Computationalism, Enactivism, and Cognition: Turing machines as functionally closed systems

Mario Villalobos[1,2] and Joe Dewhurst[3]

[1] School of Philosophy and Psychology, Universidad de Tarapaca, Chile
[2] Instituto de Filosofa y Ciencias de la Complejidad, Santiago, Chile
[3] Philosophy, Psychology and Language Sciences, University of Edinburgh, UK

Abstract. In cognitive science, computationalism is the thesis that natural cognitive systems are computing systems. Traditionally, computationalism has understood computing and cognitive systems as functionally open systems, i.e., as systems that have functional entries through which they receive inputs, and exits through which they emit outputs. In opposition to this view, enactive theory claims that natural cognitive systems, unlike computing systems, are autonomous systems whose functional organization does not have inputs and outputs. Computationalism and enactivism seem to share an assumption that computing systems are input-output functional systems. In this paper, such an assumption will be critically reviewed by appealing to the cybernetic notion of functional closure. The notion of functional closure, as elaborated in Maturanas cybernetic neurophysiology, refers to a closed functional network in which, due to the circularity of the dynamics, we cannot distinguish inputs and outputs as intrinsic functional properties of the system. On the basis of this conceptualization, it will be argued that some paradigmatic cases of computing systems (notably a physically realized Turing machine) are actually functionally closed systems, and therefore computing systems without inputs and outputs. If this analysis is right, then the incompatibility that enactivists see between computing systems and organizationally closed functional systems would no longer hold, as it would not be true that computing systems must necessarily be understood as input-output systems.

Keywords: Computationalism Enactivism Functional closure Turing machine Input-output systems

1 Introduction

The computational theory of cognition, or cognitivist computationalism, has played a major role in the foundation and development of cognitive science [1]. Very simplified, computationalism is the thesis that cognitive systems are computing systems, or that cognition is essentially computation [1]. Traditionally, computationalism has understood computing systems as having at least two central properties: (i) representational status, and (ii) open functional organization

[2]. That computing systems have representational status means that computational states are at least partially individuated in terms of their representational content ([3], p.27). That computing systems have open functional organization, on the other hand, means that they have functional entries through which they receive inputs, and exits through which they emit outputs; i.e., that computational systems are input-output systems [4].

This representational and functionally open characterization of computing systems is typical of almost all classical versions of computationalism, no matter the kind of hypothesized vehicle or architecture. For example, the classical version of computationalism conceives of computing systems as input-output functions that manipulate symbolic (linguistic like) representational vehicles ([5] [6] [7] [8]), whereas the connectionist version thinks of them as input-output systems that operate over non-symbolic (subsymbolic, distributed, or even analog) representational vehicles ([9] [10] [11] [12]). What remains as a common assumption, despite these disagreements, is that computing systems are representational input-output systems.

Although dominant since the beginnings of cognitive science, computationalism has been criticized and challenged from several theoretical positions. One of the most serious and frontal attacks in this context is made by the enactive theory of cognition, or enactivism. Enactivism, like computationalism, comes in different versions. Here we will concentrate mainly on the original and canonical version elaborated by Varela, Thompson & Rosch [13], and further developed by authors such as Thompson [14], Di Paolo [15], and Froese [16]. Enactivism holds, basically, that cognitive systems are sense-making systems, not computing systems, or that cognition is not computation but sense-making. Enactivism takes living beings as its paradigmatic cognitive model, and sets a neat contrast between them and computing systems. According to enactivism, living systems, unlike computing systems, are organizationally closed functional systems (i.e., systems without inputs and outputs) that instead of representing (or processing information about) an external world, bring one forth by making sense of an otherwise meaningless external environment [14]. Put like this, enactivists seem to have good reasons to think that computationalism is incompatible with their view of cognition and cognitive systems in general. But is this necessarily so? Let us examine the issue closer.

A key aspect in the way enactivism opposes computationalism is that it does so by accepting, as a starting point, the traditional computationalist view above described. The enactive reasoning is more or less like this:

P1: Computing systems are representational systems.

P2: Computing systems are functionally open systems.

P3: Cognitive systems (i.e., living systems) are functionally closed non-representational systems.

Therefore,

C: Cognitive systems cannot be computing systems.

Starting from premises 1 and 2, the enactivist argument seems valid, but is it necessarily sound? We think it is not; i.e., we think that both premises 1 and 2 are objectionable. Out of these premises, in this paper we will focus on premise 2.

Premise 1, i.e., the representational view of computation, has been challenged by authors such as Piccinini, who, based on a mechanistic approach, has argued that computation does not presuppose representation ([3], p. 118, original emphasis). In line with this thought, Dewhurst [17] has explicitly argued that enactivism might find in Piccinini′s non-representational account an opportunity for reconciliation with computationalism. Premise 2, i.e., the functionally open view of computation, however, has not yet been challenged (as far as we know) by an explicit and systematic philosophical counterargument. The aim of this paper is to make such a challenge.

The idea that computing systems are inherently organized in terms of inputs and outputs will be critically reviewed by appealing to the cybernetic notion of functional closure. The notion of functional closure, as elaborated in Maturanas cybernetic neurophysiology ([18] [19] [20]), refers to a closed functional network in which, due to the circularity of the dynamics, we cannot distinguish inputs and outputs as intrinsic functional properties of the system. Originally intended as a way to understand the sensorimotor dynamics of living beings, the notion of functional closure has been deepened and expanded by Villalobos [4]. According to Villalobos, there exists functional closure in every system of processes in which (at least) some part of the procedural chain reenters the system thus functionally closing it on itself. Such systems, whatever their origins (i.e., natural or artificial), material constitution or thermodynamic regime, would not have inputs and outputs as intrinsic functional properties but only as observer-relative ascriptions [4]. On the basis of this conceptualization, it will be argued that some paradigmatic cases of computing systems (notably a physically realized Turing machine) are actually functionally closed systems, and therefore computing systems without inputs and outputs. If this analysis is right, then the incompatibility that enactivists see between computing systems and organizationally closed functional systems would no longer hold, as it would not be true that computing systems are necessarily understood as input-output systems.

2 Functional closure

The notion of functional closure, along with the associated epistemological implications that interest us here, first appears in Maturanas work on second-order cybernetics [18], although it has important precursors in first-order (i.e. classical) cybernetics (see e.g. [21] [22] [23]). It is used by Maturana to characterise the functional organisation of sensorimotor systems, which, according to him, do not have inputs or outputs as intrinsic features ([19] [20]). In this section we will try to demonstrate that certain computational systems, considered as sensorimotor circuits, can exhibit functional closure. Specifically, we will take the example of a concrete Turing machine. We will argue that a physical implementation of a Turing machine [4] constitutes an archetypal case of a functionally closed system, and thus constitutes an example of computation without input or output. Note that our strategy here is distinct from the approach taken by Piccinini [3] and Dewhurst [17], both of whom, whilst conceding the possibility of computation without input and output, relegate it to the status of an uninteresting edge case. By focusing on a physically implemented Turing machine we hope to demonstrate that functional closure is a central feature of certain computing mechanisms.

Let us start with a general characterisation of the notion of functional closure, as introduced by Maturana [18] in the context of sensoeffector systems. A sensoeffector system is a system composed of two (or more) connected transducers. Sensoeffector systems can be categorised in various ways. Here we need only distinguish two broad kinds: open (or linear) systems and closed (or circular) systems. Consider a basic thermostat, consisting of two transducers: a sensor (a bimetallic strip) and an effector (a radiator). If we put the sensor in one house and the effector in another, we will have an open sensoeffector system, where the sensors dynamics influences the effectors dynamics (through some wiring or connection), but not vice versa (i.e. the house containing the effector will warm up if the house containing the sensor is cold, but not vice versa).

A more conventional way of setting up a thermostat is with both components in the same house, forming a closed system where the dynamics of the effector loop back, via the ambient temperature of the air, to exert an influence on the dynamics of the sensor. This is what is meant by a functionally closed sensoeffector system (see figures 1 and 2).

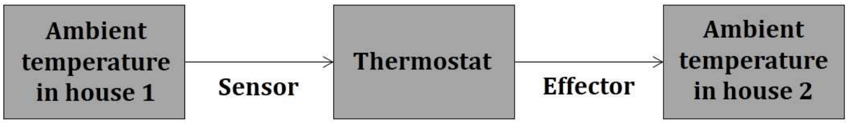

Fig. 1. Functionally open sensoeffector system.

---

[4] Physical implementations of the Turing machine, as originally formulated by Turing [24] (excepting the infinite extension of the tape) are scarce, but they exist. See for example the model built by Anders Nissen, Martin Have, Mikkel Vester and Sean Geggie (Aarhus University) (http://legoofdoom.blogspot.com), and the one built by Jeroen van den Bos and Davy Landman (Centrum Wiskunde & Informatica, Amsterdam) (http://www.legoturingmachine.org). See also the almost literal model built by Mike Davey (http://aturingmachine.com/index.php). Although of little or null interest from the practical point of view, the concrete operation of these physical models is, as we shall see, relevant for the purposes of the present analysis.

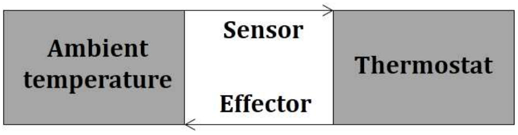

Fig. 2. Functionally closed sensoeffector system.

From the point of view of their physical constitution, both the open and the closed thermostatic system are the same. Whether set as an open or closed circuit, the thermostat is always composed of a sensor device, an effector device, and the wiring that links them. Elements such as the air of the room, the walls, the house and everything else, remain always external to the system. Yet from the functional point of view there is an interesting difference. When the system is open, its functional organization exhibits, as an intrinsic property, an entry through which it receives inputs (the sensor device), an intermediate mechanism (the wiring of the thermostat), and an exit through which it emits outputs (the effector device). But when the thermostatic circuit is closed on itself, and what we called before the output of the system is now at the same time the input for another part of the system, these distinctions do not hold any more as intrinsic properties of the system. From the functional point of view, the air of the room is now equivalent to the wiring of the thermostat; i.e., they both connect the sensor and effector devices, though in opposite directions and through different physical substrates. The air, functionally speaking, becomes a complementary wiring through which the thermostatic circuit closes on itself, and may be counted now as a part of the system, not as something external. The observer, of course, may still choose to treat the ambient air as the input to the system, i.e., as something external, but such a description will not reveal any intrinsic feature of the system.

Maturana [18] and more recently Villalobos ([4]; see also [25]), have argued that a living organisms sensorimotor system is organised as a functionally closed system, just like a thermostat with both components installed in the same house. The nervous system responds to the dynamics of its sensory organs by using its motor organs to establish a new environmental orientation, which in turn provokes a change to the dynamics of the sensory organs. The organism moves in its environment according to what it senses, and what it senses is determined by how it moves in its environment.

This sensorimotor circularity had already been noticed and analysed by classical cyberneticists, where it is referred to as feedback ([21] [22] [23]), and also by phenomenological theories of perception ([26]; see also [13]). However, Maturanas novel contribution was to make the following epistemological point: if a sensoeffector system is functionally closed, where are its entries (to receive inputs) and exits (to deliver outputs)? Where are the openings through which something gets into or goes out of the circuit? Maturana provides answers to these questions in the context of a living organisms sensorimotor organisation, and we follow Villalobos [4] in extending these answers to give a more general analysis of functional closure.

Consider, once again, the humble thermostat. As users we typically interpret the thermostat as receiving inputs via its sensor component, in the form of a measurement of the air temperature, and emitting outputs by switching a radiator on or off. These points are for us respectively the entry and the exit of the system. However, if we view the thermostat and its environment as a unitary feedback loop, we see that the ambient air temperature functions as just one more link within the circuit, not as something external. Considered as a functional circuit the thermostat is not open to its environment, the ambient air temperature, but rather closes on itself through it. By closes on itself we mean to say that a full functional description will treat the ambient air temperature as a part of the system, rather than as a distinct source of inputs or receiver of outputs.

Since the system exhibits functional closure in this way, it becomes equally valid to think of the effector component as an input device receiving, through the wiring, stimuli from the sensor, the sensor as an output device providing, through the wiring, stimuli to the effector, and the ambient air temperature as a functional node connecting the two. Of course we users do not usually think in these terms, because we are interested in the thermostat as a mechanism for controlling ambient air temperature rather than as a mechanism for controlling its own internal circuitry, but from a neutral observational point of view, both descriptions are equally valid. This indicates that the functional distinction between input and output is an observer-relative feature of our description, and is not intrinsic to the system itself. The point, as we have said, is not to deny that from the structural physical point of view there is always a clear distinction to be made between the thermostat and the ambient air temperature (between the system and the environment), but to see that from the functional point of view, i.e., when the thermostat is working, the ambient air temperature counts as a part of the circuit and not as something external. Neither does this mean that there is no distinction to be made between what is inside and what is outside the system as a whole.

For example, with respect to the closed thermostatic circuit as a whole (i.e., sensor and effector set in the same house), the ambient air temperature of other houses is clearly not a part of the system. So in this instance there is indeed a useful distinction to be made between what is functionally included or not in the system. What is functionally included depends on the particular coupling established by the system. If the thermostat, for instance, is set as a closed circuit in another house, room or building, the ambient air of these new locations will constitute the new functional links through which the system, invariably, closes on itself (just as new wirings will constitute the functional links through which the sensor device gets connected to the effector device). The functional dynamics of the system remain the same regardless of which environment it finds itself in, and included in these dynamics is the requirement that the system closes on itself through the environment, in the manner that we have described above.

This epistemological point extends to every functionally closed system, including, as we will now demonstrate in the next section, certain kinds of computing mechanism. Whilst the point might apply in principle to any feedback computing mechanism, we think it is most apparent in the organisation of a physically implemented Turing machine.

3 The Turing machine

A Turing machine is composed of a read/write device (or head) that operates upon a tape, and a mobile automaton that moves the head up and down the tape [24]. The head reads a character off the tape and then performs one of four possible actions, according to an algorithm contained within the automaton: moving along the tape, erasing or writing a character, and/or changing the internal state of the automaton (which governs future behaviour).

Usually, when talking about a Turing machine, both the mobile automaton (with its head) and the tape along which the automaton moves are considered parts of the machine. However, as Wells [27] has clearly pointed out, this view masks an important distinction that was originally made by Turing himself. The computing system formulated by Turing [24] was an analogue of a human being doing computations with the aid of paper and pencil (not of a human being doing computations within the head). In Turings original formulation, the mobile automaton with its head represents a human being that is able to read and manipulate pencil and paper, and the tape the paper upon which he or she writes (or erases) mathematical symbols. As Wells indicates in his interactive interpretation of the Turing machine (arguably just an elucidation of Turings original interpretation), the mobile automaton and its head represents the agent, and the tape its environment ([27], p. 272).

Following this characterization, and in terms of physical implementation, we see that the head is basically a sensor device that can identify characters on the tape, combined with an effector device that manipulates those characters. The automaton is a machine that mediates the behaviour of the two devices and controls a motor device that moves the whole system along the tape. A physically implemented Turing machine is therefore, in a non-trivial way, a sensoeffector system whose environment is constituted by the tape (and the symbols on the tape). The important point, however, is that just as in the case of the thermostat, this sensoeffector system is also a functionally closed system. The sensor device, via the automaton, influences the effector device and the motor, which in turn influences the sensor device via the medium of the tape. What the effector device and the motor do depends upon what the sensor reads, and what the sensor reads depends upon what the effector device and motor do. This functional organization, notice, is no different from the functionally closed organisation of the biological sensorimotor systems that Maturana was interested in, or of the thermostatic system that we described above.

The point is easy to see if we try to visualize what a functionally open computing system might be (see figures 3 and 4). The head would read some symbols in one tape, and the automaton, according to this reading and the prescribed algorithm, would command operations to be executed upon another tape (e.g., to write/erase symbols in another tape). Such a system would have a functional entry (input from, say, tape 1) and a functional exit (output to tape 2).

Fig. 3. Functionally open Turing machine.

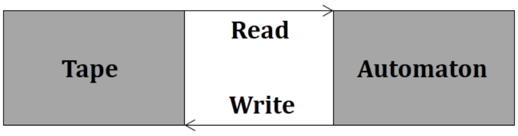

Fig. 4. Functionally closed Turing machine.

In the functionally open computing machine we can see, for example, that the output is effectively and unambiguously an output because what is done upon tape 2 never comes back to the system, i.e., it does not affect or condition what the reading device will find in the future in tape 1. The Turing machine, by contrast, is designed as a closed circuit wherein the tape (the environment) acts as a functional node that links the effectors dynamics to the sensors dynamics, thus becoming a part of the computing system as a whole.

A Turing machine, according to this view, is a functionally closed system. However, as in the example of the thermostat, an observer or user can still assign inputs and outputs to the Turing machine. Typically this means viewing the tape as providing inputs to (and receiving outputs from) the head, just as in the case of the thermostat it is the ambient air temperature that we are most interested in. Nonetheless, as we saw previously, this does not reveal any features that are intrinsic to a functionally closed system, but is rather a descriptive convention adopted by the observer according to her interests. It would be equally valid, though probably of little interest to the observer, to take the viewpoint of the wiring between the sensor device and the effector device, from where the sensor provides output and the effector consumes input. Since the Turing machine is a deterministic system, the observer would find a different but equally perfect mapping or function between these alternative inputs and outputs.

A physically implemented Turing machine, insofar as its functional dynamics are concerned, lacks input and output as intrinsic features of its functional organisation, and thus exhibits functional closure. Being an input-output system is therefore at best a contingent feature of computing mechanisms, not a necessary prerequisite as the classical view would have it.

4 Conclusion

We have argued that, contrary to the classical understanding, a computing mechanism does not necessarily require inputs or outputs. This is because a computing mechanism and its environment can be organized as a functionally closed system, where the sensor and effector surfaces are connected through the environment in such a way as to make any non-arbitrary distinction input and output impossible. We illustrated this with the example of a physically implemented Turing machine, which following Wells [27] we interpret as consisting of a computing mechanism (the automaton) and its environment (the tape). The Turing machine, thus interpreted, exhibits functional closure as one cannot non-arbitrarily distinguish between the tape providing inputs to the automaton on the one hand, or the automaton providing inputs to the tape on the other (and mutandis mutatis for outputs). Whilst an observer can decide to describe the system in terms of inputs and outputs, this description is non-essential to the functional structure of the system itself.

The physically Turing machine provides a conveniently simple case with which to illustrate functional closure, but the point generalizes to any computing mechanism whose interaction with the environment demonstrates the same kind of functional dynamics. For instance, a computational thermostat arranged so as to control the temperature of the same building that its sensors were placed in would also exhibit functional closure, as would any computing mechanism whose effectors are able to exert an influence, via the environment, on its own sensors. Understood in this way, the class of computing mechanisms that do not exhibit functionally closure is liable to turn out to be quite limited, as for many practical applications a computer is used in order to regulate some kind of environment variables in much the same way as a thermostat is used to regulate temperature. Perhaps most interestingly, any computational treatment of cognitive systems will inevitably also feature functional closure, as cognitive processes typically involve environmental feedback between sensor and effector surfaces. Our argument therefore eliminates another of the obstacles standing in the way of reconciliation between computationalist and enactivist approaches to cognitive science.

References

1. Rescorla, M.: The Computational Theory of Mind. The Standford Encyclopedia of Philosophy (2015)
2. Sprevak, M.: Computation, individuation, and the received view on representation. Studies in History and Philosophy of Science Part A, 41/3, pp. 260-70 (2010)

3. Piccinini, G.: Physical computation: A Mechanistic Account. UK: Oxford University Press (2015)
4. Villalobos, M.: The Biological Roots of Cognition and the Social Origins of Mind. PhD Thesis, University of Edinburgh (2015)
5. Fodor, J.: Representations: Philosophical Essays on the Foundations of Cognitive Science. Cambridge, MA: MIT Press (1981)
6. Fodor, J.: Concepts. Oxford: Clarendon Press (1998)
7. Pylyshyn, Z.: Computing and Cognition. Cambridge, MA: MIT Press (1984)
8. Fodor, J. & Pylyshyn, Z.: Connectionism and cognitive architecture. Cognition, 28, pp. 3-71 (1988)
9. Smolensky, P.: The constituent structure of connectionist mental states: A reply to Fodor and Pylyshyn. Southern Journal of Philosophy, 26 (Supplement), pp. 137163 (1987)
10. McClelland, J., Rumerhalt, D., and the PDP Research Group.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vols. 1 and 2. Cambridge, MA: MIT Press (1986)
11. Clark, A.: Microcognition: Philosophy, Cognitive Science, and Parallel Distributed Processing. Cambridge, MA: MIT Press (1989)
12. Churchland, P. S. & Sejnowski, T.: The computational brain. Cambridge, MA: MIT Press (1992)
13. Varela, F., Thompson, E. & Rosch, E.: The Embodied Mind: Cognitive Science and human experience. MIT Press: Cambridge (1991)
14. Thompson, E.: Mind in Life: Biology, phenomenology, and the sciences of mind. Cambridge, MA: Harvard University Press (2007)
15. Di Paolo, E.: Autopoiesis, adaptivity, teleology, agency. Phenomenology and the Cognitive Sciences, 4(4), pp. 429452 (2005)
16. Froese, T. & Ziemke, T.: Enactive artificial intelligence: investigating the systemic organization of life and mind. Artificial Intelligence, 173, pp. 466500 (2009)
17. Dewhurst, J.: Computing Mechanisms and Autopoietic Systems. In Muller (ed.), Computing and Philosophy, pp. 17-26 (2016)
18. Maturana, H.: The Biology of Cognition. BCL Report 9.0. Biological Computer Laboratory, Department of Electrical Engineer, University of Illinois (1970)
19. Maturana, H.: The organization of the living: A theory of the living organization. International Journal of Man-Machine Studies, 7, pp. 313332 (1975)
20. Maturana, H.: The biological foundations of self-consciousness and the physical domain of existence. In N. Luhmann, H. Maturana, M. Namiki, V. Redder, & F. Varela (Eds.), Beobachter: Convergenz der erkenntnistheorien? (pp. 47117). Mnchen: Wilhelm Fink Verlag. (2003)
21. Weiner, N.: Cybernetics. Cambridge, MA: MIT Press (1948)
22. Ashby, R.: An introduction to cybernetics. London: Chapman and Hall (1956)
23. Ashby, R.: Design for a Brain. London: Chapman and Hall (1960)
24. Turing, A.: On Computable Numbers, With an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society, 2/4, pp. 230265 (1937)
25. Villalobos, M. & Ward, D: Living systems: autopoiesis, autonomy and enaction. Philosophy and Technology, 28 (2), pp. 225-239 (2015)
26. Merleau-Ponty, M.: The Phenomenology of Perception. London: Routledge & Kegan Paul (1962)
27. Wells, A. J.: Turings analysis of computation and theories of cognitive architecture. Cognitive Science, 22 (3), pp. 269-294 (1998)