Synthese (2018) 195:1891–1908
https://doi.org/10.1007/s11229-017-1386-z

S.I. : NEUROSCIENCE AND ITS PHILOSOPHY

Enactive autonomy in computational systems

Mario Villalobos[1,2] · **Joe Dewhurst**[3]

Received: 17 October 2016 / Accepted: 22 March 2017 / Published online: 6 April 2017

Abstract In this paper we will demonstrate that a computational system can meet the criteria for autonomy laid down by classical enactivism. The two criteria that we will focus on are operational closure and structural determinism, and we will show that both can be applied to a basic example of a physically instantiated Turing machine. We will also address the question of precariousness, and briefly suggest that a precarious Turing machine could be designed. Our aim in this paper is to challenge the assumption that computational systems are necessarily heteronomous systems, to try and motivate in enactivism a more nuanced and less rigid conception of computational systems, and to demonstrate to computational theorists that they might find some interesting material within the enactivist tradition, despite its historical hostility towards computationalism.

Keywords Enactivism · Computationalism · Closure · Autonomy · Autopoietic theory

1 Introduction

Enactivism is one of the main theoretical approaches belonging to the so-called "post-cognitivist" paradigm in cognitive science (Wallace et al. 2007), the central negative thesis of which is that "cognition is not computational" (Fresco 2014, p. 215). Different branches of enactivism offer different reasons to justify this anti-computational

✉ Joe Dewhurst
joseph.e.dewhurst@gmail.com

Mario Villalobos
mario.kirmayr@gmail.com

1 Universidad de Tarapaca, Arica, Chile

2 Instituto de Filosofía y Ciencias de la Complejidad, Santiago, Chile

3 University of Edinburgh, Edinburgh, UK

attitude (e.g. Hutto and Myin 2013; Noë 2009). In this paper, we will concentrate on the 'classical' (also called 'canonical,' 'autopoietic,' or 'autonomist') version of enactivism (see Villalobos and Ward 2015 for discussion). Classical enactivism (hereafter 'CE', or simply 'enactivism') was originally introduced by Varela, Thompson and Rosch in *The Embodied Mind* (1991), and subsequently developed through a series of important contributions (Weber and Varela 2002; Thompson 2005, 2007; Di Paolo 2005, 2009; Froese and Ziemke 2009; Stewart et al. 2010; Di Paolo and Thompson 2014). Historically, CE has had two reasons to reject computational theories of cognition. First, CE assumes that cognition, at the most fundamental level, does not involve representations. To the extent that traditional computational theories presuppose that computation involves representation, CE rejects the notion of computation as a theoretical tool to characterize and explain cognitive systems. Second, CE argues that a distinctive feature of cognitive systems is that they are autonomous. To the extent that traditional computational systems, according to CE, are not autonomous, CE rejects the characterization or explanation of cognitive systems in computational terms. This reasoning has led CE, in general, to hold a strong anti-computationalist stance, and to be seen by most traditional computationalists as an 'outsider' and/or uninteresting research program. The anti-computationalist discourse of CE might also be off-putting to less traditional theorists who might otherwise have a legitimate interest in exploring and assimilating some enactivist insights.

However, as we will try to show here, there are reasons to think that CE's anti-computationalist stance might be more an aspect of its presentation as a post-cognitivist, emergent, and alternative research program than a strictly necessary component of its theoretical core. In this paper we will focus on the enactive concept of autonomy, and argue that the strong incompatibility that CE seems to see between autonomous systems and computational systems is not really so necessary. Enactivists, to the best of our knowledge, have never said that computational systems can be autonomous systems, but they have never explicitly denied such a possibility either.[1] In fact, we think that once our analysis has been developed, enactivists should find no serious problems with recognizing that some computational systems, under certain conditions, can exhibit autonomy.

We will demonstrate this by showing that the main criteria CE applies to characterize the nervous system as an autonomous system also apply to a physical instantiation of a traditional computational system, namely a Turing machine. In doing so, we will show that traditional computational systems, under certain specific conditions of implementation, can in fact exhibit the kind of autonomy that enactivism considers important in cognitive systems.

Overall, this paper offers a double invitation. First, to the enactivists, the invitation is to see and recognize that even traditional computing systems, when properly instantiated, can exhibit the kind of autonomy that they consider relevant to characterize and understand cognitive phenomena.[2] Second, and more importantly, to those

[1] We thank an anonymous reviewer for calling our attention to this point.

[2] This is *not*, notice, to invite CE to embrace the idea that cognitive systems may be computing systems. For CE, autonomy is certainly an important and distinctive feature of cognitive systems, but it is not the only one. Another equally important mark of cognitive systems, according to CE, has to do with the

who cultivate a computational approach but feel that there are certain enactivist ideas which would be worth incorporating, such as the idea of autonomy, the invitation is to not be discouraged by the enactivist anti-computational discourse. Viewing cognitive systems as computing systems, we will argue, is not incompatible with viewing them as autonomous systems. It is worth clarifying, however, that by doing this we do not intend to defend either the enactivist or the computationalist view of cognitive systems. Our purpose is to dissolve an alleged conceptual incompatibility between the notions of computation and autonomy, not to make a defence of any theory in particular. In our analysis, consequently, we will remain neutral about the correctness or not of both approaches.

Before starting, it is important to make explicit the notion of computation we have in mind for our analysis. This corresponds, to a large extent, to the one described by relatively recent mechanistic accounts of computation (see e.g. Miłkowski 2013; Fresco 2014; Piccinini 2015). What these accounts have in common is that they deny that computational states must be individuated according to their representational or semantic content, thus immediately removing one major objection that CE has historically held towards computationalism. Nonetheless, such accounts of computation remain fairly classical in the sense that the systems they describe perform operations over discrete states, unlike the dynamical or connectionist accounts with which CE might have more in common.[3] To be clear, then, we think that mechanistic accounts of computation provide an interesting target for analysis, insofar as they have removed one obstacle for CE, i.e. the notion of intrinsic representational content, but remain 'traditional' enough that it is not immediately obvious that they are compatible with CE's autonomy requirement.

We will first introduce some of the conceptual basis upon which CE rejects computationalism, focusing our analysis on the notion of autonomy as applied to the nervous system. We will then review the autopoietic/cybernetic foundations of the enactive notion of autonomy, and illustrate it with two simple examples (open and closed thermostatic systems). After that, we will examine Wells' (1998) discussion of the neglected role of the environment in the analysis of Turing machines, which will form the foundation for our demonstration of autonomy in physically implemented Turing machines. We will then proceed to demonstrate that a specific physical implementation of a Turing machine may exhibit, at least in theory, the features that CE considers

Footnote 2 continued
phenomenological dimension, i.e., the particular subjective experience associated with cognitive systems, which would be rooted in a "deep" continuity between life and mind (Weber and Varela 2002; Di Paolo 2005; Thompson 2007). Showing that some computing systems are autonomous systems, therefore, does not force CE to change anything about its anti-computationalist conception *of cognitive systems*, since there could be, for CE, other important (perhaps insurmountable) reasons to reject such a conception. We are totally aware of this fact, and consequently, do not pretend to use our analysis as an argument in that direction. We just aim to show that, according to CE's own definition, some computing systems can legitimately be considered autonomous systems. This point might be of potential interest for computationalists who are sympathetic to some of the ideas found in enactivism, even if enactivists themselves do not find it interesting.

[3] For example, connectionist computing systems such as unsupervised neural networks might already qualify as being at least partially autonomous. And something similar could be said about Bittorio, the interactive cellular automaton that Varela et al. (1991) use to illustrate the notions of enaction and autonomy. We thank an anonymous reviewer for bringing these points to our attention.

relevant to qualify the nervous system as an autonomous system.[4] Finally, we will give a preliminary response to the question of precariousness in Turing machines, before concluding with a short discussion of the potential reach of our analysis.

2 Enactive anti-computationalism: autonomy, self-determination and closure

Di Paolo and Thompson, perhaps the main representatives of contemporary CE, claim that the principal concept that differentiates enactivism from other approaches in cognitive science is the concept of autonomy (Di Paolo and Thompson 2014; Thompson 2005, 2007; Di Paolo 2009). The same idea can be found in other enactive representatives such as Froese and Ziemke (2009), and Barandiaran (2016). The enactive notion of autonomy includes several aspects and dimensions. From a metaphysical viewpoint, the notion of autonomy characterizes what CE calls self-determining systems; i.e., systems that instead of receiving inputs in the form of external instructions, specify their own domains of interaction (Varela et al. 1991; Thompson 2007). From a formal systemic approach, autonomy has to do with the closed (circular) organization present in certain systems (Varela 1979; Thompson 2007; Barandiaran 2016). Finally, from a more concrete and bottom-up approach, autonomy has to do with the constitutive precariousness of those systems that, as is paradigmatically the case in living organisms, need to sustain themselves and counteract the tendency towards thermodynamic decay (Di Paolo and Thompson 2014; Di Paolo 2009; Ruiz-Mirazo and Moreno 2004).

According to CE, "an autonomous system is a self-determining system, as distinguished from a system determined from the outside, or a heteronomous system" (Thompson 2007, p. 37). This form of autonomy, which we might call 'self-determination autonomy' (SDA), is, according to Thompson, present in biological systems and absent in artificial systems such as human-made machines:

> On the one hand, a living cell, a multicellular animal, an ant colony, or a human being behaves as a coherent, self-determining unity in its interactions with its environment. An automatic bank machine, on the other hand, is determined and controlled from the outside (Thompson 2007, p. 37)

Additionally, autonomous systems are characterized by their organizational and operational closure. "*Organizational closure* refers to the self-referential (*circular and recursive*) network of relations that defines the system as a unity, and *operational closure* to the *re-entrant* and recurrent dynamics of such a system" (Thompson 2007, p. 45, emphasis partially modified). CE follows Varela in holding that "[e]very autonomous system is organizationally closed" (Varela 1979, p. 58). This form of autonomy, which we might call 'organizational-operational autonomy' (OA), can also be recognized,

[4] Note that we are not concerned here with the question of whether autopoietic systems, or biological systems broadly speaking, are Turing computable (see e.g. Letelier et al. 2003; Rosen 2000; Thompson 2007). Whilst this is an interesting question, our concern is different: whether physically implemented Turing machines may meet the criteria that enactivism considers relevant to qualify the nervous system as an autonomous system. We thank an anonymous reviewer for bringing this issue to our attention.

CE claims, at different levels of the biological realm; single cells, nervous systems, immune systems, colonies, social systems, ecosystems, etc.

OA, however, according to enactivists, needs to be complemented with the idea of precariousness (Di Paolo 2009; Di Paolo and Thompson 2014). This is because OA, as originally formulated by Varela (1979), is a formal property that may be found, in principle, in systems that are not genuinely autonomous in the full sense that matters to CE (e.g., cellular automata, as abstract models, may meet the criterion of operational closure). Thus, to qualify a system as genuinely autonomous, the system must also be precarious. The enactive notion of precariousness is an important one, and will be analyzed towards the end of the paper (Sect. 7), focusing on the case of a physically implemented Turing machine.

For now, let us begin our analysis of the enactive anti-computational stance by addressing the first two types of autonomy, namely SDA and OA. A key feature of autonomous systems, according to CE, is that they are not organized in terms of inputs, processing, and outputs. That is, autonomous systems "need to be seen as sources of their own activity, specifying their own domain of interactions, not as transducers or functions for converting *input instructions* into *output products*" (Thompson 2007, p. 46, emphasis added). The presence or absence of inputs and outputs marks a crucial difference between autonomous and heteronomous systems, and is, according to CE, one of the features that makes computational systems non-cognitive:

> A heteronomous system is one whose organization is defined by input-output information flow and external mechanisms of control. Traditional computational systems, cognitivist or connectionist,[5] are heteronomous systems. (...) An autonomous system, however, (…) does not have inputs and outputs *in the usual sense*, and determines the cognitive domains in which it operates. (Thompson 2007, p. 43, emphasis and footnote added)

The nuance "in the usual sense" is important here. The usual sense in which, according to CE, it is said that a system has inputs and outputs, and that is inadmissible for autonomous systems, has two aspects. The first aspect has to do with the notion of input as an external instruction that is received and "obeyed" (internalized) by the system, specifying or determining its responses. This, according to CE, cannot happen in an autonomous system, because autonomous systems are self-determining systems; they determine, according to their own dynamics, how to react in response to external stimuli, and more than that, they specify ("choose"), according to their own constitution or nature, what does and what does not count, out of the many events that take place in the environment, as a stimulus for them. The second aspect has to do with the notions of inputs and outputs as "entries" and "exits" of the system, and refers to the distinction between operationally open (linear) and operationally closed (circular) systems. Whereas linear systems exhibit functional entries and exits, circular systems do not (we will comment in more detail on this distinction in Sect. 3).

[5] Here we assume Thompson has in mind *supervised* connectionist networks, as it seems plausible that unsupervised networks might in some cases exhibit autonomy. We would like to thank an anonymous reviewer for bringing this point to our attention.

These are, according to CE, the usual senses in which it is said that a system has inputs and outputs, and that do not apply to autonomous systems but do apply to heteronomous systems. What about the "unusual" sense in which the enactivist might recognize that an autonomous system has inputs and outputs? Here the enactivist has in mind a notion of input as mere perturbation or triggering factor (which is innocent because it does not imply an instructive interaction): i.e., "[i]nputs are described as perturbations to the system's intrinsic dynamics, rather than as instructions to be followed" (Thompson 2007, p. 11). Correlatively, the innocent notion of output is simply a system's response that is not the product of following external instructions, but rather the expression of its own intrinsic dynamics.

Let us see, following Thompson's canonical presentation (2007), how CE applies these categories (closure, circularity, self-determination, etc.) to the special case of the nervous system. Thompson claims that "the nervous system is an autonomous dynamic system" (2007, p. 13) whose "fundamental logic (…) is to couple movement and a stream of sensory activity in a continuous circular fashion" (*ibid*., p. 47). Given this circularity or closure, "neuronal networks establish and maintain a sensorimotor cycle through which what the animal senses depends directly on how it moves, and how it moves depends directly on what it senses" (*ibid*., p. 47). Also, thanks to the presence of the nervous system, "[n]o animal is a mere passive respondent; every animal meets the environment *on its own sensorimotor terms*" (*ibid*., p. 47. Emphasis added). To expand the latter idea, which highlights the self-determining dynamics conferred by the nervous system, Thompson resorts to a famous passage from Merleau-Ponty's *The structure of behaviour*:

> [T]he form of the excitant [stimulus] is created by the organism itself, by its proper manner of offering itself to actions from outside. (…) [I]t is the organism itself—according to the proper nature of its receptors, the thresholds of its nerve centers and the movements of the organs—which chooses the stimuli in the physical world to which it will be sensitive. (Merleau-Ponty 1963: 13. Quoted by Thompson 2007, p. 48)

This characterization, that according to Thompson "clearly expresses an autonomy perspective" (*ibid*., p. 48), tells us that it is the nervous system itself (its structure, the nature of its sensors, its thresholds, its motor effects, etc.) that determines ("chooses") what does and what does not count as a stimulus for it.

For us, the relevant point is that, according to CE, among the essential features that make the nervous system an autonomous system are 1) its closed (circular) organization, and 2) its ability to specify its own domain of interactions with the environment. The first condition points to OA, and the second condition to SDA. Jointly, they define the nervous system as a system without inputs and outputs (in the 'usual' sense that is relevant for CE). In what follows we will try to show that physically instantiated computational systems can in fact meet the two aforementioned conditions, which, according to CE, characterize the nervous system as an autonomous system.

We will first review the theoretical roots of the enactive notion of autonomy, in the specific senses of organizational/operational closure and self-determination, and the original philosophical motivations that link these notions with the analysis of input-output systems. Tracing CE's ideas back to its original source will help us to

see that, well examined, OA and SDA are conditions that apply to quite traditional computational systems. The original source of CE's notion of autonomy, as regarding OA and SDA, lies in Maturana's autopoietic theory of cognition. In the autopoietic theory (Maturana 1970, 1975, 1981, 1987, 2003; Villalobos 2015) autonomy is a concept that appears in the description of living beings, but that is not assigned a central role. Nonetheless, the notions of organizational/operational closure and self-determination that underpin CE's concept of autonomy do appear in the autopoietic literature and play an important role (Villalobos and Ward 2015), as we shall now see.

3 Operational and functional closure: circuits without inputs and outputs

The notion of organizational/operational closure, as associated with the critical analysis of the notions of input and output, has its root in Maturana's notion of functional closure. The notion of functional closure first appears in Maturana's work on second-order cybernetics (Maturana 1970, 1975). It is used by Maturana to characterize the functional organization of sensorimotor systems, which, according to him, do not have inputs and outputs as intrinsic features (Maturana 1975, 2003).

Let us start with a general characterization of the notion of functional closure, as introduced in the context of sensoeffector systems (Villalobos 2015). A sensoeffector system is a system composed of two (or more) connected transducers. Sensoeffector systems can be categorized in various ways. Here we just need to 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 sensor's dynamics influences the effector's 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 Figs. 1, 2).

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

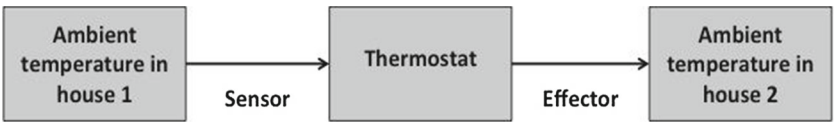

Fig. 1 Functionally open sensoeffector system

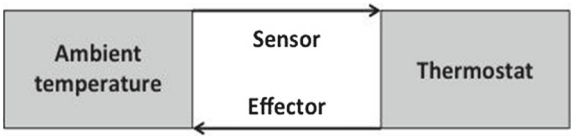

Fig. 2 Functionally closed sensoeffector system

else, remain always external to the system. Yet from the *functional* point of view, i.e., when the thermostat is working, there is an interesting difference. When the system is open, its functional organization exhibits 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 via the ambient air temperature, what we called before the output of the system is now at the same time the input for another part of the system, and there are, functionally speaking, no entries and exits to the system anymore. 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 something external to the system, but such a description is not justified based on the functional organization of the system alone.

Maturana and Varela (1980), and more recently Villalobos (2015; see also Villalobos and Ward 2015), have argued that a living organism's sensorimotor system is organized 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, notice, is just the same circularity that CE has in mind when talking about the organizational/operational closure of the nervous system.

> The fundamental logic of the nervous system is to couple movement and a stream of sensory activity in a continuous *circular* fashion. (…) [N]euronal networks establish and maintain a *sensorimotor cycle* through which what the animal senses depends directly on how it moves, and how it moves depends directly on what it senses. (Thompson 2007, p. 47)

Both Maturana's concept of functional closure and the enactive notion of organizational/operational closure are basically ways of referring to the circular organization of the nervous system. Now, the circular nature of the nervous system, as depicted in these notions of closure, had already been noticed and analyzed by classical cyberneticists, where it is referred to as feedback (see Weiner 1948; Ashby 1956, 1960), and also by phenomenological theories of perception (Merleau-Ponty 1963). However, Maturana's novel contribution, which enactivism follows in its analysis of input-output

systems, 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?

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 (i.e. the ambient air temperature), but rather closes on itself through it. By "closes on itself" Maturana means 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 (cf. Virgo et al. 2011). The system, understood as a functional circuit, does not have inputs and 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 as 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. Maturana's 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, 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 temperature 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 analysis, we think, extends to every functionally closed system, including, as we will now demonstrate, certain kinds of computing mechanism.

4 The computational environment of a Turing machine

A Turing machine is usually characterized as a device composed of three components: a tape, a read/write head, and a mobile automaton that moves the head up and down the tape (see e.g. Barker-Plummer 2016, Sect. 1). The tape is divided into equal sections, each of which contains one character from a finite, discrete set – for the sake of simplicity we will assume that the characters are drawn from the binary set {0, 1}. The head reads a character off the tape and then performs one or more of four possible actions, according to an algorithm contained within the automaton: moving along the tape, erasing a character, writing a character, and changing the internal state of the automaton (which governs future behaviour).

In the characterization given above, the tape was described as a component of the Turing machine, and in modern contexts it is usually equated with the memory component of a conventional digital computer. However, as Wells (1998) has clearly pointed out, this description masks an important distinction that was originally made by Turing himself. Turing's original (1936) formulation of the machine was inspired by an analogy with a human performing computations on paper. The system composed of the mobile automaton plus the read/write head was supposed to correspond to the human plus their pencil and eraser, whilst the tape corresponds to the paper upon which he or she writes and erases mathematical notation. Wells argues for an "interactive" reinterpretation of the Turing machine (arguably just an elucidation of Turing's original interpretation), where the mobile automaton and its head are "the agent", and the tape its "environment" (1998, p. 272).[6]

One consequence of this approach is that the computational theory of mind it implies is inherently world involving, in contrast with classical computationalist accounts of cognition, and perhaps more in line with the enactivist tradition. Distinguishing between the role played by the brain (equivalent to that of Turing's automaton) and the environment (equivalent to that of Turing's tape) "leads to a view of cognitive computations as processes of structured interaction between the control architecture of the brain and the input architecture of the environment" (Wells 1998, p. 280). This helps to resolve various issues that Wells identifies with the classical approach, including what he calls the evolutionary problem and the transduction problem. We will not discuss these points any further here, although we do note that they are,

[6] The inspiration for the Turing machine was a person performing computations on paper. In that case, the paper, being external to the agent, is a part of the environment, but as part of the environment it is nonetheless relevant to the computational process. The same is the case with the tape of the Turing machine. So, Wells might have qualified the concept by speaking of the tape as the 'computational environment' of the Turing machine. A person is also surrounded by air, chairs, walls, etc., all of which is part of its environment in a wide sense, but they are not a part of the computational process. The air, for example, is an enabling condition for the person. In that wide sense, the air is necessary for the person to perform computations, but that does not turn the air into a component of the computational process. Similarly, a physical Turing machine will have the tape as its computationally relevant environment, and the wider physical environment as enabling conditions (e.g., energy supply, ambient temperature, humidity, etc.).

perhaps, potential areas where a synthesis between the enactive and computational frameworks might prove fruitful.[7]

In the next section, we will demonstrate that a physically implemented Turing machine exhibits organizational/operational closure, and therefore exhibits OA.

5 Closure and OA in a physically implemented Turing machine

We saw previously that the enactive notion of organizational/operational closure, like Maturana's concept of functional closure, refers essentially to the circular (cyclic, re-entrant) dynamics of sensoeffector systems (of which the nervous system is an example). And we also saw that CE considers organizational/operational closure to be one of the marks of an autonomous system. In this section, we will demonstrate that a Turing machine, understood according to Wells' interactive interpretation, exhibits organizational/operational closure, and therefore exhibits OA.

Following Wells' characterization, we can see that the read/write head of a Turing machine is basically a sensor device that can identify characters on the tape, combined with an effector device that manipulates those characters, whilst the tape itself constitutes an environment external to the automaton. 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, from the functional/computational point of view, 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 typically forms a functionally closed circuit. The sensor device, via the automaton, influences the motor and the effector device, which in turn influences the sensor device via the medium of the tape. What the effector device and 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 operationally closed organization of the biological sensorimotor systems that Maturana was interested in, or of the thermostatic system that we described in Sect. 3.

The point is easy to see if we try to visualize what a functionally open Turing machine might be (see Fig. 3). The head would read some symbols on 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 on, and to move up or down along, a second tape). Such a system would have a functional entry (input from tape 1) and a functional exit (output to tape 2).

In the functionally open Turing machine we can see that what is done upon tape 2 never "comes back" to the system, so to speak; i.e., the movements along tape 2 do not affect or condition what the reading device will find in the future on tape 1. The conventional Turing machine, by contrast, is designed as a closed circuit wherein the tape (the environment) acts as a functional node that links the effector's dynamics to

[7] Wells is aware of the potential comparison between his reconceived Turing machine and the work of Maturana & Varela, although he does not pursue this comparison in his paper, partly due to reservations about the "relativistic epistemology" that he thinks their approach leads to (Wells 1998, p. 280).

Fig. 3 Functionally open Turing machine

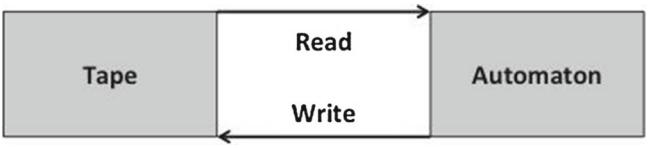

Fig. 4 Functionally closed Turing machine

the sensor's dynamics, thus becoming a part of the computing system as a whole (see Fig. 4).

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 typically characterize as external to the thermostatic system. 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 outputs and the effector consumes inputs. 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 conventional physically implemented Turing machine, insofar as its functional dynamics are concerned, lacks input and output in the systemic (organizational/operational) sense that is relevant for CE, and therefore exhibits OA. OA, however, is just one of the criteria CE considers relevant to qualify the nervous system as an autonomous system. What about SDA? Can a physical Turing machine meet this criterion? To answer this question, it will be useful, once more, to trace CE's ideas back to its original source in the autopoietic theory of cognition.

6 Structural determinism: self-determination in a physically implemented Turing machine

What enactivists call self-determination, i.e., the fact that the system does not admit external specifications or determinations, corresponds basically to what Maturana (1975) calls 'structural determinism'. A structurally determined system is a system within which everything that happens is determined or specified by its own structure – by its own constitution or composition. External factors or interactions can trigger or initiate certain structural changes in the system (reactions, processes) but cannot specify or instruct them.

> [T]he structural changes that a composite unity [in Maturana's jargon 'composite unity' means 'system'] undergoes as a result of an interaction are (…) determined by the structure of the composite unity. Therefore, an external agent which interacts with a composite unity only triggers, rather than determining in it a structural change. Since this is a constitutive condition for composite unities, nothing external to them can specify what happens in them: *there are no instructive interactions for composite unities*. (Maturana 2003, p. 61, emphasis added)

This is basically the sense in which, as we saw, CE claims that self-determining systems do not receive instructional inputs. Now, according to Maturana, the structural determinism of systems defines four different domains of states and interactions, only one of which (c) is relevant for our analysis:

> [C]omposite unities [i.e., systems] are structure-determined systems in the sense that everything that happens in them is determined by their structure. This can be systematically expressed by saying that the structure of composite unity determines it at every instant:
> (a) the domain of all the structural changes that it may undergo with conservation of organization (class identity) and adaptation at that instant; I call this domain the instantaneous domain of the possible changes of state of the composite unity.
> (b) the domain of all the structural changes that it may undergo with loss of organization and adaptation at that instant; I call this domain the instantaneous domain of the possible disintegrations of the composite unity.
> (c) *the domain of all the different structural configurations of the medium that it admits at that instant in interactions that trigger in it changes of state; I call this domain the instantaneous domain of the possible perturbations of the composite unity.*
> (d) the domain of all the different structural configurations of the medium that it admits at that instant in interactions that trigger in it its disintegration; I call this domain the instantaneous domain of the possible destructive interactions of the composite unity. (Maturana 2003, pp. 61–62, emphasis added)

The domain c) refers to the selective nature of the system regarding its interactions. Because of its structural determination, a system interacts with certain elements or aspects of its environment, i.e. with certain structural configurations, and not with others. The structural constitution of a system specifies which aspect of its environment triggers (or not) in it a change of state, and which external factor counts (or not) as a perturbation. A marble table, in its structural composition, determines a particular domain of perturbations wherein, for example, a subtle touch of a finger cannot trigger any change of state. A mobile phone, in its structural composition, specifies a domain of perturbations wherein a subtle touch of a finger (on its touch screen) does trigger a change of state (cf. Villalobos 2015). Structural determinism means, basically, that a system faces the environment on its own terms, configuring its own domains of interactions. This is the sense, recall, in which CE refers to the nervous system as a self-determining system. It is the structure of the nervous system (the particular

nature of its sensor devices, its specific thresholds, etc.) that determines, out of the many physical events and properties of the surrounding, what counts and what does not count as a perturbation (stimulus). Thanks to the self-determining nature of the nervous system, "every animal meets the environment on its own sensorimotor terms" (Thompson 2007, p. 47).

Before addressing the case of the Turing machine, notice that, interestingly, Maturana speaks of "composite unities" (systems) in general, without making any distinction among them. This is because, according to Maturana, every physical system, whether alive or not, natural or artificial, is a structurally determined system (Maturana 1975, 1981, 1987, 2003). Indeed, in contrast with CE, when Maturana illustrates the notion of structural determinism, he does not use biological systems but man-made machines as examples:

> We know [that systems are structurally determined] very well from using any of those pushbutton machines, in which if you push a button something happens—it washes, it glows, it plays music—which is not determined by your pushing the button, but, rather, is triggered by the pushing of the button (...) You do not instruct a system, you do not specify what has to happen in the system. If you start a tape recorder, you do not instruct it. You trigger it. (Maturana 1987, p. 73)

We think that this characterization is correct, and we will argue now that a physical Turing machine is a structurally determined system. In doing so, we hope to show that a physical Turing machine is a self-determining system in the sense that is relevant for CE.

A physical Turing machine, as we saw in the previous section, has a sensor device (the head) that "reads" the inscriptions on the tape (the environment). The inscriptions on the tape trigger certain changes of state and reactions in the machine, all of which are specified by the configuration of the machine itself. If the sensor device is, say, a photosensitive plate, it is the particular composition of this plate, its threshold and resolution capability, that determines what kind of configuration or pattern in the environment (i.e., the tape) counts or not as a triggering factor. Too faint inscriptions, or too little characters, might not be readable, and thus trigger no change of state in the machine. Now, when the sensor does read the inscription on the tape, the consequent change of state in the machine is determined by the structure and configuration of the machine itself, not instructed or specified by the inscription. If the machine, encountering the inscription "0" on the tape, erases it, writes a "1" and moves upward, that action comes determined by the configuration of the machine, not by the nature of the digit "0." The same digit put in front of a Turing machine set up in a different way will trigger different changes of state and actions (e.g., leaving the inscription and moving downward). In each case, it is the configuration of the Turing machine that specifies what to do in front of a given character.[8]

[8] Although we have taken as an example a special-purpose Turing machine, we think the analysis also applies, in essence, to a Universal Turing machine. Universal Turing machines are said to read and receive "instructions" from the tape, yet, as in the example of the machine provided by Maturana (Sect. 6), such a description would be rather a shortcut to recognise the fact that the machine does what it does only because it has been made and designed in a particular way. Put the same "instructions" in front of another Turing machine (e.g. a special-purpose one), and the "instructions" will lose all of their instructive power. Anyway,

A physically instantiated Turing machine, through its structure and configuration, specifies its own domain of interactions (perturbations) and meets the environment on its own terms. It is a system that does not admit inputs in the instructional sense that is relevant for CE, and therefore a system that exhibits SDA.

7 Precariousness in a physically implemented Turing machine

If our analysis in the previous sections is correct, then we have shown that a physical Turing machine, a familiar and representative example of a computational system, can exhibit two essential features of autonomous systems, namely self-determination and organizational/operational closure. However, as we noted in Sect. 2, there is a third requirement that CE establishes with respect to the notion of autonomy. According to CE, only precarious systems can be recognized as autonomous systems (Di Paolo 2009; Di Paolo and Thompson 2014). This requirement, which we could call 'precarious autonomy' (PA), is understood as the condition in which, "in the absence of the *enabling* relations established by the *operationally closed* network, a process belonging to the network will stop or run down" (Di Paolo and Thompson 2014, p. 72, emphasis added). Enabling relations are stronger than other kinds of relations such as coupling or control relations. The latter entail (merely) contextual effects (e.g. modulation), whereas the former entail a kind of strong dependency. Di Paolo and Thompson (2014) give the example of a thermostat. The process of temperature regulation performed by a thermostat will change if we reset the target temperature value (contextual condition), but will fully stop if we unplug the whole apparatus (enabling condition). The presence of enabling relations, however, is not sufficient for a system to exhibit PA. PA requires these relations to be organized as an operationally closed network, in such a way that the functioning of the network critically depends on, and only on, the continuity of the network itself. Put in negative terms:

> When a process is enabled by the operationally closed network and by external processes as well, if the network is removed and the process remains—in the new circumstances—thanks only to the external support, then that process is not precarious (Di Paolo and Thompson 2014, p. 72)

According to CE, biological systems in general, from cells to ecological networks, including the nervous system, are precarious systems in this specific sense, although not necessarily at the same level of realization (Di Paolo 2009; Thompson and Stapleton 2009; Di Paolo and Thompson 2014). Now, our question is: could a physical computational system such as a Turing machine be precarious in this sense? Or, in negative terms, is there any consideration in the physical implementation of a Turing machine that prevents, in principle, a precarious implementation? We think that there is not. Leaving aside engineering technicalities, it should be apparent that the material implementation of a Turing machine might be, in principle, as robust or as precarious

Footnote 8 continued
beyond this note, it should be clear that to make our point we do not need to demonstrate that *all* Turing machines conform to the criterion of self-determination or structural determinism. It suffices to show that some of them do.

as we want, so long as the machine continues to possess the correct computational structure. All that is needed, if we follow CE's criteria, is that the machine instantiates, with the participation of its computations, some kind of operationally closed system of enabling conditions.

Usually, if we think of implementing a computing system, we try to make it as robust and durable as possible. For example, we design the machine so that its energy supply remains independent from the computations performed by the machine (i.e. the computations of the machine may be right or wrong, but that does not impact or alter the energy supply for the machine). A computing machine, typically, is conceived of to solve a problem that is relevant for us as users, not for the machine itself. Yet this kind of implementation and task design is necessary only in order to satisfy our preferences, needs, or purposes as users, and does not reveal any necessary element, from the theoretical point of view, in the constitution of a computing machine. We can conceive of, and perhaps build, a Turing machine whose computations address a problem that is relevant only for the machine itself. Crucially, this could be a problem in which the very continuity of its computations is at stake.

We could imagine, for example, a mobile Turing machine whose role as a computing system is enabled by the supply of some kind of chemical energy, and whose computations, in turn, address precisely the problem of how to find and get enough sources of said energy. In that case, it seems, we would be close to a system constituted as a closed network of enabling conditions. The machine's computations are enabled by the energy supply, and the energy supply, in turn, is enabled by the computations. This would be a case, it seems, of a physical Turing machine exhibiting at least some degree of PA, or something close to that.

Of course, merely visualizing the possibility of such a physical Turing machine does not demonstrate, in any conclusive way, that a physical computing system can exhibit, in the full sense that matters to CE, precariousness. For there are, according to CE, important implications and ramifications associated with the condition of precariousness, such as the phenomena of normativity, teleology, and individuation (Di Paolo and Thompson 2014), none of which we are addressing here. However, we think that this exercise suggests, albeit in a modest sense, at least the conceptual possibility of a precarious computational system.

8 Conclusion

Our aim in this paper has been to demonstrate that, under certain conditions of implementation, it is possible for a computational system to exhibit the kind of autonomy that CE assigns to a cognitive system such as the nervous system. Our demonstration began by first clarifying (in Sect. 2) the three criteria required by enactivism of an autonomous system: organizational/operational closure, self-determination, and precariousness. We then proceeded (in Sect. 3) to describe the first of these requirements in more detail, with reference to the autopoietic theory that enactivism draws upon. In Sect. 4 we introduced the idea (first noted by Wells) of treating the tape of a Turing machine as its environment, and then in Sect. 5 we demonstrated that a Turing machine (thus understood) exhibits organizational/operational closure. Then, in Sect. 6, we

showed how the very same Turing machine also exhibits self-determination, again with reference to autopoietic theory. Finally, in Sect. 7, we addressed the question of precariousness, and argued that a physically implemented Turing machine could under some circumstances qualify as a precarious system. All of this leads us to conclude that at least a certain class of computational system, i.e. a simple Turing machine, can meet the requirements for autonomy imposed by CE.

The purpose of the demonstration is twofold: on the one hand, to encourage computational theorists to look more closely at concepts from enactivism (and related traditions) that they might find interesting, and not to be put off by the purported anti-computationalism of these traditions; and on the other, to encourage enactivists to consider that traditional computing systems might sometimes be autonomous systems, and to reflect on what this would mean for their anti-computationalism. What we have not done is attempted to argue for or defend either position (enactivism or computationalism). Our hope is simply to have clarified the relationship between the two positions by demonstrating that enactivism, on the basis of the autonomy requirement alone, need not rule out computational characterizations of the nervous system.

Acknowledgements Several anonymous reviewers gave detailed comments on various versions of this paper, all of which have helped us improve it greatly. We are especially thankful to Gualtiero Piccinini, who has provided invaluable encouragement and support for this project over the past two years. Mario Villalobos' contributions to this paper were funded by a grant from the Comisión Nacional de Investigación Científica y Tecnológica, Chile (FONDECYT INICIACIÓN 11150652), and partially supported by Performance Agreement UTA-MINEDUC.

References

Ashby, R. (1956). *An introduction to cybernetics*. London: Chapman and Hall.

Ashby, R. (1960). *Design for a brain*. London: Chapman and Hall.

Barandiaran, X. E. (2016). Autonomy and enactivism: Towards a theory of sensorimotor autonomous agency. *Topoi*. doi:10.1007/s11245-016-9365-4.

Barker-Plummer, D. (2016). Turing Machines. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy*. (Spring 2016 Edition). <http://plato.stanford.edu/archives/spr2016/entries/turing-machine/>

Di Paolo, E. (2005). Autopoiesis, adaptivity, teleology, agency. *Phenomenology and the Cognitive Sciences*, *4*(4), 429–452.

Di Paolo, E. A. (2009). Extended Life. *Topoi*, *28*, 9–21.

Di Paolo, E. A., & Thompson, E. (2014). The enactive approach. In L. Shapiro (Ed.), *The Routledge handbook of embodied cognition* (pp. 68–78). New York: Routledge Press.

Fresco, N. (2014). *Physical computation and cognitive science*. Berlin, Heidelberg: Springer-Verlag.

Froese, T., & Ziemke, T. (2009). Enactive artificial intelligence: Investigating the systemic organization of life and mind. *Artificial Intelligence*, *173*, 466–500.

Hutto, D., & Myin, E. (2013). *Radicalizing enactivism: Basic minds without content*. Cambridge, MA: MIT Press.

Letelier, J. C., Marín, G., & Mpodozis, J. (2003). Autopoietic and (M, R) systems. *Journal of Theoretical Biology*, *222*, 261–272.

Maturana, H. (1970). Biology of cognition. *Biological Computer Laboratory, BCL Report 9*, University of Illinois, Urbana.

Maturana, H. (1975). The organization of the living: A theory of the living organization. *International Journal of Man-Machine studies*, *7*, 313–332.
Maturana, H. (1981). Autopoiesis. In M. Zeleny (Ed.), *Autopoiesis: A theory of living organization* (pp. 21–33). New York; Oxford: North Holland.
Maturana, H. (1987). Everything is said by an observer. In W. I. Thompson (Ed.), *GAIA: A way of knowing* (pp. 65–82). Hudson, N.Y.: Lindisfarne Press.
Maturana, H. (2003). 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. 47–117). München: Wilhelm Fink Verlag.
Maturana, H., & Varela, F. (1980). *Autopoiesis and cognition: The realization of the living*. Dordrecht, Holland: Kluwer Academic Publisher.
Merleau-Ponty, M. (1963). *The structure of behavior*. Pittsburgh, PA: Duquesne University Press.
Miłkowski, M. (2013). *Explaining the computational mind*. Cambridge, MA: MIT Press.
Noë, A. (2009). *Out of our heads*. New York: Hill and Wang.
Piccinini, G. (2015). *Physical computation*. Oxford: OUP.
Rosen, R. (2000). *Essays on life itself*. New York: Columbia University Press.
Ruiz-Mirazo, K., & Moreno, A. (2004). Basic autonomy as a fundamental step in the synthesis of life. *Artificial life*, *10*(3), 235–259.
Stewart, J., Gapenne, O., & di Paolo, E. A. (Eds.). (2010). *Enaction: Towards a new paradigm for cognitive science*. Cambridge MA: MIT Press.
Thompson, E. (2005). Sensorimotor subjectivity and the enactive approach to experience. *Phenomenology and the Cognitive Sciences*, *4*, 407–427.
Thompson, E. (2007). *Mind in life: Biology, phenomenology, and the sciences of mind*. Cambridge, MA: Harvard University Press.
Thompson, E., & Stapleton, M. (2009). Making sense of sense-making: Reflections on enactive and extended mind theories. *Topoi*, *28*, 23–30.
Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. In *Proceedings of the London Mathematical Society, Series 2, 42,* 230-265; 43, pp. 544-546.
Varela, F. (1979). *Principles of biological autonomy*. New York: Elsevier.
Varela, F., Thompson, E., & Rosch, E. (1991). *The embodied mind*. Cambridge, MA: MIT Press.
Villalobos, M., & Ward, D. (2015). Living systems: Autopoiesis, autonomy and enaction. *Philosophy and Technology*, *28*(2), 225–239.
Villalobos, M. (2015). *The Biological Roots of Cognition and the Social Origins of Mind*. Ph.D. Thesis, University of Edinburgh.
Virgo, N., Egbert, M. D., & Froese, T. (2011). Advances in artificial life: Darwin meets von Neumann. In Kampis G, I. Karsai, & E. Szathmáry (Eds.), *The role of spatial boundaries in autopoiesis* (pp. 240–247). Berlin: Heidelberg: Springer-Verlag.
Wallace, B., Ross, A., Davies, J., & Anderson, T. (Eds.). (2007). *The mind, the body, and the world: psychology after cognitivism?*. Exeter: Imprint Academic.
Weber, A., & Varela, F. (2002). Life after Kant: Natural purposes and the autopoietic foundations of biological individuality. *Phenomenology and the Cognitive Sciences*, *1*, 97–125.
Weiner, N. (1948). *Cybernetics*. Cambridge, MA: MIT Press.
Wells, A. J. (1998). Turing's analysis of computation and theories of cognitive architecture. *Cognition*, *22*(3), 269–294.