UNIVERSITY OF EXETER

The 6th AISB Symposium on Computing and Philosophy: The Scandal of Computation - What is Computation?

Mark Bishop and Yasemin J. Erden (editors)

AISB Convention 2013 • University of Exeter • 3rd–5th April, 2013

Foreword from the Convention Chairs

This volume forms the proceedings of one of eight co-located symposia held at the AISB Convention 2013 that took place 3rd-5th April 2013 at the University of Exeter, UK. The convention consisted of these symposia together in four parallel tracks with five plenary talks; all papers other than the plenaries were given as talks within the symposia. This symposium-based format, which has been the standard for AISB conventions for many years, encourages collaboration and discussion among a wide variety of disciplines. Although each symposium is self contained, the convention as a whole represents a diverse array of topics from philosophy, psychology, computer science and cognitive science under the common umbrella of artificial intelligence and the simulation of behaviour.

We would like to thank the symposium organisers and their programme committees for their hard work in publicising their symposium, attracting and reviewing submissions and compiling this volume. Without these interesting, high quality symposia the convention would not be possible.

Dr Ed Keedwell & Prof. Richard Everson
AISB 2013 Convention Chairs

Published by
The Society for the Study of Artificial Intelligence and the Simulation of Behaviour
http://www.aisb.org.uk

ISBN: 978-1-908187-31-4

The 6th AISB Symposium on Computing and Philosophy: *The Scandal of Computation - What is Computation?*

What is computation? Society builds and uses millions of computers each year so at first sight the answer seems trivial. A computer is merely a general purpose, typically electronic device, that can be programmed to carry out a finite set of arithmetic or logical operations. These days they announce their ubiquity to the world in phones, desktop devices, washing machines, even lawn mowers.

Historically, however, the etymology of the word (from the OED) informs us that the notion of computation was identified with the action of humans who make calculations, often with the aid of calculating machines. In the 1940s this definition was refined with that of an “effective method” (a procedure that reduces the solution of problems to a series of rote steps which is bound to give the correct answer in finite time for all possible inputs), to yield the notion of the algorithm an effective method for calculating the values of a function and the notion of the effective calculability of functions with an effective method (algorithmic solution). In this way, the notion of computation came to be identified with the actions [steps] carried out by [automated] computers to produce definite outputs [in finite time]. This notion frames computation in terms of an agent, which raises the questions of what computation is per se - merely the dynamics of information flow? And in this scenario, how can computational data be meaningful? How can meaningful data acquire truth-values?

For a long time our ideas about computations (or about the underlying computational models) were more or less rigid, fixed, established in the middle of the twentieth century. In the centre there was the model of a classical Turing machine, with its scenario of a finite computation defining a fixed mapping from the inputs to the outputs. The computations of Turing machines served as a means for defining the complexity of computations, the notion of the universality of computations, and the notion of computability (historically, the lastly mentioned three notions should have been listed in a reversed order). Nevertheless, with the advent of modern computing technologies, networking, and advances in physics and biology, has emerged the ideas that computation is a far broader, far more common, and more complex phenomenon than that modelled by Turing machines. It has been increasingly more difficult to see newly emerging models of computations through the optics of Turing machine computations. Examples include biologically inspired models—such as neural nets, DNA computing, self-assembled structures, molecular computers, cognitive computing, brain computing, swarm computing, etc., or physically inspired models, such as quantum computing, relativistic computers, hyper-computers, and, last but not least, “technologically enabled” models, with the prominent example of the Internet, but also various (also mobile) networks.

In order to further explore these and related questions, the papers in this Symposium cover key related issues including, but not limited to:

Computationalism and Neural Systems; Models of Computation; Natural Computing; Computation as Knowledge Generating Processes; Computational Complexity Theory; Quantum Computing; Trivialization Arguments; Natural Computation; Dynamical Systems Theory; Computation and Pragmatics; Dynamics of Information; Interactive Computation; Intentional and Functional Concepts; Hypercomputers; Pancomputationalism; Digital Systems; Type and Token; Computational Universe; Special Relativity; Turing Machines; Stochastic Diffusion Search; Monte-Carlo Tree Search; Computational Platform; Game-Playing; Observer-Relativity; Phenomena and Noumena.

On behalf of the Organising Committee of this Sixth AISB Computing and Philosophy Symposium, we would like to thank all the members of the Programme Committee for their generous support, and for the excellent work in refereeing submissions. We hope that participants will find the event stimulating and enjoyable.

Mark Bishop (Dept. of Computing, Goldsmiths, University of London, UK)
Symposium Chair and AISB Chair
Yasemin J. Erden (Philosophy, St Mary's University College, UK)
Symposium Co-Chair and AISB Committee Member

Symposium Organising Committee:
Slawomir Nasuto (University of Reading, UK)
Stephen Rainey (St Mary's University College, UK)
Jiri Wiedermann, (Academy of Sciences of the Czech Republic, CZ)

Symposium Programme Committee:
Ron Chrisley (University of Sussex, UK)
S. Barry Cooper (University of Leeds, UK)
José Félix Costa (IST Technical University of Lisbon, PT)
George F. R. Ellis (University of Cape Town, SA)
Peter beim Graben (Humboldt-Universität zu Berlin, DE)
Yuri Gurevich (Microsoft Research, USA)
Phyllis Illari (University College London, UK)
Robert W. Kentridge (Durham University, UK)
Jan van Leeuwen (Universiteit Utrecht, NL)
Matthias Scheutz (Tufts University, USA)
Oron Shagrir (The Hebrew University of Jerusalem, IL)
Mariarosaria Taddeo (University of Hertfordshire, UK)
Mario Villalobos (The University of Edinburgh, UK)

Contents

What is a Digital State?

Vincent C. Müller[1]

Abstract. There is much discussion about whether the human mind is a computer, whether the human brain could be emulated on a computer, and whether at all physical entities are computers (pancomputationalism). These discussions, and others, require criteria for what is digital.

I propose that a state is digital if and only if it is a token of a type that serves a particular function – typically a representational function for the system. This proposal is made on a syntactic level, assuming three levels of description (physical, syntactic, semantic). It suggests that being digital is a matter of discovery or rather a matter of how we wish to describe the world, if a functional description can be assumed. Given the criterion provided and the necessary empirical research, we should be in a position to decide on a given system (e.g. the human brain) whether it is a digital system and can thus be reproduced in a different digital system (since digital systems allow multiple realization).

1. MOTIVATIONS: THE COMPUTATION–ALIST PROGRAM, ARTIFICIAL INTELLIGENCE, PANCOMPUTATIONALISM

Given that the ontology of digital states is hardly an established philosophical problem, it will be useful to briefly motivate its discussion. A clarification as to what constitutes a digital state is necessary primarily in the context where digital states are part of a certain kind of digital systems, namely digital computers. There is a significant confusion over which objects in the world are computers because there is no agreement on the criteria; Shagrir calls this the “problem of physical computation” (1, 394ff). For example, on the one hand there are the proponents of a computational representational theory of mind (CRM or “computationalism”) who believe that the human mind is a functional computational mechanism operating over representations. These representational abilities are then to be explained naturalistically, either as a result of information-theoretical processes (2, 3), or as the result of biological function in a “teleosemantics” (4, 5).

On the other hand, the opponents of computationalism divide into two camps: those who think that some natural mechanisms may be computers, but the human mind is not one of these, and those who think that all systems can be *interpreted* as computers and so the human mind is just a computer like everything else: “every natural process is computation in a computing universe” (6, 10) – this position is now often called “pancomputationalism” (see 7). Finally, the question to what extent artificial intelligence (AI) is possible, requires an explanation what kinds of machines computers are and what they can do in principle – given that digital computers are currently the main kind of mechanism that is used for AI.

So, if the brain is a computer, could we perhaps reproduce the brain on different hardware? If we would scan the whole brain of a human and run it on a (different) Turing machine, *would it produce intelligence?* (And all the other cognitive features of humans?) If we could emulate the brain on different hardware it would show the same external behavior or output as does the emulation of earlier software or hardware, e.g. of a WWII ‘Enigma’ machine or of programs that ran on some of the first computers, like the ‘Manchester Mark I’.

This might be a possibility if certain conditions are met, in particular: “Computability: brain activity is Turing-computable” (8) and “At present there is no convincing empirical evidence for uncomputability in the brain, although there is no shortage of claims for it.” (9).

Several further problems for a computational theory of the mind would benefit from a resolution of what constitutes a digital state. Within the context of the discussion of the computationalism mental processes are traditionally understood as information processing through computational operations over *representations*. Is representation a necessary feature of computing? If yes, perhaps something can be called a digital state only on *presupposing* mental processes in the system, so there is a threat of a circle here (unless we are looking at a feedback circle). Another is the problem of “grounding” for computational systems: “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?” (10, 335). We have argued in recent papers (11, 12) that a nonconceptual phenomenal content should be at the base of such grounding. If it were to turn out that such content is necessary but is analogue and cannot be present in purely digital systems, this would show that human cognition is not purely digital – and that AI on purely digital computers is impossible. In separate work, we argue that nonconceptual content is precisely non-digital content.

I will argue in the following that being a digital state is to be a state of a type or category, but that we should not conclude from this that being a digital state is “description-dependent”. In particular, a state can be digital if it fulfills a particular function in a system of which it is a part – e.g. a representational function.

2. BASIC CHARACTERISTICS

In a first approximation, being digital means being in a discrete state, a state that is strictly separated from another, not on a continuum. Prime examples of digital states are the states of a digital speedometer or watch (with numbers as opposed to an analog hand moving over a dial), the digital (binary) states in a conventional computer, the states of a warning light, or the states in a game of chess. Some digital states are binary, they have only

---

[1] Anatolia College/ACT, Pylaia & FHI, Dept. of Philosophy, University of Oxford. vmueller@act.edu, www.sophia.de

two possible states, but some have many more discrete states, such as the 10 numbers of a digital counter or the 26 letters of the standard English alphabet.

2.1. Multiple Realization

Goodman had pointed out in his early theory of representation that digital "marks" (physical entities) are "differentiated", as he called it, precisely if they can have an exact replica: one can write the same letter "A" twice, since "A" is differentiated from any other letter. Analog marks, in contrast, are "dense", meaning that for any pair of similar but non-identical marks, there is space for another mark in between (13, cf. 14). So, the states of an analog speedometer with a hand moving in analogy to the speed of the vehicle are continuous, just as the speed it represents, and for any two places where the hand can be, there is a third in between.

As we already pointed out with reference to Goodman, it is characteristic of a digital mark that it can be realized several times. So, one can write the same word twice, even if one cannot make exactly the same mark on paper twice. John Haugeland usefully explains this phenomenon with games: chess is a digital game because we can reproduce an earlier position precisely; we can even resume the same game with different pieces. Billiards, on the other hand, is analogue, because we can reproduce an earlier position only to a certain degree of measured precision, and if we were to reproduce the same position with different physical objects, it would not be the same position (15, 57, earlier in 16). The possibility of multiple realization is a *result* of digital states being discrete: Since a white bishop in chess can be clearly on field C3, we can move it back to C3, or replace it with a different bishop; it does not matter that it is not identical to the earlier one, provided it is clearly a white bishop on C3.

2.2. Discrete vs. Continuous

But which of the two characteristics is crucial for an analog state, the analogy to the represented, or the continuous movement?

This question becomes relevant in the case of analogous representations that proceed in steps, e.g. a clock the hands of which jump from one discrete state to another. Zenon Pylyshyn argues that the underlying process is analog, and this is what matters: "an analog watch does not cease to be analog even if its hands move in discrete steps" (17, 200, 18, 332 agrees). James Blachowicz also thinks that being on a continuum is sufficient for being analog, taking the view that "differentiated representations may also be analog – as long as they remain *serial*", his example is a slide rule with "clicks" for positions (19, 71). (Note how these authors assume a functional description, an issue to which we shall return later.)

These views ultimately fail to differentiate between analogue and digital representations. Note that the very same underlying mechanism could give a signal to a hand to move one step and to a digit to go one up (this is actually how clocks are controlled in centralized systems, e. g. at railway stations). In any case, some classic examples of digital states are clearly in a series, indeed a series of infinitely many steps: the series of the natural numbers. These two points rule out Blachowicz' proposal to take being serial as a criterion. Pylyshyn, on the other hand, would presumably say that the underlying mechanism is already digital, so the clock is digital in this case – but surely there are systems where a digital signal is converted into an analogue one (the speedometer in most modern cars) and where an analogue signal is converted into a digital one (an analogue central clock that controls several digital clocks), so we should then say that the system has digital and analog parts. I conclude that the first crucial feature of a digital state is indeed that of being a discrete state – not excluding that of being in a series, even in a series that is analogue to what is represented.

3. EVERYTHING IS ANALOGUE – AND DIGITAL, TOO?

A given blob of ink on a piece of paper might be in a particular digital state but it has several analogue properties, too, such as a color, a shape, a history, a value, etc. In fact, all digital states we have seen so far are states of physical entities, and thus have analogue properties as well. (For our purposes, we can leave aside the question whether abstract objects can be digital.) Being digital is a property of certain physical entities that are also analogue – though they might not be analogue representations. But of which entities? Negroponte puts it nicely: "A bit has no color, size or weight, … It is a state of being: on or off, true or false, up or down, in or out, black or white." (20, 14) But, which of the black things are in the state of being of a bit? What determines whether something is a bit?

It may seem that *we* can just define what counts as digital as we please, so everything is digital. Say, for example, the two of us agree that if I light a fire on a particular hill that means "the King is out of town". Is the hill henceforth in a binary digital state? Is anything not in any number of digital states, then?

For a given physical thing (say, my desk lamp), there are descriptions as continuous (where is the light, what is its shape, what its color?) and as digital (is it on/off?), so a natural response is to say that being a digital state is relative to a particular description: Under one description the light is digital, under another it is not, so we have at least a "relativity of descriptions" (21, 29).

This consequence is very tempting for digital computation and its algorithmic procedures. Alan Turing already seems to have already gone in this direction: "The digital computers […] may be classified amongst the 'discrete state machines', these are the machines which move by sudden jumps or clicks from one quite definite state to another. […] Strictly speaking there are no such machines. Everything really moves continuously. But there are many kinds of machine, which can profitably be thought of as being discrete state machines." (22, 439).

John Searle takes it one step further: "The electrical state transitions are intrinsic to the machine, but the computation is in the eye of the beholder." (23, 64). Oron Shagrir concurs: "… to be a computer is not a matter of fact or discovery, but a matter of perspective" (1, 393), and about algorithms: "… whether a process is algorithmic depends on the way we describe the process." "… processes are not really step-satisfaction [algorithmic]. It is simply useful to describe them this way." "Whether a system is digital depends not only on its natural properties, but chiefly on the context in which it is described." (18, 321, 331, 335).

It now seems that not only do we have a relativity of descriptions, but that a *description dependence of facts*: it would then be constitutive of being a digital state that its existence is dependent on contingent social interests, namely the interest in a particular feature that makes a digital state. To illustrate this with a classical example: Being ‘digital’ is more like the word ‘constellation’ than the word ‘star’. What is part of a stellar constellation depends on what *we* make part of it. What is a star depends on the world and is a matter of astronomic discovery (cf. 21, 18, 28, 24)

4. CLARIFICATION I: TYPE/TOKEN

Understanding the true nature of relativity here requires some further clarifications. Haugeland defines as follows: “A *digital system* is a set of positive and reliable techniques (methods, devices) for producing and reidentifying tokens, or configurations of tokens, from some prespecified set of types … A *positive* technique is one that can succeed absolutely, totally, and without qualification; … Many techniques are positive and reliable. Shooting a basketball at the basket is a positive method (for it getting through), for it can succeed absolutely and without qualification;” (15, 53f). Demopoulos thus calls being a digital mechanism of a certain type being a member of an “equivalence class” (25). Harnad talks about “symbol tokens” – but not of types (10, 1.2). The characteristic of “multiple realization” (see above 2.1) is crucial, so there must be a “positive technique” to produce perfect realizations that are clearly of *this* digital state.
Multiple realization, however, this is not a feature of *certain* types, it is a feature of types, quite generally. For example, a transistor can be in a voltage state that is clearly of type “on” or “off”, but it can also be on the borderline between the two – it just so happens that our computing machines are made with systems that do not usually get stuck in intermediate states. Every digital state is also on a continuum: the digital speedometer might change quickly from one number to another, but it does have intermediate states – just that these are not states of numbers, not states of these types, of these descriptions. Being of a digital state is thus not a statistical question of whether “all or none” states occur often, since what counts as “all” depends on the type. Just looking at the physical distribution will not tell us anything.

What is crucial here, therefore, is that a digital state is *of a type*. If it is of a type, then there can be multiple perfect realizations of it: No matter how many borderline cases a type happens to have (some have many, some have none), there is always the possibility of *clear* cases, and that is what is needed for being a digital state; we need to fulfill the implied semantic normativity of the “token of a type”. A digital type can be vague, it just needs possible clear cases. So, we require in a first instance that *a digital state is a state that is a token of a type*. What we need to see now is which tokens of a type are the digital states.

5. CLARIFICATION II: LEVELS OF DESCRIPTION

In a next step, it is helpful to differentiate at least three levels of description of a proposed candidate for being in digital or digital computational states: (a) *physical*, (b) *syntactic* and (c) *semantic* levels – something only very few people do, despite the tradition of functionalism (26, 57, 27, 402f, cf. 28) (29).

The *physical level* (a) is that of the physical ‘realization’ of the computation – this is presumably what Searle had in mind with his “electrical state transitions” (above).

That physical state is in (b) a particular digital state on the *syntactic level* (a binary state, or a number, a letter, a word). It is at this level that a particular mathematical function is computed, it is fully specified by specifying it on this level.

(c) That digital state in turn may represent something else, e.g. a truth value, a time, or a color, let us call this the *semantic level*. What is represented at this level may, again, have representational functions on several levels (the color can represent a political opinion, etc.).

A digital computer works because it is constructed in such a fashion that its physical states cause other physical states in a systematic way, and these physical states are also digital states on the syntactic level. (The physical states need not be of the same physical type, they can be voltages and magnetic fields, for example.) The semantic level is not necessarily present and is not necessary for the digital system or digital mechanism. Contrary to popular belief, a computer does not require semantic content to function, (e.g. 30, 31, 1414ff) and (15, 66, 32, 385).

Given this clarification of levels, we can re-evaluate the understanding of digital states. The *semantic* level allows for a true relativity of facts, not just of descriptions: The same computer following the same algorithm can be said to compute different things. This is hardly surprising. For example, it may well be that what a computer does with the same binary sequence is to add two numbers or to change one letter to another. Whether we want to regard the binary sequence as the one or the other will depend on the context. So, these syntactic binary states can have many different contents, on a semantic level (just like “2 + 2 = 4” can add apples or pears). The semantic level, however, is irrelevant to the specification of the digital states, which reside on the syntactic level. – Contrary to popular belief, relativity on the semantic level does not show that there is a relativity of facts on the *syntactic* level. (Note that I am not thereby claiming that “digital state” is a natural kind, i.e. roughly a kind where belonging to the extension is determined wholly by criteria that are themselves natural kinds [recursively], that is, their existence is independent of any conceptual system that has a label for them.)

Remember that the digital states in a system often represent other digital states (e.g. the binary states represent numbers), which relates to the discussion over whether our mental computation is computation over material symbols, as is writing down a mathematical proof. (Do I think in words?)

6. CLARIFICATION III: A DIGITAL STATE IS A TOKEN OF A FUNCTIONAL TYPE

It is useful to note that not all systems that have digital states are digital *systems*. We can, for example, consider the male and female humans entering and leaving a building as digital states, even as a binary input and output, but in a typical building these humans do not constitute a digital system because a relevant causal interaction is missing. In the typical digital system, there will thus be a *digital mechanism*, i.e. a causal system with a *purpose*, with parts that have *functions*. Digital mechanisms in this sense may be artifacts (computing machines) or natural objects (perhaps the human nervous system). However, it seems clear that not all digital states are parts of computational systems: the words in this paper are digital states, but their function is not computational.

If being of a type was the criterion for being digital, then everything would be in any number of digital states, depending on how it is described. However, what we really should say is that something is digital because that is its particular *function*. My desk lamp is always in a digital state, because being on/off is part of its function. The first letter of this sentence is in the digital state of being a "T" because that is its function – it is not an accidental orientation of ink or black pixels. The sun, on the other hand, is not in a digital state at present, though it can be shining or not shining at some place.

We make artifacts where some physical states cause other physical states such that these are physical states of the same set of types, e.g. binary states. (Note that one machine might produce binary states in several different physical ways, e.g. as voltage levels and as magnetic fields.) If someone would fail to recognize that my laptop computer has binary digital states, they would have failed to recognize the proper (non-accidental) function of these states for the purpose of the whole system – namely what it was made for. The fact that a logic gate in my laptop is a binary state depends on whether it *has* that function and is not description dependent. (And the fact that it computes is crucial to its function, but not to that of, say, my shaving brush – so pancomputationalism seems misleading here.)

I conclude that we should say a state is digital if and only if it is a token of a type that serves a particular function.

So, the function is that determines whether something is a token of a type or not. The normativity of having or fulfilling a function generates the normativity of being of a type. The type has the function, being of the type allows to fulfill the function. (Even though *we* would prefer to design digital *systems* such that they have very few borderline cases and almost only clear cases, this is not a criterion for being a digital system.)

6.1. Which Function?

At this point, it is clear that the description dependence of being digital depends on that of having a function. Functions are a very large issue, let me just indicate why one might think that there may be some facts here that are not description dependent.

In the case of an artifact, we *assume* a functional description. If the oil-warning light on a car dashboard is off, is it in a digital state? Yes, if its function is to indicate that nothing is wrong with the oil level. (It may serve all sorts of other accidental functions for certain people, of course.) But if the light has no electricity (the ignition is off), or if it was put there as a decorative item, then the lamp is not in a digital state "off". It would still be off, but this state would not be digital, would not be a token of the same functional kind.

In the case of a natural object, the allocation of proper function is dependent on teleological and normative description of systems (33, esp. 2.2) – a problematic but commonplace notion. The function of a human's legs seems to be locomotion (and kicking balls), but we are not tempted to say that the leg is in digital states, while perhaps the muscle cells are – with respect to *their* function. Whether or not the legs are digital, they can be simulated (to an arbitrary degree of precision) on digital systems – only that the simulation will not walk, it will just "walk in the simulation".

The description of an artifact in terms of function is to say that something is a means to an end, it serves the function to achieve that end – a function that can be served more or less well. (Note that serving a function does not mean being used for that function; there may well be no agent that can properly be said to be using the artifact, e.g. if it is part of a large and complex system.)

The dependence of being a digital type on having a proper function can be illustrated by looking at a digital information channel. Even if that channel is already defined as digital, it is not clear which parts are relevant and carry a function. For example, in computer security, there is the question which aspects of the information are used to convey information, searching for possible "covert channels", for example in the time delays between signals (see 34). These time delays can be used to convey information, similar to the delays in Morse code. So, given a time sequence of digital signals, it is still not clear which digital signals are present, unless function is specified, e.g. by stating whether particular time delays are significant or not.

6.2. Too many Functions, too Many Types

At this point, we need to see whether the account so far is sufficient to identify the digital states. A little reflection will reveal that it captures the standard samples of digital states, such as the states inside a digital computing system, or the states of a digital clock or indicator. However, it is very hard to see how the account as stated can be prevented from incorporating all too many states that are, intuitively, not digital states. For example, it will not only include the oil indicator lamp on my dashboard, but any lamp. After all, whatever the proper function of a particular lamp may be (such as to shed light on a desk), it fulfills that function by being "on" and does not fulfill it when "off". So, all lamps are in digital states. But so are all hats: Whatever the proper function of a particular hat may be (such as to shade the head), it fulfills that function by being "on the head" and does not fulfill it when "off the head". Wherever we look, we seem to find functionally determined clearly discrete tokens of types – but the flood is too hard to stem with just the sloppy notion of "function" that we used so far.

One diagnosis of the situation is the following: What we explained so far is really not *digital type*, but *type* in general.

The notion of function is really necessary for any type/token distinction. Take the word "tree". Which sounds or graphical shapes are tokens of that type? This is not just determined by some particular sound or graphical pattern but by whether a given sound or shape serves the function of being of that type. What is more, the type is functionally individuated: Which words are of the type "tree"? Those that serve the function of talking about trees.

So, the notion of function is not characteristic of digital types only, but of types quite generally. Of course, one might want to say that all types are digital, but it will become clear presently that this would constitute a deviation from current usage.

7. WHICH FUNCTIONAL TYPES ARE THE DIGITAL ONES?

We will discuss briefly some proposals for restricting the functional types to the desired digital ones. Each of these proposals contains a grain of truth that will be used in the final theory.

7.1. Computational

Since computational systems is what we are interested in, it seems natural to say that digital states are those that serve a computational function in a computational system. On this proposal, the notion of computation would be explained first, in formal, mathematical terms, and being digital is dependent on that explanation.

It so happens that all digital states can be part of a digital computational system, but do not have to be. It is irrelevant for its being digital whether a single warning light is or is not part of a computational system; it is still accurate to call it a digital representation. A digital clock can be just the output system of a purely analog time-keeping device. It appears that the class of digital non-computational states is not empty, so we cannot use computation to distinguish the digital types from the others. (And it follows from our previous discussion of digital vs. discontinuous that these are cases of digital states.) Having said that, it is a virtue of this proposal that it stresses the formal, syntactic, nature of computing – a feature that we will use presently.

7.2. In the System

In specifying the function of a state, one is often required to take recourse to the system of which it is a part, in particular to the *digital system*. This is clearly the case in conventional digital computers, so could this not provide a narrowing down of the right functional types?

In the case of conventional computing machines this proposal would work just like the one to start with "computing"– so there is clearly some unity here. However, if we allow systems to cover other systems that include the digital clock or warning light, how are we then going to limit the notion of "system" in the necessary way? It appears that a "system" will suffer from just the same defect as the "function" itself: anything is a system if it has some function or other. So, yes, all digital states are states in a system – but so are too many other states.

7.3. Representational

Perhaps we really have *two* notions here, the discrete states in general, and those discrete states that are representational, which we could call "digital"?

This has two disadvantages: It explains one obscure notion with an even more obscure one ("representation"). More importantly, it restricts digital states to those that represent, which is just what we should not do. It is precisely characteristic of the digital states in our prime example, the binary digital computer, that they do *not* represent. Just being of the state is what allows the system to perform its operations. The basic units have no meaning or representation – though they can be so *interpreted*, if desired, and in various ways, as we indicated above. So, we should not assume that all digital states are representational, but we must keep in mind that many are.

7.4. Pre-Defined

In at least some digital systems, notably in binary computers, the set of digital types is an explicitly pre-defined finite set. Is this the characteristic feature?

Is this necessary, however, in order to make a digital system, or a system with digital states? And *how* are they pre-defined? The proposal appears unnecessarily narrow, because it excludes digital states in systems that are not formally constructed. It would show a priori, for example, that the human brain does not have digital states (except if a creator pre-defined them).

7.5. Syntactical

The peculiar distinction of those types that are digital is that they are so devoid of content; it really does not matter at all how they are realized, what properties they have, provided that they are generally recognizable as being of the particular type. This aspect is probably best described by saying that digital types are *syntactic* types. A token of a syntactic type thus only contributes its *being of that type* to a larger syntactic system, nothing else. In particular, it cannot be said to have a meaning. In this sense, binary code is syntactic, and so are letters of an alphabet. Already words or lexemes are not syntactical, since they are semantically defined. Of course, lamps or hats are not syntactic items, so it appears that the requirement of syntactic definition of the type does our job to narrow down the many functional types.

8. WHICH STATES ARE DIGITAL?

In the case of the human nervous system, there are the questions whether it is a digital system on the level of mental functions, and whether it is a digital system on the level of cell properties and interactions. Many neuroscientists think of the latter in digital computational terms. [Piccinini now argues that "spikes" in neural activity do not constitute digital states. (35-37)] Computationalists think that representational function makes the mental level a digital computational system as well. Reproducing it in an AI computer system would thus yield mental properties.

REFERENCES

1. Shagrir O. Why we view the brain as a computer. Synthese. 2006;153(3):393-416.
2. Dretske F. Knowledge and the flow of information. Cambridge, Mass.: MIT Press; 1981.
3. Dretske F. Naturalizing the mind. Cambridge, Mass.: MIT Press; 1995.
4. Millikan RG. Language: A biological model. Oxford: Oxford University Press; 2005.
5. Macdonald G, Papineau D, editors. Teleosemantics: New philosophical essays. Oxford: Oxford University Pres; 2006.
6. Dodig-Crnkovic G. Epistemology naturalized: The info-computationalist approach. APA Newsletter on Philosophy and Computers. 2007;6(2):9-14.
7. Dodig-Crnkovic G, Müller VC. A dialogue concerning two world systems: Info-computational vs. mechanistic. In: Dodig-Crnkovic G,

Burgin M, editors. Information and computation: Essays on scientific and philosophical understanding of foundations of information and computation. Boston: World Scientific; 2011. p. 149-84.

8. Sandberg A, Bostrom N. Whole brain emulation: A roadmap. Oxford: Future of Humanity Institute; 2008.
9. Sandberg A. Feasibility of whole brain emulation. In: Müller VC, editor. Theory and Philosophy of Artificial Intelligence. Berlin: Springer; 2013. p. 251-64.
10. Harnad S. The symbol grounding problem. Physica D. 1990;42:335-46.
11. Raftopoulos A, Müller VC. The phenomenal content of experience. Mind and Language. 2006;21(2):187-219.
12. Raftopoulos A, Müller VC. Nonconceptual demonstrative reference. Philosophy and Phenomenological Research. 2006;72(2):251-85.
13. Goodman N. Languages of art. Indianapolis: Bobbs-Merril; 1968.
14. Lewis D. Analog and digital. Nous. 1971;5(3):321-7.
15. Haugeland J. Artificial intelligence: The very idea. Cambridge, Mass.: MIT Press; 1985.
16. Haugeland J. Analog and analog. Philosophical Topics. 1981;12:213-26.
17. Pylyshyn ZW. Computation and cognition. Cambridge, Mass.: MIT Press; 1984.
18. Shagrir O. Two dogmas of computationalism. Minds and Machines. 1997;7:321-44.
19. Blachowicz J. Analog representation beyond mental imagery. The Journal of Philosophy. 1997;94(2):55-84.
20. Negroponte N. Being digital. New York: Vintage; 1995.
21. Boghossian PA. Fear of knowledge: Against relativism and constructivism. Oxford: Oxford University Press; 2006. 139 p.
22. Turing A. Computing machinery and intelligence. Mind. 1950;LIX:433-60.
23. Searle JR. Mind: A brief introduction. Oxford: Oxford University Press; 2004. 224 p.
24. McCormack P, editor. Starmaking. Cambridge, Massachusetts: MIT Press; 1996.
25. Demopoulos W. On some fundamental distinctions of computationalism. Synthese. 1987;70:79-96.
26. Pylyshyn ZW. Computing in cognitive science. In: Posner MI, editor. Foundations of cognitive science. Cambridge, Mass.: MIT Press; 1989. p. 49-91.
27. Harnish RM. Minds, brains, computers: An historical introduction to the foundations of cognitive science. Oxford: Blackwell; 2002.
28. Floridi L. The method of levels of abstraction. Minds and Machines. 2008;18(3):303-29.
29. Müller VC. Representation in digital systems. In: Briggle A, Waelbers K, Brey P, editors. Current issues in computing and philosophy. Amsterdam: IOS Press; 2008. p. 116-21.
30. Boden MA. Escaping from the Chinese room. In: Boden MA, editor. The philosophy of artificial intelligence. Oxford: Oxford University Press; 1990. p. 89-104.
31. Boden MA. Mind as machine: A history of cognitive science. Oxford: Oxford University Press; 2006. 1704 p.
32. Haugeland J. Syntax, semantics, physics. In: Preston J, Bishop M, editors. Views into the Chinese room: New essays on Searle and artificial intelligence. Oxford: Oxford University Press; 2002. p. 379-92.
33. Krohs U. Der Funktionsbegriff in der Biologie. In: Bartels A, Stöckler M, editors. Wissenschaftstheorie: Texte zur Einführung. Paderborn: Mentis; 2007. p. forthcoming.
34. Berk V, Giani A, Cybenko G. Detection of covert channel encoding in network packet delays. Dartmouth College Department of Computer Science, Technical Reports. 2005(TR 536).
35. 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. 2004;141(2):175-215.
36. Piccinini G. Digits, strings, and spikes: Empirical evidence against computationalism. NA-CAP Conference; 28.07.2007; Chicago2007.
37. Piccinini G, Bahar S. Neural Computation and the Computational Theory of Cognition. Cognitive Science. 2012:1-36.