



Computing Mechanisms Without Proper Functions

Joe Dewhurst¹ 

Received: 30 January 2018 / Accepted: 17 August 2018 / Published online: 27 August 2018
© The Author(s) 2018

Abstract

The aim of this paper is to begin developing a version of Gualtiero Piccinini’s mechanistic account of computation that does not need to appeal to any notion of proper (or teleological) functions. The motivation for doing so is a general concern about the role played by proper functions in Piccinini’s account, which will be evaluated in the first part of the paper. I will then propose a potential alternative approach, where computing mechanisms are understood in terms of Carl Craver’s perspectival account of mechanistic functions. According to this approach, the mechanistic function of ‘performing a computation’ can only be attributed relative to an explanatory perspective, but such attributions are nonetheless constrained by the underlying physical structure of the system in question, thus avoiding unlimited pancomputationalism. If successful, this approach would carry with it fewer controversial assumptions than Piccinini’s original account, which requires a robust understanding of proper functions. Insofar as there are outstanding concerns about the status of proper functions, this approach would therefore be more generally acceptable.

Keywords Computation · Mechanistic explanation · Proper functions · Perspectivalism

In Sect. 1 I review Piccinini’s mechanistic account of computation, and explain the apparent need for a notion of proper functions to make this account work. I then consider Piccinini’s own ‘objective goal’ account of proper functions, and raise one potential problem with this account, suggesting that it might be too early to say how successful the account will be. In Sect. 2 I develop a version of the mechanistic account that does not require any notion of proper functions, instead describing computing mechanisms in terms of perspectival functions. Finally, in Sect. 3 I address an obvious concern about the risk of pancomputationalism and triviality, by arguing that perspectival attributions of computational functions will be constrained by the physical structure of a mechanism, and thus will not entail unlimited pancomputationalism. Once we

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

¹ University of Edinburgh, Edinburgh, UK

have adopted an explanatory perspective there will be a fact of the matter about which parts of a system will qualify as computational components, based on the requirement that they possess a relevantly similar physical structure to one another. This raises the question of how to determine which physical structures count as relevantly similar, and I briefly consider two options for how to do this, one based on Isaac's (2013) definition of similarity in terms of 'homomorphisms induced by causal processes', and the other based on Millhouse's (2018) proposal for a simplicity criterion on computational mappings. Neither option is fully satisfactory, but both offer some potential routes for future investigation into how underlying physical structures might constrain the perspectival attribution of computational functions. In any case, reaching agreement on what constitutes a relevantly similar physical structure should be more straightforward than agreeing on what constitutes a proper function, and so I conclude that we should adopt a mildly perspectival version of the mechanistic account of computation.

1 Computing Mechanisms with Proper Functions

According to Piccinini's mechanistic account of computation, a physical computer is a kind of mechanism whose function is to perform systematic transformations over medium-independent vehicles.¹ He describes several different versions of the account (see Piccinini 2015: chapter 7), but I will focus here just on digital computation, which will serve as a simple test case for the possibility of computing mechanisms without proper functions. At the end of Sect. 3 I will briefly consider how this approach might generalise to other kinds of computation, but a deeper exploration of this topic will have to wait for another day.

Piccinini's account of computation is grounded in the mechanistic approach to explanation, which has recently become popular in the philosophy of biology and cognitive science (see e.g. Machamer et al. 2000; Glennan 2002, 2017; Bechtel and Abrahamsen 2005; Craver 2007). According to this approach, an explanation of some phenomenon (such as the circulation of blood around the body) is given by a physical mechanism (such as the heart) whose activities (such as pumping) produce the phenomenon. Mechanisms are further understood to be compositional, such that they are composed of parts (components), which can themselves be considered sub-mechanisms that contribute to the overall production of the target phenomenon. A crucial point here is that a mechanism is only ever a mechanism *for some phenomenon*, i.e. it has the function of producing that phenomenon, and a mechanistic explanation can only be given once a target phenomenon has been identified. In the case of computation the function being performed is characterised by Piccinini as the systematic transformation of medium-independent vehicles (2015: 121), and the phenomenon of interest is just whatever application the system is being used for.²

¹ Milkowski (2013) and Fresco (2014) also give mechanistic accounts of computation, but I focus here on Piccinini's account as it is probably the most popular and well-developed. Many of the points made here would apply equally well to Milkowski's and Fresco's accounts, insofar as any mechanistic account of computation must say something about how we determine the function of a mechanism.

² Typically information processing, but there are other ways of characterising computational phenomena, such as adaptive control or mathematical calculation.

A digital computing mechanism consists of at least two kinds of components, digits and processors, and potentially several other kinds of components, including input devices, output devices, and memory units (cf. Piccinini 2015: chapter 11). Digits and processors are defined relative to one another, such that a processor is able to transform digits according to its specified function, and a digit can be recognised by the relevant processors and transformed accordingly. The other non-essential components are defined similarly: an input device transforms external stimuli into digits, an output device transforms digits into external actions, and a memory unit preserves a string of digits. The physical structure of these components is relevant only insofar as it allows them to fulfil these functions, hence making them “medium-independent”.

For example, consider a mechanism whose components include two kinds of digit, which we will call ‘0’ and ‘1’, and one kind of processor. The function of the processor is to take strings of two digits and transform them into strings of one digit. If both digits in the first string are 1, it produces a single digit 1. In all other cases it produces a single digit 0. The digits and the processor could be instantiated in many different physical forms (voltage levels and silicon chips, holes in punch cards and a punch card reader, and so on). All that matters is that the processor is structured such that it is correctly sensitive to the physical structure of the digits, and vice versa. We could describe the function of this processor according to the Table 1.

It is important to note that the digits in this example bear no intrinsic content. While it is conventional to interpret ‘1’ as ‘TRUE’ and ‘0’ as ‘FALSE’, making this processor perform the function ‘AND’, we could just as easily give them the opposite interpretation, under which the processor would perform ‘OR’ (see Dewhurst 2018 for further discussion). According to the mechanistic account it is possible to individuate computational states and processes, such as digits and processors, purely in terms of their physical structure along with some understanding of mechanistic functions.

If successful, this account would be able to provide us with an objective and non-semantic definition of computation. Such an account would tell us which systems compute and which systems do not compute, and could thus potentially serve as the basis for a computational theory of cognition (provided that the brain or nervous system turned out to have the correct kind of mechanistic structure). It would then be a further question whether computational systems, thus defined, were also semantic systems, and if so, what kind of semantic content they possess.

Table 1 A simple processor

String 1	String 2
0, 0	0
1, 0	0
0, 1	0
1, 1	1

Piccinini's account of computing mechanisms relies heavily on the notion of a proper function, i.e. a function that it is in some sense the *purpose* of a system to perform (exactly what this sense is remains to be seen). In order to individuate the components of a mechanism we need to be able to say what the function of that mechanism is, and correspondingly what the function of each component is. He is quite upfront about this requirement, defining a physical computing system as a functional mechanism whose teleological (i.e. proper) function is to perform computations (2015: 121). As such, he owes us some account of proper functions. He considers and rejects two such accounts (etiological and causal/perspectival), before presenting his own 'objective goal' account. I will briefly rehearse each of these accounts in turn, and then outline a potential issue for Piccinini's own account, which suggests that it might be too early to say whether this account will be successful.

One popular approach in philosophy of biology has been to ground our understanding of proper functions in the evolutionary history of the organism or system in question (see e.g. Millikan 1989; Neander 1991; Griffiths 1993; Godfrey-Smith 1994; Schwartz 2002; cf. Piccinini 2015: 101–103). Accounts of this kind determine the function of an organ or biological component by appealing to its historical contribution to the reproductive success of the organism. So the function of the heart is to pump blood, rather than to make a beating sound, because it is this activity that has historically contributed to reproductive success. A similar account can be extended to artefacts, either by appealing to the intentions of the evolved organism who created it, or by appealing to the function that this artefact has been used for in the past. Applied to computation, this account would say that a mechanism's function is to compute if performing computations has contributed to the reproductive success of an organism's ancestors, or to the historical usefulness of an artefactual mechanism.

The problem with selectionist/etiological accounts, according to Piccinini, is that they typically appeal to unknown causal histories, "making function attribution difficult or even impossible" (Piccinini 2015: 102). While it might be true that the computations performed by the brain have contributed to the past reproductive success of an organism, we have no direct way of learning about that contribution *today*, and it does not seem that scientists interested in the computational properties of the brain typically appeal to its evolutionary history to determine these properties (although there are of course exceptions, see e.g. Barrett 2012). Furthermore, these accounts run into well-known metaphysical issues concerning the causal impotency of an organism or artefact's history (cf. Davidson's swamp man 1987 and Dennett's two-bitser 1987). The kinds of mechanistic functions that Piccinini is interested in should depend entirely on synchronic causal structure, such that a computing mechanism created spontaneously should have precisely the same function as a physically identical computing mechanism with a complex causal history. Piccinini concludes that whilst the evolutionary history of an organism might play a useful heuristic role, it is unsuitable for fixing the proper function of a mechanism.

Piccinini briefly considers causal role accounts that do not appeal to the history of a mechanism (e.g. Cummins 1975; Craver 2001), but points out that according to such accounts "everything or almost everything (of sufficient complexity) ends up having functions" (Piccinini 2015: 103). Applied to his account of computing

mechanisms, this would result in something like the pancomputationalism or triviality that he precisely wants to avoid (see Piccinini 2015: chapter 4). Given the relative simplicity of basic computational operations, it is plausible that many physical systems could be construed as performing computations, rendering the notion of physical computation relatively uninteresting and/or non-explanatory.

In order to get around this problem, causal role accounts can appeal to the explanatory perspective of the relevant scientific community (see e.g. Hardcastle 1999; Craver 2013). According to perspectival accounts, the function we attribute to an organ such as the heart depends on our explanatory interests: if we are trying to explain the circulation of blood around the body, then the heart functions as a pump, but if we are trying to explain the synchrony of an infant's breathing with its mother's heartbeat, then the heart might function as a metronome. Applied to computation, this would mean that a mechanism would only have the function of computing in contexts where this function contributes to our explanation of some phenomenon. For Piccinini, this introduces an unacceptable level of observer-relativity into what is meant to be an objective account of computation. In the next section I will argue that the perspectival approach can in fact avoid most of Piccinini's concerns, and thus offers a potential alternative to proper functions.

Piccinini's own proposal is that we should define teleological functions as stable contributions to the objective goals of an organism (2015: 108; see also Maley and Piccinini 2017), where objective goals are to be understood in terms of the survival and inclusive fitness of the organism (Piccinini 2015: 106). In this way he hopes to ground teleological functions (pumping blood) in non-teleological truthmakers (the survival of the organism). Note that unlike the selectionist/etiological account there is no appeal to evolutionary history here: it does not matter why or how the heart evolved, but only that it currently contributes to the survival of the organism. The account is also able to accommodate artefactual functions, which are simply understood as contributing to an objective goal of the organism that created the artefact (*ibid*: 111). There are some additional subtleties to the account that I will not consider here (see Piccinini 2015: 109–117), but it seems able to avoid many of the problems associated with the other accounts that Piccinini considers.

A computing mechanism, according to this account, would be a mechanism whose function is to compute, where this function is understood as contributing to the survival or inclusive fitness of an organism (either directly or via a created artefact). So the electronic computer that I am typing on has the function of computing because it has been designed to perform this function, and this function (at least indirectly) contributes to the objective goals of its designers. If my brain can accurately be described as a computing mechanism, it is because it performs computations that contribute to my survival and inclusive fitness (insofar as the brain might enable perception, motor control, etc., this seems plausible). If it were successful then this account would be able to explain, in objective and non-historical terms, what it means for a mechanism to have the function of performing computations.

One potential issue with this account is that it is not clear how we are supposed to understand the contribution of a mechanism to an organism's survival and inclusive fitness without first understanding the structure of that mechanism, which we can only identify once we know what function it performs. The proper function of a

mechanism is understood in terms of whatever contribution it makes to the objective goals of an organism, but we can only identify that contribution if we already know what the mechanism does. So it looks like some independent means of identifying mechanistic structures is required *before* we can identify what the proper function of a mechanism is. The contribution a mechanism makes to survival or inclusive fitness does not provide any independent explanatory power over and above the causal structure of that mechanism. Once we understand the causal structure of a mechanism we can come up with a description of the contribution it might make to survival and inclusive fitness, but what we cannot do (on pain of circularity) is appeal to that description in order to explain the causal structure of the mechanism itself.³

Consider the heart. According to Piccinini's account, the proper function of the heart is to pump blood, as pumping blood contributes to an objective goal (survival) of the organism. However, we can only identify this contribution once we understand the mechanistic structure of the heart, i.e. once we understand that it functions as a pump. Prior to having at least some understanding of its mechanistic structure, we would have no basis for attributing this function. But to understand this mechanistic structure, according to Piccinini's account, we would already need to have identified its proper function, which is only possible once we have determined the contribution that it makes to the objective goals of the organism. So we seem to be stuck in a sort of dilemma: either we can independently identify the mechanistic structure of a system, in which case there seems to be no need for the attribution of proper functions, or we cannot identify the mechanistic structure without first appealing to proper functions (and hence, objective goals), which leads us into circularity.

It's possible that it in some cases this circularity could be avoided by first coming up with a rough approximation of what the system does, prior to developing a full understanding of its mechanistic structure. This would be what Piccinini and Craver (2011) describe as a 'mechanism sketch', which they argue applies to all forms of functional analysis, and could allow us to bootstrap our way to a full mechanistic explanation.⁴ In the case of the heart it could plausibly be argued that we already have at least a rough idea of what contribution the organ makes, just by observing its surface behaviour and relationship to the rest of the cardiovascular system. This might be sufficient to get the account off the ground, at least for systems that have a relatively transparent functional structure.

However, the problem seems to be worse for computational structures, which can only be individuated once we know what role they are supposed to play. This

³ This problem of circularity was first raised by Dewhurst (2016), and is similar to one that Piccinini himself describes when criticising inferential role accounts of semantic content (2015: 33–34). I take it that the kind of circularity at stake here is inferential, in the sense discussed by Humberstone (1997) and Burgess (2007). Dewhurst (2016) also suggests that Piccinini's account of proper functions might be self-undermining, insofar as if it were successful then it could be used to ground a semantic account of computation based on an 'objective role' version of teleosemantics. Of course, this is not a reason to reject the account, but rather a reason to think that Piccinini might be better off avoiding proper functions altogether.

⁴ A similar proposal is made in the case of evolutionary explanations by Hull (1967), and discussed by Walton (1985) as a way of potentially redeeming (some) circular explanations or arguments.

is because the functional characterisation of a computing mechanism is highly abstract and thinly specified, such that almost any physical system can potentially be *described* as performing some computation (I will return to this issue in Sect. 3). Unlike the relatively transparent way in which the heart might be described as functioning as a pump, there is no obvious sense in which a surface level functional analysis will tell us whether or not a system is computing, or what computation it is performing. It is only when we delve deeper and investigate the fine-grained mechanistic structure of a system that we can begin to get a sense of the kind of computations it might be performing. Prior to doing this we will have no clear way of determining what contribution, if any, a computing mechanism makes to the objective goals of an organism. So if mechanistic analysis is only possible once the proper function of a system has been fixed, and if proper functions can only be fixed once we understand the contribution a system makes to survival or inclusive fitness, it seems like it might be impossible to break out of the circularity described above (at least in the case of computation).

Piccinini's account of proper functions is relatively new, and it is as yet unclear how successful it will be. Even if the problem I have described above can be overcome, it is likely that others will arise. As such it currently represents a potential weakness in his overall account of physical computation: if his account of proper functions is not successful, then his mechanistic account of computation may also be unsuccessful, depending on whether or not there is any other way to preserve the core features of the account. As such, my aim in the rest of this paper is to develop a version of the mechanistic account that does not rely on any notion of proper functions, and could thus withstand the failure of Piccinini's proposed 'objective goal' account. I will do this by rehabilitating a version of the perspectival story about mechanistic functions, which he considers only briefly before rejecting.

2 Computing Mechanisms Without Proper Functions

Piccinini briefly considers and rejects accounts of mechanistic functions that appeal to explanatory perspectives. Craver (2013; cf. Craver 2001) gives one such account, arguing that whilst functional descriptions of mechanisms are "ineliminably perspectival" (*ibid.*: 133), they are perspectival in a sense that does not threaten the objectivity of mechanistic explanations. Craver's account builds on Cummins' (1975) causal role account of functions, according to which any causal process can be described as functional. In order to avoid having to say that everything has a function, Craver introduces the notion of an explanatory perspective from which functional attributions are made, and which constrains functional attributions to those which are of interest from that perspective. In the context of mechanistic explanation, an explanatory perspective is already adopted in order to determine the target phenomenon, and it follows naturally from this that functional attributions might also be made from the same perspective, constraining the attribution of functions to those that are suitable for explaining the production of the target phenomenon. My aim in the rest of this paper is to use this account in order to propose a way of thinking about computing mechanisms with perspectival functions that does not thereby render those

mechanisms completely observer relative. If successful, this account would be able to identify and individuate computing mechanisms without appealing to any notion of proper functions, while nonetheless avoiding unlimited pancomputationalism (see Sect. 3).

Craver distinguishes between three kinds of functional attribution: “as a way of tersely indicating an etiological explanation, as a way of framing constitutive explanations, and as a way of explaining the item by situating it within higher-level mechanisms” (2013: 133). I will first describe each kind of functional attribution before explaining why Craver considers them all to be perspectival and why he does not think that perspectivalism of this kind threatens scientific objectivity. I will then apply this account of perspectival functions to Piccinini’s mechanistic account of computation.

2.1 Etiological Functions

An etiological function is one defined in terms of the history of a system or mechanism, typically its evolutionary history in the case of biological systems (cf. Craver 2013: 145–149). As Craver describes it, this usually means appealing to the adaptive value of the mechanism, such that its function is whatever it was selected for in previous generations. The heart’s etiological function is to pump blood, for example, because it is this activity that contributed to the survival and reproductive success of the organism’s ancestors. In contrast, making a thump–thump sound did not contribute to survival and reproductive success, and so is not the heart’s etiological function.

Recall that Piccinini dismisses etiological accounts for being excessively speculative (i.e., we can only guess at what historical contribution to adaptive success a mechanism may have made), and also for rendering our functional attributions causally impotent (i.e. that the heart’s function *historically* may have been to pump blood does not explain why its function *today* is to pump blood). Craver raises similar concerns, but also makes the additional point that etiological functions seem to be burdened with normative implications that he thinks have no place in a fully mechanistic biology or cognitive science (2013: 148), i.e. they carry with them the implication that a mechanism *ought* to function in a certain way, due to its evolutionary history.⁵ He argues that while the attribution of etiological functions “can be heuristically useful as a guide to creative thinking about what an organism or organ is doing” (*ibid*), they do not identify anything intrinsic to the structure of a mechanism, but are rather a feature of our explanatory perspective.

⁵ It’s possible that Craver conflates two senses of normativity here, i.e. normativity in the biological or evolutionary sense and normativity in the sense of something we ‘ought’ to do. Given that Piccinini also dismisses etiological account of functions for independent reasons, nothing much rests on this point for my purposes.

2.2 Constitutive Functions

An attribution of a constitutive function is a description of the synchronic causal structure of a mechanism (cf. Craver 2013: 149–151). For example, one might describe the constitutive function of the heart in terms of the physical activity of constricting and releasing which serves to push blood around the body. Attributions of constitutive functions are perspectival in the sense that there are many distinct ways in which to characterise the physical activities of a system, none of which is obviously privileged outside of an explanatory context. So when we are trying to explain the pumping of blood, it makes sense to attribute to the heart the constitutive function described above, but if we were trying to explain an avant-garde musical performance, it might sense to instead describe how the motion of the heart creates a rhythmic beat. Without first adopting one of these explanatory perspectives, there is no sense in which one of these constitutive functions is more fundamental than the other.

2.3 Contextual Functions

Finally, a contextual function situates the activity of a mechanism within a broader system (cf. Craver 2013: 151–154). For example, it only makes sense to say that the heart's function is to pump blood when that heart is situated within a broader system composed of veins, arteries, etc. Without this system the heart would not be pumping blood, it would just be expanding and contracting. Given that there are many different contexts within which a mechanism is simultaneously embedded, it once again does not make sense to say that any particular contextual function is *the* function of a mechanism. Prior to adopting an explanatory perspective, we cannot privilege any one context over another.

Craver concludes that all three kinds of functional attribution are in a sense perspectival, but he does not think that this should give us much cause for concern. The reason for this is that Craver thinks a mild form of perspectivalism is just an ineliminable feature of scientific practice (at least in the life sciences), insofar as we must always make choices about which aspects of a system to focus on when giving an explanation (Craver 2013: 155). This is part and parcel of the mechanistic worldview, which emphasises the important explanatory role played by the identification and characterisation of an explanandum phenomenon (cf. Shagrir and Bechtel 2017), and accepts that this must be done from within our own value-laden and epistemically-limited perspective (simply because we have no other perspective available to us). Nonetheless, a mild perspectivalism of this kind does not entail that 'anything goes', and should be contrasted with a strong perspectivalism or epistemic relativism where any explanation or functional attribution is as good as any other (for further discussion see Baghrarian and Carter 2017, especially sec. 4.4.3). A key point here is that in all three cases (etiological, constitutive, and contextual) our attributions of mechanistic functions are still constrained by the physical structure of the system, and once we adopt an explanatory perspective, there will typically be a clear fact of the matter about what the function of a mechanism is. So while these

attributions are perspectival, they are not arbitrary, and from within a perspective we should usually be able to agree on what the function of a mechanism is.

Piccinini himself concedes that mechanistic explanations may be perspectival in a harmless sense (2015: 142), but dismisses Craver's account of perspectival functions for introducing observer-relativity into otherwise objective scientific practice. However, I think he is too quick to dismiss the possibility of a perspectival account of functions, which I will now argue can serve as an adequate foundation for an objective (enough) account of computing mechanisms. In the case of computing mechanisms, it seems appropriate to focus on constitutive functions, which most closely match the way that computational systems are usually described, i.e. in terms of mappings from inputs to outputs (Craver also refers to constitutive functions as 'input–output' functions). As Piccinini argues, individuating computational states in terms of their etiological functions fails to reflect our interest in synchronic computational structures. One desideratum of his account is that two physically identical computing mechanisms should qualify as performing the same (non-semantic) computations, regardless of how they have been used historically. The same kind of concern rules out individuation in terms of (wide) contextual functions,⁶ which would mean that two physically identical computing mechanisms might turn out to be performing distinct computations, depending on the wider context that they find themselves in (see Dewhurst 2014, 2018 for further discussion; cf. Shagrir 2001). Piccinini does retain a role for "the interaction between mechanisms and their contexts" (2015: 139) in fixing functional attributions, but such a role will inevitably introduce a perspective of sorts (i.e. that of the observer classifying this interaction), and so will be compatible with my general approach. I will focus here on constitutive functions, and try to demonstrate how to apply a perspectival version of constitutive functional attribution to Piccinini's mechanistic account of computation.

As I described earlier, Piccinini's mechanistic account of computation requires some notion of function in order to individuate computational states and processes. We need to be able to say that some given physical configuration *functions* as a digit or *functions* as a processor, where this function is understood in constitutive terms, i.e. as exhibiting the right kind of causal interactions with other digits and processors. Piccinini tries to ground these functions in the objective goals (i.e. survival and/or inclusive fitness) of an organism, but I want to avoid making the account reliant on any notion of proper functions. How, then, might a perspectival account of mechanistic functions go about individuating computational states and processes?

We can stipulate that the constitutive functions we are interested in are those that correspond to computational states and processes, as defined according to Piccinini's account of computing mechanisms. According to this account, a digit is a physical component that interacts with another physical component, a processor, to produce more digits in a systematic manner. So the parts of a physical

⁶ Piccinini does allow that the *internal* context of a computational component might play some role in its individuation, insofar as components (such as digits and processors) are defined in relation to one another.

system that interact in this way can be said to have the constitutive functions of a digit, a processor, etc. The larger physical system composed of these parts has the constitutive function of computing, according to this description. Nothing about the description itself is intrinsic to the system, but the description can only be applied to systems that possess the correct physical structure, thus constraining the class of computational descriptions that can be applied to any given system. This means that while our individuation of computational states and processes must be made from within an explanatory perspective, there are nonetheless constraints on how we go about performing that individuation once we have adopted an explanatory perspective. In the rest of this section I will say more about these constraints, and in the next section I will respond to the concern that they may not be sufficient to rule out pancomputationalism or triviality.

Assume that we are interested in determining whether a system performs the computation captured by Table 1. We can see that in order to perform this computation, a system must possess at least three kinds of component: a digit corresponding to ‘0’, a digit corresponding to ‘1’, and a processor that systemically performs the correct transformations on strings of these digits. Therefore, we must be able to identify physical components in the system corresponding to these two kinds of digit and one kind of processor, where ‘corresponding’ simply means that we can identify physical structures that are able to play the role of those components. There are many different physical structures that could correspond to these components: all that is required is that the processor is correctly sensitive to the digits, such that it is able to distinguish them from one another, and transform them in the manner described by the table.

Table 1 (repeated) A simple processor

String 1	String 2
0, 0	0
1, 0	0
0, 1	0
1, 1	1

One candidate system might consist of a silicon chip (A) that is sensitive to two different voltage levels, 0 V and 5 V. In the most straightforward case, the wiring connected to this chip only ever carries pulses of 0 V or 5 V, making it simple to interpret the system in a way that matches up with our desired transformation (see Table 2). However, it could also be the case that the wire carries strings consisting of

Table 2 The transformations performed by chip A

Input	Output
0 V, 0 V	0 V
5 V, 0 V	0 V
0 V, 5 V	0 V
5 V, 5 V	5 V

Table 3 The transformations performed by chip B

Input	Output
0–2.5 V, 0–2.5 V	0–2.5 V
5 V, 0–2.5 V	0–2.5 V
0–2.5 V, 5 V	0–2.5 V
5 V, 5 V	5 V

Table 4 Another simple processor

String 1	String 2
1, 1	1
0, 1	1
1, 0	1
0, 0	0

other voltage levels, which the chip is not systematically sensitive to.⁷ In this case we would just ignore the additional voltage levels, which, relative to our explanatory perspective and the chip (i.e. processor) in question, would not constitute computational components.

Take this same wiring and connect it up to a second chip (B), however, and those additional voltage levels could constitute computational components. Chip B is sensitive not only to 0 V and 5 V, but also to 2.5 V, which it treats identically to 0 V (see Table 3). This chip would continue to perform the same basic computation operations as those performed by chip A (presented in Table 1), despite being sensitive to an additional voltage level, although this additional sensitivity would mean that its outputs could be distinct (depending on what inputs each system receives). A third kind of chip might be sensitive to all three voltage levels, but treat them each as a distinct kind of digit, resulting in a distinct set of computational individuations (for further discussion see Shagrir 2001; Dewhurst 2018).

There are many further permutations of physical computing mechanisms that could be described in this manner. The important point is that once we have adopted this explanatory perspective, there will be a limited (although still wide) range of interpretations that each of these physical systems can be given. We could give each kind of digit the opposite interpretation, mapping both chips to the operation described by Table 4, rather than Table 1, but we could not (for example) count

⁷ It is true the chip is likely to react in some way to all kinds of irrelevant stimuli, including other voltage levels, being dropped on the floor, or being immersed in water (reactions in these cases might include overheating and eventually melting, breaking apart physically, or short-circuiting). These reactions are either going to be unsystematic, in which case they cannot appropriately be interpreted as components in a computing mechanism, or else systematic enough that they could plausibly be interpreted as computational components. In the latter case it would be quite possible to adopt a perspective from which, say, heat transference from an overheating processor does constitute a computational output, i.e. a digit of sorts, although this might not be an especially helpful perspective to adopt. This concession may raise some concerns about triviality and/or pancomputationalism, which I respond to in the next section.

2.5–5 V as a single kind of digit in the case of chip B, as the physical structure of the chip is not consistently sensitive to this range of voltage levels (i.e. it does not treat voltage levels in this range in a consistent manner).

In this section I have presented Craver's (2013) account of mechanisms with perspectival functions, and argued that this approach is suitable for the attribution of functions to computing mechanisms. According to this perspectival account of computational functions, what it means for a mechanism to perform the function of computing is to possess the correct kind of physical structure to be interpreted as performing this function from an explanatory perspective. A perspectival approach of this kind raises obvious concerns about pancomputationalism and triviality, as it is notoriously easy to interpret any physical system as performing some (or perhaps any) computation, such that it will turn out that every physical system is a computer. In the next section I will respond to this concern by explaining in more detail how the physical structure of a system might constrain the range of computational functions that we can attribute to it, thus avoiding unlimited pancomputationalism.

3 Pancomputationalism, Observer-Relativity, and Triviality

According to the strategy outlined in the previous section, any given computational system can be interpreted as performing a number of distinct computations, depending on one's explanatory perspective. Coupled with the ease with which physical systems can be interpreted as performing computations, this might raise familiar concerns about pancomputationalism, triviality, or observer relativity. Piccinini (2015: chapter 4) distinguishes between several different varieties of pancomputationalism, ranging from the strongest version, where "every physical system performs every computation" (*ibid*: 51), to weaker versions where every physical system performs one (or a few) computations (*ibid*: 52). He also considers different sources of pancomputationalism, including 'interpretivist', 'causal', and 'information-based' (*ibid*). The perspectival account advocated here is most vulnerable to a combination of interpretivist and causal pancomputationalism, which might be either strong or weak, depending on how the details are cashed out. In the rest of this section I will argue that we should bite the bullet and accept a limited version of pancomputationalism, where many (or perhaps any) physical systems can be interpreted as performing some computation *relative to an explanatory perspective*. However, it will turn out that once we have adopted an explanatory perspective, the range of functional attributions that we can make is constrained by the underlying physical structure of the system that we are interested in, thus avoiding completely unlimited pancomputationalism. One consequence of this approach is that we can no longer appeal to the computational status of a system in order to determine whether or not it is cognitive, thus ruling out an exclusively computational theory of mind, although this is not to say that computation could not make *some* contribution to our understanding of mind or cognition.

There is one additional variety of pancomputationalism, which is the idea that the entire universe might be a computer. According to this view, the fundamental nature of the universe is computational, and all physical processes are the outcome of underlying

computational processes.⁸ It is an important proposal, with many interesting implications,⁹ but these are ultimately orthogonal to the current discussion. If the whole universe is a computer, then the question of whether any particular system is computational becomes moot, or alternatively, we would need to redefine what we mean by ‘computational’ in order to salvage an interesting sense of the term. One way of doing this would be to adopt the perspectival account of computational functions that I presented in the previous section: even if the whole universe *was* a computer, it might not always be useful or interesting to describe it in computational terms, and so we will need some criteria to determine when a computational *description* of a system is most appropriate. The best way to do this, I claim, is in terms of explanatory perspectives. Moving forward I will focus on the question of how to define the relationship between a perspective and a system such that we can distinguish between more or less appropriate attributions of computational functions, and thus avoid completely unlimited pancomputationalism.

The perspectival account of computational functions introduced in the previous section is obviously observer relative to some extent, insofar as it says that the function of ‘computing’ (and mechanistic functions more generally) are always attributed to a system from an explanatory perspective. This should not be seen as a negative feature of the account, but is rather a consequence of the way in which we characterise computational processes such that any given system can be seen as performing multiple computations simultaneously. Each system can nonetheless be given a physical description (i.e. in terms of voltage levels), which provides a strong limitation on the range of computations it can be described as performing.¹⁰ This rules out of the kind of radical observer-relativity that Piccinini is concerned with, for so long as there is a fact of the matter about the physical interactions going on in a system, there will be only a limited range of computations that we can legitimately interpret it as performing.¹¹

Most arguments for unlimited pancomputationalism involve arbitrary mappings between a computational formalism and the physical structure of a (supposedly)

⁸ Piccinini (2015: 56–60) calls this view “ontic pancomputationalism”, and historical proponents include Zuse (1970, 1982), Feynman (1982), Toffoli (1982), Wheeler (1982), and Wolfram (2002). Piccinini offers some empirical and metaphysical objections to the view, but I will not take a stand on it either way in the current article.

⁹ See e.g. Pexton’s (2015) article in this journal, which argues for a kind of ontological emergence based on ontic pancomputationalism. Pexton draws on the same idea of informational compression in terms of Kolmogorov complexity that I will later suggest can help constrain attributions of computational functions, but his argument otherwise has little to do with the current topic, as it assumes a radical version of pancomputationalism that I think renders the question of computational identity/individuation relatively moot—I discuss this point in more detail in the main text.

¹⁰ Coelho Mollo (2017) has recently argued that this method of computational individuation forces us “to give up any useful notion of computational equivalence”. His proposed solution, however, is reliant on teleological functions, and as such would not be suitable for an account that aimed to do without any notion of proper function. A full response would not be appropriate here, but in brief I think that it will be possible to fix a level of physical description where, at least relative to explanatory interests, we are able to identify the required computational equivalences.

¹¹ Fresco (2015) defends a similar (although less permissive) position, according to which multiple *semantic* interpretations can be made of a single computational system, while nonetheless remaining constrained by the physical structure of that system.

computational system. For example, Searle's classic argument for pancomputationism involves identifying an isomorphism between arbitrary molecular movements in a physical system (a wall) and the formal structure of the Wordstar program¹² (see Searle 1992: 208–209; cf. Putnam 1988: 121–125 for a more formal version of this argument), without having to say anything at all about the mechanistic structure of the wall itself. My account is able to avoid having to allow computational mappings of this kind, as we will not typically be able to identify stable physical configurations corresponding to digits and processors in (intuitively) non-computational systems such as a wall or a bucket of water. A processor must be able to systematically identify and transform discrete physical configurations (i.e. digits), which cannot happen in an unstructured physical system. Within any given computing mechanism, every type-identical digit or processor must possess a relevantly similar physical structure, which rules out the kind of post hoc or retroactive mappings that arguments for pan-computationism typically rely on.

More will need to be said in the future about how exactly to spell out this notion of 'relevantly similar physical structure', but one option might be to adopt a version of the account presented by Isaac (2013), who argues for a notion of similarity based on 'homomorphisms induced by causal processes'. The idea here is that in order to qualify as 'similar' in an objective sense, two structures must not only be homomorphic to some degree, but must also bear this homomorphism as the result of a causal process, either from one to the other, or from some common ancestor. While Isaac uses the account to elucidate the use of mental representations in psychological science, it could be repurposed to give a more general analysis of structural similarity that makes no assumptions about representation or semantic content. In the case of physical computing mechanisms, this would allow us to individuate components (such as digits and processors) according to their physical structure, where two discrete components are counted as the same component-type only if they are structurally homomorphic in the relevant sense *and* this homomorphism can be traced back to a common cause (such as the designer of an artificial system, or evolutionary mechanisms in the case of natural computation). The immediate benefit of this approach is that it would rule out gerrymandered or ad hoc mechanisms like those described by Putnam and Searle, but a downside is that it begins to resemble the etiological accounts of proper functions that I criticised in Sect. 1, as it relies on some access to the causal history of a system. As such, it is not yet clear that this approach is suitable for present purposes, although it has some promise and could be developed further in future work.

Another possible option would be to adopt the 'simplicity criterion' recently proposed by Millhouse (2018), according to which we can measure the relative (Kolmogorov) complexity of a computational mapping and thus determine to what extent a proposed implementation is tracking 'real patterns' in the underlying physical structure.¹³ The idea is that a computational mapping could be considered more

¹² An early word processing program.

¹³ Fresco (2015: 1037) makes a similar suggestion, and Ladyman and Ross (2007) originally made the connection between Dennettian real patterns and informational complexity.

robust (and less arbitrary) to the extent that it provides a more compressed description of the dynamics of the physical structure. So a ‘good’ mapping will provide a more compressed description, whereas a ‘bad’ mapping (such as Searle’s Wordstar wall) will require a very lengthy, uncompressed description. There are some further details that I am not able to go into here, but this approach offers a promising way of thinking about the relationship between a computational formalism and its physical implementation.

Millhouse concedes that his criterion would get us only an ordering of which mappings are more or less ‘simple’, rather than a conclusive answer to the question of whether or not a system implements a computation, but according to the perspectival approach I am endorsing here this should not concern us too much. We can concede that it might be possible to map any computational formalism to any physical structure without thereby denying that some mappings (or functional attributions) are more useful than others. What Millhouse’s criterion give us is an objective measure of how well a computational function describes an underlying physical structure (in terms of the informational complexity of that description), which could also allow us to compare the structures directly, in terms of which computational functions can be most naturally implemented upon them (cf. Millhouse 2018: 16–17).

However we do it, once we have identified a sense of ‘relevantly similar physical structure’ that most agree upon, we will be able to assess different attributions of computational functions to structures in a relatively non-perspectival manner.¹⁴ While it will remain the case that a given attribution might be more or less valuable relative to our different explanatory perspectives, we will nonetheless be able to agree upon the value of that attribution *relative to the perspective in question*. Millhouse’s (2018) proposal is once again valuable here, as we can understand the notion of an ‘explanatory perspective’ (*qua* computation, at least) in terms of the perspective from which an interpretation is made (*ibid*: 12). So even if it is the case that stable physical configurations to which we can attribute computational functions are quite common, there will still be a fact of the matter about which of these attributions is of more value (in terms of information compression) from a given explanatory perspective.¹⁵ In some cases the computational description of a given

¹⁴ I have suggested two possible ways of understanding ‘relevantly similar physical structure’ here, both of which have their downsides. Possible alternative approaches include Eva et al.’s (forthcoming) assessment of the similarity of causal structure in evidential and counterfactual terms, and Schiller’s (2018) proposal for assessing the similarity of computational structures in terms of a ‘swapping constraint’. The latter could be particularly appropriate for present purposes, but as it was not yet published at the time of writing I have not been able to consider it in any detail here.

¹⁵ In an earlier draft I suggested that such stable configurations might be quite uncommon, but as an anonymous reviewer pointed out, there are many kinds of common physical interaction that can easily be interpreted in computational terms, such as two lanes of traffic merging in a systematic manner, or a wall undergoing gradual collapse as a result of stable geological processes. In the latter example there are unlikely to be discrete parts of the system that are similar enough to be treated as instances of the same computational component type, but in the former example it is plausible that there could be, in which case I would be willing to accept that there might be a perspective from which this constitutes a form of computation. Nonetheless, some systems will still be more amenable to computational description than others, a feature that Millhouse’s account manages to capture well.

system may even be less compressible than a simple physical description, suggesting that the attribution of a computational function to this system, while possible, is explanatorily superfluous (see Millhouse 2018: sec. 3.1).¹⁶

A perspectival account of computing mechanisms will nonetheless involve accepting a more limited form of pancomputationalism. It is the case that for any given physical system, there is likely to be an explanatory perspective from which some of the physical structures within that system can be interpreted as the components of a computing mechanism. This would seem to suggest that, according to the perspectival account, every physical system performs at least some computation, resulting in limited pancomputationalism. I do not think that the perspectivalist about computational functions should try to deny this. However, as Schweizer (2014, 2016) has recently argued, pancomputationalism of this kind should not concern us, provided that our aim is simply to describe how computational explanation (in cognitive science and elsewhere) can proceed. If our aim was to provide an account of which systems are cognitive, as classical computationalism has attempted to do, then this limited form of pancomputationalism would be more concerning, as it would turn out that there is nothing unique about the computational capacities of cognitive systems. I take it that adopting the perspectival account would simply give us reason to avoid a computational criterion for cognition, and to look elsewhere for a ‘mark of the cognitive’ (if indeed there is one to be found). This is not to say that cognitive systems are not computational—they may well be—but rather that we should not define them as cognitive simply in virtue of being computational.

So, by adopting a perspectival account of mechanistic functions, it is possible to develop a version of Piccinini’s mechanistic account of computation that makes no reference to proper functions, but is nonetheless capable of non-arbitrarily individuating computational states and processes. Doing so involves adopting an explanatory perspective, but perspectivalism of this kind is perfectly innocent and widespread across the biological sciences. While such an account is not totally objective, as it requires us to acknowledge the perspective of an observer, it does not collapse into total observer-relativity in a way that would render the notion of computation trivial. A perspectival account of mechanistic functions is in this sense no worse off than perspectival accounts in any other scientific discipline (see e.g. Giere 2006 for a general account of scientific perspectivalism, and Massimi 2016 for a recent analysis of different kinds of perspectival realism).

I have focused here on defending a perspectival account of *digital* computing mechanisms, and there is an additional question of whether and to what extent this account will generalise to other kinds of computation, such as analog computation (Piccinini 2015: chapter 12), neural computation (Piccinini and Bahar 2012),¹⁷ and

¹⁶ One could even adopt this as a definition of computation if so desired, ruling out cases where the computational description is less compressible than the physical description, although this measure will still allow many more computational systems than are usually accepted.

¹⁷ Neural computation may or may not be a species of analog computation—see Maley (2018) for further discussion.

‘unconventional computing’ understood more broadly (see e.g. Adamatzky 2015).¹⁸ One concern here might be that, once we allow in these other kinds of computation, the physical constraints described above will no longer be sufficient to prevent unlimited pancomputationalism. For example, analog computations are defined over continuous variables rather than discrete digits, rendering problematic the ‘stable physical configuration’ constraint on computational attributions. This might mean that any physical interaction can be trivially interpreted as performing any analog computation, reintroducing unlimited pancomputationalism. One obvious option here would be to distinguish between the different kinds of computation, such that digital computations are constrained in the ways suggested above, and other kinds of computation are either constrained in some other way, or unconstrained such that they allow for unlimited pancomputationalism. We could subsequently distinguish different kinds of pancomputationalism (digital, analog, etc.), and we might at least be able to avoid unlimited *digital* pancomputationalism. This would be sufficient for my current purposes, although in future work I would like to explore the possibility of physical constraints on the perspectival attribution of computational functions of other (i.e. non-digital) kinds.

4 Conclusion

In Sect. 1 I introduced Piccinini’s mechanistic account of digital computation and described the apparent need for some notion of proper functions in order to make this account work. I raised one concern with Piccinini’s own preferred account (his ‘objective goal’ account), suggesting that it might suffer from a form of inferential circularity between the attribution of proper functions and the identification of mechanistic structures, which both seem to rely upon one another. Even if this argument is not convincing, it would be preferable if the mechanistic account could avoid appealing to proper functions, in order to remove a potential weakness of the account. Given the relatively controversial status of proper functions in biology and cognitive science, it would be better if the account were not reliant on proper functions at all. As such, I have suggested that Craver’s account of mechanisms with perspectival functions might provide a suitable basis for a version of the mechanistic account that does not rely on proper functions at all. In Sect. 2 I proposed a way of thinking about computing mechanisms as having perspectival functions. Finally, in Sect. 3 I argued that this approach is able to avoid some obvious concerns about pancomputationalism and observer-relativity by appealing to the underlying physical structure of a mechanism used for a particularly explanatory purpose, which serves as a constraint on our attributions of computational functions. However, it will still be the case that any physical system can potentially be interpreted as performing *some* computation, resulting in limited pancomputationalism. This is a consequence of the perspectival approach that I think we should accept, although this is undoubtedly a controversial position. In future work I would like to explore in more

¹⁸ I thank an anonymous reviewer for bringing this important issue to my attention.

detail how a physical structure can constrain the range of possible computational interpretations, and why I do not think we should be concerned by the form of limited pancomputationalism implied by the perspectival approach presented here.

Acknowledgements Earlier versions of this paper benefited greatly from discussions with Alistair Isaac, Paul Schweizer, and Dimitri Coelho Mollo, as well as feedback from the audience at IACAP 2018. I am also grateful to two anonymous reviewers for this journal, who provided helpful comments and suggestions that allowed me to improve the paper significantly.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Adamatzky, A. (2015). Slime mould processors, logic gates and sensors. *Philosophical Transactions of the Royal Society A*, 373, 20140216.
- Baghramian, M., & Carter, J. A. (2017). Relativism. In Zalta (ed.), *The Stanford Encyclopedia of Philosophy*. <https://plato.stanford.edu/archives/sum2017/entries/relativism/>. Accessed 22 Aug 2018.
- Barrett, H. C. (2012). A hierarchical model of the evolution of human brain specializations. *Proceedings of the National Academy of Sciences*, 109, 10733–10740.
- Bechtel, W., & Abrahamsen, A. (2005). Explanation: A mechanistic alternative. *Studies in History and Philosophy of the Biological and Biomedical Sciences*, 36, 421–441.
- Burgess, J. A. (2007). When is Circularity in Definitions Benign? *The Philosophical Quarterly*, 58(231), 214–233.
- Coelho Mollo, D. (2017). Functional individuation, mechanistic implementation: The proper way of seeing the mechanistic view of concrete computation. *Synthese*. <https://doi.org/10.1007/s11229-017-1380-5>.
- Craver, C. (2001). Role functions, mechanisms and hierarchy. *Philosophy of Science*, 68, 31–55.
- Craver, C. (2007). *Explaining the brain*. Oxford: Clarendon Press.
- Craver, C. (2013). Functions and mechanisms: A perspectivalist account. In P. Huneman (Ed.), *Functions*. Dordrecht: Springer.
- Cummins, R. (1975). Functional Analysis. *Journal of Philosophy*, 72(20), 741–765.
- Davidson, D. (1987). Knowing One's Own Mind. *Proceedings and Addresses of the American Philosophical Association*, 60L, 441–458.
- Dennett, D. (1987). *The intentional stance*. Cambridge, MA: HUP.
- Dewhurst, J. (2014). Rejecting the received view. In *Proceedings of the 50th anniversary convention of the AISB*.
- Dewhurst, J. (2016). Review of physical computation. *Philosophical Psychology*, 29, 795–797.
- Dewhurst, J. (2018). Individuation Without Representation. *British Journal for the Philosophy of Science*, 69(1), 103–116.
- Eva, B., Stern, R., & Hartmann, S. (Forthcoming). The similarity of causal structure. *Philosophy of Science*.
- Feynman, R. P. (1982). Simulating Physics with Computers. *International Journal of Theoretical Physics*, 21(6-7), 467–488.
- Fresco, N. (2014). *Physical computation and cognitive science*. New York: Springer.
- Fresco, N. (2015). Objective Computation Versus Subjective Computation. *Erkenntnis*, 80(5), 1031–1053.
- Giere, R. (2006). *Scientific perspectivism*. Chicago: University of Chicago Press.
- Glennan, S. (2002). Rethinking Mechanistic Explanation. *Philosophy of Science*, 69, S342–S353.
- Glennan, S. (2017). *The new mechanical philosophy*. Oxford: OUP.
- Godfrey-Smith, P. (1994). A modern history theory of functions. *Noûs*, 28(3), 344–362.

- Griffiths, P. E. (1993). Functional Analysis and Proper Functions. *British Journal for the Philosophy of Science*, 44, 409–422.
- Hardcastle, V. G. (1999). Understanding functions: A pragmatic approach. In V. G. Hardcastle (Ed.), *When biology meets philosophy* (pp. 27–46). Cambridge, MA: MIT Press.
- Hull, D. L. (1967). Certainty and Circularity in Evolutionary Biology. *Evolution*, 21, 174–189.
- Humberstone, I. L. (1997). Two Types of Circularity. *Philosophy and Phenomenological Research*, 57(2), 249–280.
- Isaac, A. (2013). Objective similarity and mental representation. *Australasian Journal of Philosophy*, 91(4), 683–704.
- Ladyman, J., & Ross, D. (2007). *Everything must go*. Oxford: OUP.
- Machamer, P., Darden, L., & Craver, C. (2000). Thinking about Mechanisms. *Philosophy of Science*, 67, 1–25.
- Maley, C. (2018). Toward analog neural computation. *Minds and Machines*, 28(1), 77–91.
- Maley, C., & Piccinini, G. (2017). A unified mechanistic account of teleological functions for psychology and neuroscience. In D. M. Kaplan (Ed.), *Explanation and integration in mind and brain science*. Oxford: OUP.
- Massimi, M. (2016). Four kinds of perspectival truth. *Philosophy and Phenomenological Research*, 96(2), 342–359.
- Milkowski, M. (2013). *Explaining the computational mind*. Cambridge, MA: MIT Press.
- Millhouse, T. (2018). A simplicity criterion for physical computation. *British Journal for the Philosophy of Science*. <https://doi.org/10.1093/bjps/axx046>.
- Millikan, R. (1989). In Defense of Proper Functions. *Philosophy of Science*, 56(2), 288–302.
- Neander, K. (1991). Functions as selected effects: The conceptual analyst's defence. *Philosophy of Science*, 58(2), 168–184.
- Pexton, M. (2015). Emergence and fundamentality in a pancomputationalist universe. *Minds and Machines*, 25, 301–320.
- Piccinini, G. (2015). *Physical computation*. Oxford: OUP.
- Piccinini, G., & Bahar, S. (2012). Neural Computation and the Computational Theory of Cognition. *Cognitive Science*, 37(3), 453–488.
- Piccinini, G., & Craver, C. (2011). Integrating psychology and neuroscience: Functional analyses as mechanism sketches. *Synthese*, 183, 283–311.
- Putnam, H. (1988). *Representation and reality*. Cambridge, MA: MIT Press.
- Schiller, H. I. (2018). The swapping constraint. *Minds and Machines*. <https://doi.org/10.1007/s11023-018-9473-6>.
- Schwartz, P. (2002). The continuing usefulness account of proper functions. In A. Ariew, R. Cummins, & M. Perlman (Eds.), *Functions: New essays in the philosophy of psychology and biology* (pp. 244–260). Oxford: OUP.
- Schweizer, P. (2014). Algorithms Implemented in Space and Time. In *Proceedings of the 50th anniversary convention of the AISB*.
- Schweizer, P. (2016). In what sense does the brain compute? In V. C. Müller (Ed.), *Computing and philosophy*. Heidelberg: Springer (Synthese Library).
- Searle, J. (1992). *The rediscovery of the mind*. Cambridge, MA: MIT Press.
- Shagrir, O. (2001). Content, computation and externalism. *Mind*, 110(438), 369–400.
- Shagrir, O., & Bechtel, W. (2017). Marr's computational level and delineating phenomena. In D. M. Kaplan (Ed.), *Explanation and integration in mind and brain science*. Oxford: OUP.
- Toffoli, T. (1982). Physics and computation. *International Journal of Theoretical Physics*, 21(3–4), 165–175.
- Walton, D. N. (1985). Are circular arguments necessarily vicious? *American Philosophical Quarterly*, 22(4), 263–274.
- Wheeler, J. A. (1982). The computer and the universe. *International Journal of Theoretical Physics*, 21(6–7), 557–572.
- Wolfram, S. (2002). *A new kind of science*. Champaign, IL: Wolfram Media.
- Zuse, K. (1970). *Calculating space*. Cambridge, MA: MIT Press.
- Zuse, K. (1982). The computing universe. *International Journal of Theoretical Physics*, 21(6–7), 589–600.