (Eds.), *Proceedings of the 1996 CLiCS Summer School, Isaac Newton Institute* (pp. 1–31). Cambridge: Cambridge University Press.

Abramsky, S. (2003). Sequentiality versus concurrency in games and logic. *Mathematical Structures in Computer Science, 13*, 531–565.

Abramsky, S. (2007). A compositional game semantics for multi-agent logics of imperfect information in interactive logic. In J. van Benthem, D. Gabbay, & B. Lowe (Eds.), *Texts in logic and games, Vol. 1* (pp. 11–48). Amsterdam: Amsterdam University Press.

Abramsky, S. (2008). Petri nets, discrete physics, and distributed quantum computation. In P. Degano, R. De Nicola and J. Meseguer (Eds.), Concurrency, graphs and models, essays dedicated to Ugo Montanari on the occasion of his 65th birthday, Vol. 5065 of lecture notes in computer science. Springer, 527–543.

Abramsky, S., & Coecke, B. (2007). Physics from computer science, Int. *Journal of Unconventional Computing, 3*(3), 179–197.

Allo, P. (2007). Formalising semantic information. Lessons from logical pluralism. In G. Dodig-Crnkovic & S. Stuart (Eds.), *Computation, information, cognition: The Nexus and the Liminal* (pp. 41–52). Cambridge: Cambridge Scholars Publishing.

Ashby, W. R. (1964). *An introduction to cybernetics*. London: Methuen.

Beall, J. C., & Restall, G. (2000). Logical pluralism. *Australasian Journal of Philosophy, 78*, 475–493.

---

[8] Some authors identify the idea of computing universe with discrete computing, which is not necessarily the only possible interpretation. Lloyd for example argues that on the quantum mechanical level both continuum and discrete structures are necessary. Discrete models are abundant and very useful, but there are also models in physics which presuppose continuous descriptions.

Beall, J.C., Restall, G. (2005). *Logical Consequence*, The Stanford Encyclopedia of Philosophy (Winter 2005 Edition). In: Edward N. Zalta (ed.). URL = <http://plato.stanford.edu/archives/win2005/entries/logical-consequence/>.

Benthem van, J. (2001). Extensive games as process models. In M. Pauly & P. Dekker, (Eds.), *Special issue of Journal of logic, language and information*, 11, 289–313.

Benthem van, J. (2003). Logic and the dynamics of information. *Minds and Machines, 13*, 503–519.

Benthem van, J. (2008). Logical pluralism meets logical dynamics? *The Australasian Journal of Logic, 6*, 182–209.

Bohan Broderick, P. (2004). On communication and computation. *Minds and Machines, 14*, 1–19.

Bringsjord, S. (1994). Computation, among other things, is beneath us. *Minds and Machines, 4*, 469–488.

Bringsjord, S., & Zenzen, M. (2002). Toward a formal philosophy of hypercomputation. *Minds and Machines, 12*, 241–258.

Burgin, M. (1999). Super-recursive algorithms as a tool for high performance computing. In Proceedings of high performance computing symposium, San Diego, 224–228. http://www.math.ucla.edu/∼mburgin/res/compsc/res2/highperfcomp.html

Burgin, M. (2005). *Super-recursive algorithms*. Springer Monographs in Computer Science.

Cantwell Smith, B. (1996). *On the origin of objects*. Cambridge, MA: MIT Press.

Chaitin, G. J. (2006). Epistemology as information theory: From Leibniz to Ω. *Collapse, 1*, 27–51.

Church, A. (1935). Abstract no. 204. *Bulletin of the American Mathematical Society, 41*, 332–333.

Church, A. (1936). An unsolvable problem of elementary number theory. *American Journal of Mathematics, 58*, 345–363.

Colburn, T. R., & Shute, G. M. (2008). Metaphor in computer science. *Journal of Applied Logic, 6*, 526–533.

Cooper, S. B., Löwe, B., & Sorbi, A., Eds. (2008). New computational paradigms: Changing conceptions of what is computable. Springer.

Copeland, B. J. (1997). 'The Church–Turing thesis'. In E. Zalta (Ed.), Stanford encyclopedia of philosophy, <http://plato.stanford.edu>.

Copeland, B. J. (1998). Super Turing-machines. *Complexity, 4*, 30–32.

Copeland, B. J. (2000). Narrow versus wide mechanism. *Journal of Philosophy, 96*, 5–32.

Copeland, B. J. (2002). Hypercomputation. *Minds and Machines, 12*, 461–502.

Copeland, B. J., & Proudfoot, D. (1999). Alan Turing's forgotten ideas in computer science. *Scientific American, 280*, 76–81.

Copeland, B. J., & Shagrir, O. (2007). Physical computation: How general are Gandy's principles for mechanisms? *Minds and Machines, 17*, 217–223.

Copeland, B. J., & Sylvan, R. (1999). Beyond the universal Turing machine. *Australian Journal of Philosopy, 77*, 46–67.

Denning, P. (2007). Computing is a natural science, communications of the ACM, 50(7), 13–18. http://cs.gmu.edu/cne/pjd/PUBS/CACMcols/cacmJul07.pdf.

Dodig-Crnkovic, G. (2003). 'Shifting the paradigm of the philosophy of science: The philosophy of information and a new renaissance'. Minds and Machines, 13(4), 521–536. http://www.springerlink.com/content/g14t483510156726/fulltext.pdf.

Dodig-Crnkovic, G. (2006). *Investigations into information semantics and ethics of computing*. Mälardalen University Press http://www.idt.mdh.se/personal/gdc/work/publications.html.

Dodig-Crnkovic, G. (2010). The cybersemiotics and info-computationalist research programmes as platforms for knowledge production in organisms and machines, entropy 12, 878–901. http://www.mdpi.com/1099-4300/12/4/878/pdf

Dodig-Crnkovic, G., & Müller, V. (2010). A dialogue concerning two world systems: Info-computational versus mechanistic. In Information and computation. World Scientific, Singapore. Preprint available at: http://arxiv.org/abs/0910.5001

Dodig-Crnkovic, G., & Stuart, S. (Eds.). (2007). *Computation, information, cognition–The Nexus and the Liminal*. Cambridge: Cambridge Scholar Press.

Floridi, L. (2004). Open problems in the philosophy of information. *Metaphilosophy*, 35.4, 554–582.

Goldin, D., Smolka, S., & Wegner P. (Eds.). (2006). *Interactive computation: The new paradigm*. Springer-Verlag.

Goldin, D., & Wegner, P. (2002). *Paraconsistency of interactive computation*, PCL 2002 (Workshop on paraconsistent computational logic). Denmark.

Hintikka, J. (1973). *Logic, language-games and information: Kantian themes in the philosophy of logic*. Oxford: Clarendon Press.

Hintikka, J. (1982). Game-theoretical semantics: Insights and prospects. *Notre Dame Journal of Formal Logic, 23*(2), 219–241.

Hodges, W. (2004). *Logic and Games*, The Stanford encyclopedia of philosophy (Winter 2004 Edition). Edward N. Zalta (Ed.). URL = <http://plato.stanford.edu/archives/win2004/entries/logic-games/>.

Hodges, A. (2009). *Alan Turing*, The Stanford encyclopedia of philosophy (Winter 2009 Edition). In Edward N. Zalta (Ed.). URL = <http://plato.stanford.edu/archives/win2009/entries/turing/>.

Japaridze, G. (2006). In the beginning was game semantics. In O. Majer, A.-V. Pietarinen, & T. Tulenheimo (Eds.), *Logic and games: Foundational perspectives*. Berlin: Springer Verlag.

Kampis, G. (1991). *Self-modifying systems in biology and cognitive science: A new framework for dynamics, information and complexity*. Pergamon: Pergamon Press.

Kelly, K. T. (2004). Uncomputability: The problem of induction internalized. *Theoretical Computer Science, 317*, 227–249.

Kuipers, T. A. F. (2006). Theories looking for domains. Fact or fiction? Structuralist truth approximation by revision of the domain of intended applications, to appear. In L. Magnani (ed.), *Model-based reasoning in science and engineering*.

Lloyd, S. (2006). Programming the universe: A quantum computer scientist takes on the cosmos. In Alfred A. Knopf.

MacLennan, B. (2004). Natural computation and non-Turing models of computation. *Theoretical Computer Science, 317*, 115–145.

Milner, R. (1989). *Communication and concurrency*. Prentice-Hall: International Series in Computing Science.

Piccinini, G. (2007). Computational modeling versus computational explanation: Is everything a Turing machine, and does it matter to the philosophy of mind? *Australasian Journal of Philosophy, 85*(1), 93–115.

Priest, G., & Tanaka, K. (2004). *Paraconsistent Logic*, The Stanford encyclopedia of philosophy (Winter 2004 Edition). In Edward N. Zalta (Ed.). URL = <http://plato.stanford.edu/archives/win2004/entries/logic-paraconsistent/>.

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

Schachter, V. (1999). How does concurrency extend the paradigm of computation? *Monist, 82*(1), 37–58.

Sieg, W. (2007). Church without dogma–Axioms for computability. In B. Löwe, A. Sorbi, & S. B. Cooper (Eds.), *New computational paradigms: Changing conceptions of what is computable* (pp. 18–44). Heidelberg: Springer.

Siegelman, H. T. (1999). *Neural networks and analog computation*. Berlin: Birkhauser.

Sloman, A. (1996). Beyond Turing equivalence. In P.J.R. Millican & A. Clark (Eds.), Machines and thought: The legacy of Alan Turing, vol I, OUP(The Clarendon Press) pp 179–219, Revised version of paper presented to Turing Colloquium, University of Sussex, 1990. http://www.cs.bham.ac.uk/research/projects/cogaff/96-99.html#1

Turing A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. In Proceedings of the London mathematical society, Vol. 42, pp. 230–265; reprinted in A. M. Turing, Collected works: mathematical logic, 18–53.

Turing A. M. (1939). Systems of logic based on ordinals.In Proceedings of the London mathematical society, ser. 2, Vol. 45, 162–228.

Turing, A. M. (1948). 'Intelligent machinery'. National Physical Laboratory Report. In B. Meltzer, D. Michie (Eds.), 1969. Machine intelligence 5. Edinburgh: Edinburgh University Press. http://www.AlanTuring.net/intelligent_machinery.

Turing, A. M. (1950). Computing machinery and intelligence, Mind LIX, 433–60. http://cogprints.org/499/0/turing.html

Wegner, P. (1998). Interactive foundations of computing. *Theoretical Computer Science, 192*, 315–351.

Wolfram, S. (2002) *A new kind of science*. Wolfram Science.