Are There Teleological Functions to Compute? Preprint

Dimitri Coelho Mollo
Berlin School of Mind and Brain, and Institut für Philosophie
Humboldt-Universität zu Berlin

Published in *Philosophy of Science* (2019)
https://doi.org/10.1086/703554
Please quote only from the published version

Abstract

I analyse a tension at the core of the mechanistic view of computation, generated by its joint commitment to the medium-independence of computational vehicles, and to computational systems possessing teleological functions to compute. While computation is individuated in medium-independent terms, teleology is sensitive to the constitutive physical properties of vehicles. This tension spells trouble for the mechanistic view, suggesting that there can be no teleological functions to compute. I argue that, once considerations about the relevant function-bestowing factors for computational systems are brought to bear, the tension dissolves: physical systems can have the teleological function to compute.

1 Introduction

The notion of computation plays an important explanatory role in several areas of science. Its application is perhaps most obvious, and least surprising, in computer science. It appears as well, and as a fundamental notion, in several branches of mainstream cognitive science and artificial intelligence research. Talk of computation is also widespread in the many fields of knowledge that employ computational modelling, from astronomy, to meteorology, to molecular biology.

There is an essential difference between the role of the notion of computation in computer and cognitive science, and its usage in other sciences. In the latter, computation most often works merely as a tool to model systems that in themselves are not computational. The computer and cognitive sciences, on the other hand, focus on systems that are supposed not only to allow computational modelling, but that are essentially computational. Electronic computers, and arguably cognitive systems, really compute. In order to capture the nature of these systems, an account of what it is for a physical system to perform computations (*vs.* to be merely computationally describable) is called for: an account of physical, or concrete computation (Piccinini 2015)[1].

In this paper, I defend the cogency of a recent theory of concrete computation: the mechanistic view (Piccinini 2015; Coelho Mollo 2018; Milkowski 2013; Fresco 2014). The mechanistic account of concrete computation is particularly promising for providing a robust, objective notion of concrete computation. However, it is a highly articulated theory, with several movable parts making different contributions to its structure. It is thereby important to ensure that all those parts are sufficiently solid, firmly attached, and well-oiled, if the theoretical machinery is to work properly. I will here tackle one apparently weak spot in the mechanistic view of concrete computation: its joint commitment to the medium-independent nature of computational vehicles, and to the existence of teleological functions to compute. This joint commitment leads to a puzzle. The best theories of teleological function available have recourse to the specific (past or present) causal powers of vehicles in bestowing teleofunctions to them and the systems they help compose. But medium-independence abstracts away from precisely such specific causal powers. How then can there be teleological functions to manipulate medium-independent vehicles, that is, to compute?

I will try and argue that the mechanistic view has the tools satisfactorily to answer this challenge, remaining our best picture of the nature of computation in physical systems. Here is how I will proceed in what follows. In section §2, I briefly present the mechanistic

[1] The objectivity of computation has been rejected by some scholars (e.g. Searle 1992; Schweizer 2016). Since these views jeopardise the explanatory role of computation in the computer and cognitive sciences, and are at odds with their practices, I will not tackle them here.

view of concrete computation, focusing especially on the notions of medium-independence and multiple realisability. In section §3 I elaborate on the puzzle hinted at above, and which it is my task in this paper to solve. In section §4 I examine some strategies to answer the challenge, which though unsuccessful, reveal interesting features of the conceptual architecture of the mechanistic view of computation. In particular, I show that the view cannot do without appeal to both medium-independent individuation and teleology. Finally, in section §5 I show that the mechanistic view is able to solve the puzzle once considerations about the causally relevant factors that bestow teleological functions on designed and biological computational systems are brought to the fore.

2 Mechanistic Computation

2.1 Mechanistic Explanation

Before going into the details of the mechanistic view of concrete computation, a brief presentation of the mechanistic approach to scientific explanation, or New Mechanism, is in order. My treatment here will be rather cursory[2], centred on the aspects of the framework that are relevant for my purposes.

According to the mechanistic approach, scientific explanation, at least in the special sciences, proceeds by providing the mechanism for a phenomenon to be explained. Mechanisms are composed of components organised in such a way that their causal interactions explain aspects of the behaviour of a system of interest. Different accounts of mechanistic explanation cash out the details differently, but for our present purposes we can ignore these subtleties, and stick to the basic notion of mechanism put forward by Illari and Williamson (2012). According to their basic definition, "a mechanism for a phenomenon consists of entities and activities organised in such a way that they are responsible for the phenomenon" (ibid., 120).

The fundamental idea, which Illari and Williamsom's formulation nicely captures, is that, given a phenomenon that we are interested in explaining, explanation proceeds by decomposing the relevant system into its relevant parts — the components, or entities

[2] For detailed treatment, see Craver and Darden (2013); Andersen (2014a,b).

— seeing what the parts do — their activities — and how the overall organisation of the parts and their activities underlie, or lead to the production, or maintenance, of the target phenomenon[3]. A mechanism is individuated partly in terms of the phenomenon that it explains. Components and activities are individuated in terms of their relevance for the explanation of how the mechanism leads to the behaviour of the system under investigation. Components and activities have functions, understood as their specific causal contributions to the overall mechanism. I will refer to this notion of function, i.e. having to do with the causal roles of components and processes inside a system — as systemic functions (Cummins 1975; Craver 2001).

Mechanisms and their components may also have functions in a different sense. They may have ends or purposes, they may be *for* something — that is, they may possess teleological functions, or teleofunctions for short[4]. Teleofunctions capture what a system or component is for in a way that can be to some extent independent from current causal roles. Teleofunctions introduce a dimension of normativity, insofar as there is something it is for a component or system to be working properly, or failing to do so. A heart that does not pump blood rhythmically, but rather at random intervals, is a malfunctioning heart — even though it can be mechanistically decomposed into components which have systemic functions that explain how come it pumps blood arrhythmically. There is nothing it is for a component to successfully or unsuccessfully perform its systemic function per se, independently of our expectations about what the component should do. Systemic functions and teleofunctions must therefore be kept apart (Godfrey-Smith 1993).

Mechanisms that possess teleofunctions are teleofunctional mechanisms (Garson 2013; Piccinini 2015). Many mechanisms are not teleofunctional. Though they have components that perform activities that explain a phenomenon, they have no end or purpose — think about planetary systems, or the water cycle. These systems can be broken down into their components and what they do in order to explain how they work. Albeit their components have systemic functions, the overall mechanisms have no teleofunction, and therefore cannot succeed or fail in any substantial way. The water cycle can be altered in

[3] The system that features the behaviour under investigation must be kept distinct from the mechanism that explains that behaviour of the system. The system and the mechanism are conceptually distinct, and often also physically distinct (Anderson 2015). A system may be decomposed into different mechanisms relevant for its different behaviours.

[4] See the essays in Huneman (2013) for a survey of recent work on teleological functions.

its workings due to changes in oceanic temperature, diminished evaporation due to urban development, etc. But it is not thereby failing to perform a teleofunction, as it has no end or purpose: it is just working differently. The water cycle is a mechanism, but not a teleofunctional one.

Mechanistic explanation normally spans multiple levels of organisation of a system. For most explanatory projects, only some levels are relevant. To explain blood circulation, for instance, it is not necessary — and not desirable — to go down to the level of atoms. Abstraction from irrelevant detail is a vital part of scientific, and mechanistic explanation (Craver and Kaplan Forthcoming). Depending on the phenomenon at hand, different levels of organisation of the system will be relevant for its explanation. Detail at all levels of organisation is not only unnecessary; it is harmful to good explanation.

2.2 Computing Mechanisms

Several theorists have recently proposed that the mechanistic approach to scientific explanation provides the right framework for making sense of computation in physical systems, that is, concrete computation (Piccinini 2015; Coelho Mollo 2018; Milkowski 2013; Fresco 2014). I will focus on the versions of the view defended by Piccinini (2015) and myself (2018), which I take to be the most promising.

According to these accounts, computational systems are a type of teleofunctional mechanism. In particular, they are teleofunctional mechanisms endowed with the teleofunction of performing (concrete) computations[5]. Piccinini (2015, 121-22) offers the following general definition, which is meant to capture all types of physical computational systems, such as digital, analogue, and quantum (slightly adapted):

> A Physical Computing System is a physical system with the following characteristics:
>
> • It is a teleofunctional mechanism — that is, a mechanism that has teleofunctions.
>
> • One of its teleofunctions is to perform computations.

[5] I will often drop the qualifier 'concrete' in what follows, as I will be concerned exclusively with that notion.

Computation, in its turn, is defined as follows (ibid.):

- The processing of vehicles by a teleofunctional mechanism according to rules that are sensitive solely to differences between different portions (i.e., spatiotemporal parts) of the vehicles — along specific dimensions of variation.
- Rule: a mapping from inputs I (and possibly internal states S) to outputs O.

In my (2018) I offer an amended version of the view, bringing explicitly to the fore the differing commitments regarding computational individuation and computational implementation. According to this account, aimed at avoiding potentially fatal objections moved by Haimovici (2013), computation is individuated purely functionally (*contra* Piccinini 2015, 124-25); while the appeal to mechanism plays a role only in accounting for how computations are implemented in physical systems. What makes the mechanistic view mechanistic, I argue, is the claim that those systems in the world that actually perform computations are mechanisms, and more precisely, teleofunctional mechanisms, to which mechanistic explanation naturally applies. Concrete computation, in my view, is individuated by the following four-way functional characterisation (Coelho Mollo 2018, slightly adapted):

1. Manipulation of medium-independent vehicles according to a rule sensitive only to their degrees of freedom.
2. The medium-independent vehicles are functionally-individuated components of a teleofunctional mechanism.
3. The manipulations that vehicles undergo are activities internal to a teleofunctional mechanism.
4. It is one of the teleological functions of the teleofunctional mechanism to carry out 1.

How the functional characterisation is implemented case by case is to be explained mechanistically, and involves detail about the physical constitution of the implementing system and of the components of the relevant mechanism. Importantly, for Piccinini and myself, a crucial feature that distinguishes computational systems is that their vehicles are individuated in medium-independent terms. Indeed, the rules that govern the behaviour of a computational system are sensitive only to the degrees of freedom of its components. Degrees of freedom, as I will be using the term, are dimensions of variation of physical variables: for instance, a rigid robot that can only move forwards, backwards, left, and right, has two degrees of freedom, insofar as its position can vary only along two spatial dimensions. Individuating systems by means of their degrees of freedom abstracts away from details about their physical constitutions, as the nature of the relevant physical variables that can vary is irrelevant — only the dimensions of variation matter.

The physical constitution of computational systems is largely irrelevant to their computational nature, as only having the appropriate degrees of freedom matters. For instance, a system can perform digital computation if it can stabilise into two distinguishable classes of physical states to which the rules for vehicle-manipulation are sensitive (and if there is a mechanism with the teleofunction of performing computations). These equivalence classes of physical states may be voltage intervals, presence/absence of beer tins in a certain location, etc. For this reason, physical systems made out of different materials can all be computational, and even perform the same computations — provided that their vehicles have the appropriate degrees of freedom. More strongly, it is partly the fact that physical constitution is largely irrelevant to their nature that makes computational systems computational. The fact that computational vehicles are individuated in medium-independent terms helps drive a wedge between computational and non-computational systems.

In sum, computational systems belong to the subset of teleofunctional mechanisms (itself a subset of mechanisms in general) characterised by the fact that vehicle individuation is medium-independent. The importance of the notion of medium-independence for a theory of computation has been recognised several decades before the advent of the mechanistic view (Haugeland 1985). The interest of the notion goes thereby beyond its application in mechanistic views of computation. At any rate, given the central role played by the notion of medium-independence in the foregoing mechanistic account of computation, it is worthwhile to examine it carefully, distinguishing it from a closely related notion, that of multiple realisability.

2.3 Medium-independence and Multiple Realisability

Medium-independent individuation places no constraints on the physical constitution of the relevant vehicles beside their having adequate degrees of freedom. This makes it possible to type together systems made of different kinds of physical components. But reliance on degrees of freedom is not the only strategy to type together physically different systems in an explanatorily useful way. We can also type systems in terms of what they do (systemic functions), or of what they are supposed to do (teleofunctions). Many functions, systemic or teleological, biological or artefactual, can be carried out by more than one arrangement of physical states and processes. It follows that some functional types are multiply realisable: systems with different physical constitutions and with different internal functional organisations can all perform the same overall function, therefore belonging to the same functional type[6].

One of the favourite case studies in recent debate about multiple realisability is that of corkscrews. To be a corkscrew is to be a physical system able to, or supposed to open bottles by lifting their corks. Several types of physical system have been designed to perform that function, perhaps the most common of which are waiter's and winged corkscrews, though it is likely that one has had at some point to confront more basic types, composed only of a handle and a screw, with, at least in the case of this author, less than satisfactory outcomes. There are two main ways in which corkscrews are multiply realisable (and realised). First, physical systems functionally arranged in different ways, with different causal profiles, can be corkscrews. Winged corkscrews have a rather different causal organisation than basic corkscrews. Second, corkscrews can be made of different materials — plastic, metal, wooden corkscrews are all corkscrews, provided that they fulfil the function, or at least belong to a type that fulfils the function, of opening corked bottles by lifting their corks. Corkscrews are thereby multiply realisable. Not all functionally-defined types are multiply realisable. As Shapiro (2000, 641) points out, a functional type defined by the capacity to scratch sapphires is not multiply realisable, as it so happens

[6] In recent years, the cogency of the notion of multiple realisability has been put into doubt, together with its applications to understanding the mind and the explanatory nature of the special sciences (Shapiro 2000). These attacks on multiple realisability have been met with compelling counterarguments (Aizawa and Gillett 2009), and discussion of these issues is ongoing. I will not delve into these recent and complex debates here, and will use throughout a fairly classical treatment of multiple realisability.

that only diamonds are hard enough to do so.

These considerations bring to the fore the main difference between the notions of multiple realisation and medium-independence. It is a difference of degree[7]. Some functional types are defined by functions that can only be performed by physical systems with specific physical properties. In the case of 'sapphire scratchers', only diamonds can belong to the functional type, as only one type of physical property can realise it. That functional type is thereby not multiply realisable. Other functional types place different degrees of physical constraints on their realisers. Any system realising a corkscrew must be made of materials able to exert enough force on corks so as to be able to remove them from bottles (without breaking the bottle in the process). This places strong constraints on the physical constitution of corkscrews, e.g. that they be rigid enough (even corkscrews that use air pressure to remove corks have rigid parts that perforate the cork and pump air below it); that they in some way interact appropriately with a physical system shaped and constituted in a particular way — i.e. the corked bottle — and so on. Some functional types may place looser constraints on the physical constitution of their realisers.

Medium-independence lies at the opposing end of the spectrum in relation to sapphire scratchers. It concerns those functional types that place no constraints on the physical constitution of realising systems, but only on their degrees of freedom. According to the computational mechanist, computational types belong to this kind of functional type. While non-computational systems may be multiply realisable to different extents, they fall short of medium-independence. Computational systems are defined in medium-independent terms, having, as it were, the highest degree of multiple realisability. As long as a system, and its internal vehicles, have the appropriate degrees of freedom, their further physical properties are irrelevant to their potential computational role[8].

Beside medium-independent individuation, the mechanistic account poses additional constraints on the kinds of systems that can physically implement computations (see section 2.2). These constraints disqualify most physical systems from counting as computational, keeping at bay the threat of pancomputationalism, i.e. the idea that all or most physical systems perform computations, which risks trivialising the notion of concrete com-

[7] I thank Michael Pauen for pressing me on this point.

[8] Piccinini and Maley (2014) explore in detail how the medium-independence of concrete computation leads to the many ways in which computational systems are multiply realisable.

putation. Most, if not all, physical systems can be described in extremely abstract fashion, leaving most of their physical details out. But the mechanistic view insists that being computationally describable should not be equated with actually computing (Piccinini 2015, 145). That a physical system can be very abstractly described, and even computationally modelled, does not entail that the system itself is computational. Weather systems are routinely computationally modelled in order to predict their behaviour. They are not though computational. We can now see why. First, weather systems are not medium-independently characterised — to be a weather system involves being composed of large amounts of gas molecules of certain kinds (depending on atmospheric composition); and having causal powers that depend on their intrinsic physical properties (e.g. density, temperature, etc.). Second, weather systems are not teleofunctional mechanisms — they have no teleofunctions (though they do have systemic functions). They are thereby excluded from being candidates for computational nature.

Though the mechanistic view is satisfactorily able to distinguish computational from non-computational systems, its complex conceptual architecture represents a threat to its cogency. With its appeal to theoretical notions such as medium-independent individuation, functional and mechanistic explanation, and teleological functions, the account presents several conceptual parts that are in themselves rather articulated. Can they fit together? In particular, can medium-independent individuation and teleology work together? More concretely yet, can there be teleofunctions to compute, i.e. to manipulate vehicles individuated in medium-independent terms?

The rest of this paper will be concerned with fully formulating and responding to the latter worry.

3 Teleological Functions and Medium-independence: a Puzzle

According to the mechanistic view of computation, teleological functions play an essential role in helping to individuate computational systems[9]. In this section, I examine a difficulty that arises for the view in this connection. For it would seem, according to our best

[9] Though see Dewhurst (2018a) for an attempt to make do without appeal to teleological functions.

theories, that the factors that determine the teleological functions of a system are sensitive to causal powers that hinge on details of its physical constitution — and cannot thereby yield functions individuated in medium-independent terms. It would follow that there can be no teleological functions to compute, and in consequence that no physical system satisfies the requirements for being a computing system by the computational mechanist's own lights[10]. It would seem, that is, that the mechanistic view of computation cannot appeal to medium-independence and teleological functions at the same time. Let us cash out this dilemma more fully, taking first a quick look at philosophical treatment of teleological functions.

Accounting for the notion of teleological function in scientifically-acceptable, naturalistic terms has proved to be a rather challenging endeavour. Indeed, the notion is related to concepts such as purpose and end that imply ideas of design and agency which are not straightforwardly accounted for in non-intentional (and non-theological) terms. Compelling and fully naturalistic theories of teleological function have nonetheless been developed; and even though scepticism about them occasionally comes to surface, the consensus for some time has been that teleological functions are *bona fide* scientific properties, especially central to the life sciences (Godfrey-Smith 1993).

Two main approaches to teleofunction have been particularly influential in the debate: dispositional theories, and selected-effects theories.

Roughly, according to dispositional theories the teleological functions of a type of system (or parts thereof) are determined by factors having to do with the occurrent causal dispositions of a (type of) system or its parts. Dispositional theories are thereby ahistorical. Dispositions are understood as counterfactually-robust regularities: were certain conditions to hold, a system or part would manifest a certain behaviour. Hearts have the disposition to pump blood, for when they are introduced in a complex system with specific characteristics, they display blood-pumping behaviour. The most promising dispositional theories of teleological functions are goal-based. They have it that the teleofunctions of a system are determined by its dispositions to contribute to certain goals of organisms (Boorse 1976; Maley and Piccinini 2017). The relevant goals are fixed by independent considerations, with the more plausible candidates including survival, reproduction, and

[10] This problem was first brought to my attention in discussion with Nicholas Shea.

inclusive fitness. In this picture, hearts have the function to pump blood because by pumping blood they contribute to the survival and inclusive fitness of organisms having them.

Selected-effects theories, in contrast, focus on the causal contributions that past instances of a type of system or its parts made that explain the existence and historical persistence of instances of that type of system or part (Millikan 1984; Neander 1991). Selected-effects theories are historical: they make reference to the past history of systems and their parts in order to fix their teleofunctions. The function of a heart is to pump blood, for it is because past instances of hearts in the phylogeny pumped blood that organisms having them survived and reproduced, leading to the existence of organisms with hearts today. Some views have it that natural selection is the only kind of selection that bestows functions on physical (biological) systems (Neander 1991; Godfrey-Smith 1994). Others tend to be more liberal, and allow other kinds of selection processes to play a role in bestowing teleological functions (Garson 2011).

The dispute between dispositional and selected-effects theories has been taken to be, at least in part, a matter of different explanatory focus. While dispositional theories are concerned with answering questions about how a type of system works, selected-effects theories focus on explaining why a certain type of system persists over time. Depending on which of these two questions one is interested in answering, one or the other type of theory may be more adequate (Godfrey-Smith 1993). This conciliatory outlook on the debate is not widely shared, with proponents of both approaches insisting that there are more substantial dimensions to the dispute — especially regarding descriptive and explanatory adequacy, and normativity (Millikan 1984; Neander 1991; Garson 2011; Piccinini 2015). For my purposes, there is no need to get into this debate.

This brief presentation of the two main approaches to explaining teleological functions suffices to bring to the fore more clearly the difficulty that the mechanistic account of computation has to confront.

The Puzzle Theories of teleofunction rely on causal contributions, present or historical, of systems and their parts. These causal contributions explain either their furtherance of some privileged class of organismic goals (goal-based dispositional theories), or their persistence over time (selected-effects theories). These causal con-

> tributions are not medium-independent: they hinge on the specific causal powers of physical systems and their parts, or past instances thereof. Their causal powers depend on what the systems, and what they interact with, are made of. If this is so, then the factors that ground bestowal of teleofunctions according to our best theories of teleological function are medium-dependent properties of systems and their parts. Theories of teleofunction cannot contemplate, therefore, the bestowing of teleological functions that involve medium-independent vehicles. The causal powers of systems and their parts that justify ascription of teleological functions, both in dispositional and selected-effects theories, are individuated in terms that cannot abstract away from their physical constitutions — and cannot thereby be individuated in a medium-independent way. Even though abstract medium-independent descriptions of systems and parts are possible (as much as computational descriptions usually are), they are not the descriptions that capture the factors that bestow teleological functions on them. For it is partly due to causal details dependent on the physical constitution of systems and their parts that a story about how they acquire teleological functions gets off the ground. In sum, it follows from our best theories of teleological function that there cannot be teleofunctions individuated in a medium-independent way. In consequence, there cannot be teleomechanisms with the teleological function to compute, *contra* the mechanistic view of concrete computation.

The mechanistic view of concrete computation seems hence to be in dire straits: it claims that computational systems are those mechanisms with the teleological function to perform computations. But the performance of computations, which essentially involves medium-independent individuation, seems to be a bad candidate for being a teleological function of any system — for teleological functions seem to hinge on medium-dependent properties of systems and their components. It looks like the computational mechanistic has to give up either the appeal to teleology, or the appeal to medium-independence. In the next section I will assess whether the mechanistic view can survive abandoning either of these two components of the theory (the answer will be 'no'). In the following section I will show that hope is not lost: there is a solution to the puzzle after all — there can be teleological functions individuated in medium-independent terms.

4 Some (unpromising) Responses

There seem to be two immediate ways out for the computational mechanist: i) eliminate or change the role that medium-independence plays in the individuation of concrete computation, so as to make it compatible with appeal to teleology; or ii) eliminate or change the role that teleological functions play in the individuation of concrete computation, so as to make it compatible with medium-independence. Let us examine each in turn.

It may be claimed that the appeal to medium-independent individuation does not eliminate or screen-off those causal contributions of vehicles that depend on their physical constitutions. Computational mechanists can insist that computational vehicles are manipulated in light of a subset of their physical, medium-dependent properties — those to which the general rule of manipulation is sensitive. This is the line that seems to be taken by Piccinini, as his definition presented in section 2.2 above suggests. Teleological functions could thereby be bestowed on computational systems in light of the causal contributions that this subset of physical properties makes to organismic goals or to their persistence through selection processes. In other words, though computational systems and vehicles are individuated in medium-independent terms, the factors that bestow the teleological function to compute on them are determined partly by the constitutive properties of the realising physical media.

I believe that this line of reply will not do. It throws the mechanistic view into a dilemma. It has either to embrace pancomputationalism (or something close to it), or give up multiple realisability.

On one hand, the computational mechanist may hold that the teleological functions of computational systems are to transform certain physical quantities in a rule-like way. On this picture, it would be transformation of physical quantities in rule-like fashion what contributes to organismic goals or to persistence in the phylogeny, thus determining the computational teleofunctions of the system. The key problem with this move is that this way of individuating computational teleofunctions is too liberal. This sort of teleofunction can be found in a host of systems, many of which intuitively non-computational. Cell membranes, thermometers and simple thermostats have the teleological functions they do due to their transforming certain physical quantities into others in a regular, rule-like manner (e.g. internal ion concentrations into other internal ion concentrations, average kinetic energy into height of mercury column or into bending of a metal strip). Their teleofunctions would hence count as computational. As they are also plausible examples of teleofunctional mechanisms, they would fulfil the characterisation of physical computational systems given in section 2.2. But ascribing computational nature to such systems seems implausible. Taking such liberally individuated teleological functions to be computational teleofunctions leads to a considerable extension of the domain of computational systems, well beyond intuitive boundaries, bordering on pancomputationalism.

On the other hand, if it be insisted, to avoid an overextension of the domain of computational systems, that teleological functions be more narrowly determined, for instance in terms of the particular kinds of physical quantities that each computational system manipulates, then the characteristic multiple realisability of computations would be lost, and with it, a useful notion of computational equivalence. For then the teleological functions of computational systems would be to manipulate vehicles with a particular type of physical constitution, e.g. voltage ranges in silicon circuits. Electronic computational systems, hydraulic computational systems, and neural-biological computational systems would not be able to perform the same computations, given that each of these kinds of systems would have the teleofunction to manipulate different kinds of physical quantities, and their computations would thus be individuated differently.

Pancomputationalism, as well as the abandonment of the multiple realisability of computation and of (broad) computational equivalence, have been taken by some not to be knock-down arguments against theories of computation[11]. But they are big bullets to bite, as they deprive the notion of concrete computation of most of its explanatory power. In sum, taking line of reply i) does not help the case of the computational mechanist. It leads to a dilemma, with each horn abutting on unfortunate consequences. Let us briefly see whether strategy ii) has something to recommend it.

The appeal to teleological functions plays a multipronged role in the mechanistic view of computation. Changing or eliminating appeal to the notion puts in jeopardy several positive aspects of the theory. I will mention three.

First, the appeal to teleological functions, taken as objective, observer-independent

[11] Chalmers (2011) defends a sophisticated pancomputationalist view. Abandoning multiple realisability and computational equivalence has been adumbrated by Dewhurst (2018b), but see Coelho Mollo (2018) for a reply.

features of the (biological) world, helps ensure the objectivity of concrete computation. As we have seen, mechanisms are partly individuated by a phenomenon or capacity of interest. Some of these phenomena or capacities may be essentially dependent on the explanatory aims and practices of theorists, thus having a marked subjective component to them. Moreover, any physical system can be mechanistically analysed in light of several different capacities. An electronic computer may be mechanistically analysed in terms of its capacity to generate heat. The claim that computational systems have the teleological function to compute provides justification both to the view that concrete computations are objective, given that some physical systems have the objective teleofunction to perform them[12]; and to the view that their capacity to perform computations is privileged over other capacities they may have, but which are not among their teleological functions, e.g. the capacity to generate heat. Reducing or eliminating the appeal to teleofunctions undermines the realist approach to concrete computation that mechanistic views want to maintain.

Second, and relatedly, teleofunctions help define the domain of computational systems in a way that is not too liberal, i.e. that does not include more systems than intuitively plausible, avoiding pancomputationalism. By individuating computation partly by means of teleological functions, the mechanistic view considerably narrows down the domain of physical systems that can count as computational. They must be teleomechanisms — ruling out mechanisms, such as weather systems, that do not have teleofunctions — and they must have the teleofunction to compute — circumscribing a rather narrow domain of physical systems that can be candidates for computational nature. Letting go of teleological functions thus risks undermining the ability of the mechanistic view appropriately to distinguish computational from non-computational physical systems.

Finally, teleofunctions introduce a dimension of normativity that is important in making sense of the possibility of miscomputation, that is, of the fact that there is something it is for computational systems to succeed or fail to perform computations appropriately [13]. Appeal to teleofunctions provides a straightforward way of accounting for this fact: computational systems miscompute when they produce outputs that are at odds with the

[12] This keeps at bay views that see concrete computation as observer-dependent, such as Searle (1992); Schweizer (2016).

[13] See Fresco and Primiero (2013); Dewhurst (2014); Piccinini (2015, 148-50).

computational function it is their teleological function to compute.

In sum, reducing or eliminating the role played by teleological functions in the mechanistic account is ill-advised. The second line of reply, as the first, creates more problems that it solves. Mechanisms, it appears, cannot have the teleological function to perform concrete computations, since these are individuated in a medium-independent way, while part of the conditions for bestowing teleofunctions, i.e. the intrisic causal powers of components and systems, are not medium-independent. If that is so, and either medium-independence or appeal to teleofunctions has to be abandoned, the computational mechanist ends up with a rather unsatisfying set of alternatives in their hands: i) either embrace pancomputationalism or give up strong multiple realisability; or ii) jeopardise the objectivity of computation and the possibility of miscomputation, all the while risking to fall into pancomputationalism.

The discouraging moral of this section points to an architectural feature of the mechanistic view of computation; an argumentative structure that cannot be meddled too much with, on pain of losing the advantages that the account is supposed to have. Move a part of the account a bit, and the whole edifice crumbles. This is so because the theory spins a closely-knit web between teleological functions, computational rules, medium-independence, and the relevant causal contributions of physical media. The subset of physical properties that makes the causal contributions of relevance to the computational behaviour of a system are determined by the general manipulation rule, i.e. the computational rule. It is the general rule, with its sensitivity to the dimensions of variation of some physical variables, but not others, that determines what parts of the causal structure of the system are important for its computational nature. But such a general rule calls for an objective factor to ground it. For there is an indefinite number of ways of mapping causal goings-on in physical systems onto computational rules (Putnam 1988; Chalmers 1996). Just looking at the causal operations of the system cannot reveal the computational rule, unless one knows what to look for, i.e. if one knows what causal goings-on are computationally relevant. Such an objective factor, according to the mechanistic view, is teleology. It is by having the performance of one or more computations as its teleological function that a system can be carved into the physical components and processes that are causally relevant for performing that function. However, it is medium-independent individuation that allows setting computational rules, and thus computational teleofunctions apart from rule-like behaviour and non-computational teleofunctions — setting computational and non-computational systems apart.

The failed attempts examined in this section notwithstanding, I believe that there is a way to keep this fragile conceptual web as it is, saving the cogency of the mechanistic view of computation, and safeguarding its qualities as a promising theory of the nature of computation in the physical world. Let us turn to that.

5 Teleofunctions to Compute

In this section, I argue that, contrary to the arguments in section §3, systems and vehicles individuated in medium-independent terms can be involved in the bestowing of teleological functions, at least in the special domain of computational systems. While the case of designed computational systems is relatively straightforward, cognitive computational systems place further challenges which, I will argue, can be addressed only by accepting a version of functionalism about cognition. Given the plausibility of functionalism, together with its foundational role in (much of) the cognitive sciences, I take that this is not a high price to pay for saving the cogency of the mechanistic view of computation.

Before moving to the more complicated and controversial case of cognitive computational systems, it is best to start with the easier one: designed computational systems. Though the debate is open on how best to account for the teleofunctions possessed by artefacts, it is plausible that the intentions of designers play some role in their bestowal, at least in the case of (human-)designed artefacts[14]. On a goal-based theory artefacts can be seen as contributing to the relevant goals of organisms, which may also include subjective goals that go beyond survival, reproduction, and inclusive fitness (Piccinini 2015, 115-17). On a selected-effects theory, artefactual teleofunctions can in some cases be determined by biological selection processes, given the specific advantages artefacts bestow on their possessors which explain their persistence; while in other cases being determined by rational selection processes, in which a rational agent selects from alternative designs the one that best fits their intentions and material capabilities, or more basically, by simple

[14] For a useful introduction to the problem of artefactual functions, see Artiga (2016).

trial-and-error [15].

In the case of designed computational systems, the intentions of designers are plausibly to build devices that behave in such a way as to respect the desired computational rules. The latter are typically given a medium-independent specification in terms of logic gates, memory storages, etc., and their mutual interactions, posing constraints only on the degrees of freedom, and the ways they modulate each other, of potential implementing physical systems. Physical media are then arranged so that their components and activities mirror that computational architecture, implementing the computational description in a physical system.

In other words, the computational blueprint lies at the basis of the intentions of designers of computational systems. Specific types of physical media are selected in light of their capacity to implement the medium-independent characterisation that they are supposed to satisfy. Computationally equivalent systems can thereby be built out of different materials and components. The contributions that computational artefacts make to goals or to selection are independent of details about their physical constitution, insofar as computationally equivalent, but differently physically constituted systems make the same contributions. There are likely to be differences in speed, size, energy consumption and so on between different types of physical implementations. Though these may influence to some extent the specific teleological functions of computational mechanisms (e.g. in limiting which functions they can compute in a certain interval of time, or in which circumstances they can be usefully applied), they do not jeopardise their general teleofunction to perform computations. These considerations lend force to the claim that, as far as designed computers are concerned, their teleological functions are primarily characterised in medium-independent terms, for the intentions of designers of computational systems — which help bestow teleofunctions — involve medium-independent blueprints.

The mechanistic view of concrete computation is thus able to avoid the worry presented in section §3, at least for what regards designed computers. The question then becomes whether a similarly fortunate result can be obtained for biological, non-designed computational systems such as, plausibly, cognitive systems. Things get trickier here, as appeals to intentions of designers are clearly not in the cards, but I believe that there are promising

[15] Wright (1973) suggests a view on these lines.

paths for a computational mechanist to tread. I will explore a similar strategy to the one applied to designed computers: showing that medium-independent characterisations of the causal goings-on in biological, cognitive computational systems are explanatorily primary in determining their teleological functions.

In arguing for this, Chalmers' (2011) notion of organisationally invariant properties proves useful. Organisationally invariant properties remain invariant as long as the abstract causal structure of a system — what Chalmers calls its causal topology — remains unchanged. In Chalmers' words, a property P counts as organisationally invariant "if any change to the system that preserves the causal topology preserves P" (2011, 337). The causal topology of a system, in turn, is defined as "the pattern of interaction among parts of the system, abstracted away from the make-up of individual parts and from the way the causal connections are implemented". The notion of organisationally invariant property is closely related to medium-independence. Insofar as such properties are individuated in a way that fully abstracts away from the physical constitution of their realisers, they are individuated in medium-independent terms[16].

The notion of organisationally invariant properties embodies an important assumption of much work in mainstream cognitive science — the endorsement of functionalism. According to functionalism, it is the special abstract causal structure of systems — their causal topology — that makes them into cognitive systems, not the physical constitution of their parts. Any system with the right kind(s) of causal structure will have states and processes that count as cognitive, regardless of the materials their vehicles are made of. What matters, in other words, are the input-output relations between components of the systems, regardless of how they are realised. By the lights of functionalism, what makes a system cognitive is its causal topology; and thereby cognitive states and processes are primarily characterised in terms of organisationally invariant properties. It follows that computational explanation is particularly appropriate to cognitive states and processes, for it captures what is most relevant about them: their role in the abstract, medium-

[16] According to Chalmers, the notion of computation provides the appropriate way of capturing the abstract causal structure of physical systems in general. This leads to his pancomputationalist view of computation: since every physical system has a causal topology which is capturable in computational terms, every physical system is computational. We do not need to embrace Chalmers' pancomputationalism; as we have seen, the mechanistic view deploys additional tools to narrow down the domain of physical systems that are computational.

independent causal structure of the cognitive system (Chalmers 2011, 337).

Functionalism about cognition opens a promising avenue for the computational mechanist, as it suggests that the causally-relevant properties when it comes to cognitive systems are organisationally invariant, and thereby medium-independently individuated. If that is so, then the appropriate way of individuating the causal properties relevant for bestowal of teleological functions, be it in terms of contributions to organismic goals or to persistence in the phylogeny, is in medium-independent fashion. In other words, if cognition is essentially a matter of having the right kind(s) of causal topology, and a cognitive system is such insofar as it instantiates the right kinds of input-output relations between its components, as functionalism has it, then the properties relevant for bestowing teleofunctions on cognitive systems are primarily organisationally invariant ones[17].

Focusing on the medium-dependent causal properties of cognitive systems would miss what is distinctively cognitive about them — their causal topology. This can be clearly illustrated by appealing to counterfactuals. Were cognitive systems composed fully or partly of a different material than neurons and glia, they would function in the same way as actual cognitive systems[18], provided that the causal topology of the former and the latter were the same (and that the different medium is suitably connected to, and able to interact appropriately with sensory surfaces and motor effectors, maintaining the appropriate input-output relations between components). Any change in physical constitution, insofar as it preserves the system's causal topology, preserves its contributions to organismic goals and persistence in the phylogeny[19]. In the case of cognitive systems, that is, what is causally relevant to bestowal of teleological functions, both for goal-based and selected-effects accounts, is their abstract, medium-independently individuated causal structure — which is best captured in computational terms (Chalmers 2011, 342).

[17] Papineau (1994) makes a similar point when defending the irreducibility of psychiatric disorders to physical disorders: as he argues, the former involve cognitive systems or their parts failing to fulfil biological functions individuated in terms of abstract causal structure, rather than physical constitution. As he puts it, in light of his selected-effects theory of teleofunctions: "Natural selection does not mind, so to speak, about the internal physics of pain mechanisms, so long as they link causes and effects in the right way" (ibid., 77). Natural selection, on this picture, plays an analogous role to that of the designer of computers, caring only for the abstract causal structure of systems. See Papineau (1993) for a more general treatment of these issues.

[18] Except for, possibly, differences in speed and size.

[19] Except in cases where differences in speed and size jeopardise those contributions.

In sum, the computational mechanist, if also a functionalist, can reply in the following way to the puzzle I set out above. Theories of teleofunction are concerned with the causally-relevant properties that contribute to organismic goals or persistence. As the foregoing line of reasoning suggests, in the case of cognitive systems, courtesy of functionalism, such properties are organisationally-invariant ones. Organisationally-invariant properties are individuated in a medium-independent way. It follows, as the computational mechanist would want, that cognitive systems possess teleological functions that are characterised in medium-independent terms.

There are thus good reasons to think that teleological functions can be individuated in a medium-independent way in the case of designed and biological computational systems. That means, *contra* the arguments in section §3, that there can be teleological functions to manipulate medium-independent vehicles according to general rules, i.e. there can be teleofunctions to compute. The success of my line of argumentation for what regards cognitive computational systems hinges on the success of functionalism as a framework for understanding cognition. Consequently, the plausibility of the mechanistic view of computation, as I here defend it, goes hand-in-hand with the plausibility of functionalism. I take this to be a comforting outcome. Despite the recent attacks against multiple realisability, and in consequence against the support for functionalism, the latter remains the most promising overall framework for understanding the nature of cognition currently available.

It does not follow from my arguments that computational and functional explanation are identical. While computational explanation is a type of functional explanation, not every functional explanation is computational. This follows from the fact that functional explanation, in contrast to computational explanation, needs not be, and in most cases is not medium-independent (and in some cases may not even involve multiple realisability). Moreover, as we have seen, most systems that are functionally explainable are not teleological, and some may not even qualify for mechanistic explanation (see section 2.1). Most importantly, the foregoing argument applies only to those systems for which organisationally invariant properties play causally-relevant roles, and are thus explanatorily primary in making sense of their workings. Physical systems of enough complexity can in most cases be functionally analysed in terms of the abstract causal roles of their components. However, for most systems their functional analysis does not involve organisationally invariant properties, in contrast to computational systems.

To illustrate the difference, take digestion[20]. A functional decomposition of the digestive system is no doubt possible. There are subsystems for allowing food to get inside the organism, for breaking food into smaller and more easily processable bits, for transporting food to subsystems dedicated to breaking it further into substances of use to the organism, for absorbing those nutrients, and finally for getting rid of the material that fails to be absorbed. The functional decomposition of the digestive system is quite illuminating: we get a breakdown of the functional components, the work they do, and how that work is subdivided into different stages responsible for different parts of the process of digestion.

The properties of subsystems that are relevant for explaining their causal contributions to digestion are not organisationally invariant properties — they are not individuated in medium-independent terms, though they may be multiply realisable. Every realisation of a digestive system must be such as to be able to break bits of food, thereby placing constraints on the possible realising physical media — constraints that go well beyond degrees of freedom. Moreover, digestion cannot take place at all without food, which though a variegated category, also places constraints on the physical constitution of its members, whereby the 'inputs' to the digestive system must be individuated in medium-dependent terms. Therefore, the causal topology of digestion, its abstract causal structure, though interesting, is not what is primarily explanatory of the capacity of digestive systems to digest. In consequence, the teleological functions of digestive systems are best characterised in terms of properties that are not organisationally invariant. Digestive systems are functional, but not computational.

The foregoing considerations lend force to the idea that, when it comes to computational systems, both biological and designed, the factors that ground their having teleological functions are organisationally invariant properties, which are individuated in a medium-independent way. The teleological functions of these systems are thereby to be charaterised in medium-independent terms. Contrary to first appearances, in the domain of designed computers and cognitive systems (in the latter case, courtesy of functionalism), there can be teleofunctions to manipulate medium-independent vehicles according to rules. That is, as the mechanistic view of computation has it, there can be teleological

[20] Digestion is used as an example by both Chalmers (2011) and Piccinini (2015).

functions to compute.

6 Concluding Remarks

In this paper, I have argued for two main points. First, I have shown that the joint commitment to medium-independence and to teleology plays an essential and ineliminable role in the mechanistic view of concrete computation. Second, I have argued that, contrary to appearances, there can be teleological functions characterised in medium-independent terms. A crucial tenet of the mechanistic view of computation is therefore secured: there can be teleological functions to compute. The account, especially in its more recent versions, proves to be conceptually coherent and stable. Furthermore, in light of its many positive features — which I have not had the chance to explore in detail here — it holds much promise for providing an adequate, naturalistic theory of what it is for physical systems to compute.

Acknowledgements I am indebted to Nicholas Shea, Michael Pauen, Margherita Arcangeli, the participants of the Philosophy of Mind colloquium at the Berlin School of Mind and Brain, and two referees to this journal for helpful comments on previous versions of this material.

References

Aizawa, Kenneth and Carl Gillett. 2009. "The (Multiple) Realization of Psychological and other Properties in the Sciences." *Mind & Language* 24 (2): 181–208.

Andersen, Holly. 2014a. "A Field Guide to Mechanisms: Part I." *Philosophy Compass* 9 (4): 274–83.

Andersen, Holly. 2014b. "A Field Guide to Mechanisms: Part II." *Philosophy Compass* 9 (4): 284–93.

Anderson, Michael L. 2015. "Beyond Compositional Constitution in the Brain - Starburst Amacrine Cells and Enabling Constraints." In *Open MIND*, edited by Thomas Metzinger and Jennifer M. Windt. MIND Group. http://dx.doi.org/10.15502/9783958570429.

Artiga, Marc. 2016. “New perspectives on artifactual and biological functions.” *Applied Ontology* 11: 89–102.

Boorse, Christopher. 1976. “Wright on Functions.” *The Philosophical Review* 85 (1): 70–86.

Chalmers, David J. 1996. “Does a rock implement every finite-state automaton?” *Synthese* 108: 309–33.

Chalmers, David J. 2011. “A computational foundation for the study of cognition.” *Journal of Cognitive Science* 12 (4): 323–57.

Coelho Mollo, Dimitri. 2018. “Functional individuation, mechanistic implementation: the proper way of seeing the mechanistic view of concrete computation.” *Synthese* 195: 3477–97.

Craver, Carl F. 2001. “Role Functions, Mechanisms, and Hierarchy.” *Philosophy of Science* 68: 53–74.

Craver, Carl F. and Lindley Darden. 2013. *In Search of Mechanisms: discoveries across the Life Sciences*. Chicago University Press.

Craver, Carl F. and David M. Kaplan. Forthcoming. “Are More Details Better? On the Norms of Completeness for Mechanistic Explanations.” *The British Journal for the Philosophy of Science* `http://dx.doi.org/10.1093/bjps/axy015`.

Cummins, Robert C. 1975. “Functional Analysis.” *Journal of Philosophy* 72 (20): 741–65.

Dewhurst, Joe. 2014. “Mechanistic Miscomputation: a reply to Fresco and Primiero.” *Philosophy and Technology* 27: 495–98.

Dewhurst, Joe. 2018a. “Computing Mechanisms Without Proper Functions.” *Minds & Machines* 28 (3): 569–88.

Dewhurst, Joe. 2018b. “Individuation without Representation.” *British Journal for the Philosophy of Science* 69 (1): 103–16.

Fresco, Nir. 2014. *Physical Computation and Cognitive Science*. Springer.

Fresco, Nir and Giuseppe Primiero. 2013. “Miscomputation.” *Philosophy and Technology* 26 (3): 253–72.

Garson, Justin. 2011. “Selected effects and causal role functions in the brain: the case for an etiological approach to neuroscience.” *Biology and Philosophy* 26 (4): 547–65.

Garson, Justin. 2013. “The Functional Sense of Mechanism.” *Philosophy of Science* 80:

317–33.

Godfrey-Smith, Peter. 1993. "Functions: Consensus Without Unity." *Pacific Philosophical Quarterly* 74 (3): 196–208.

Godfrey-Smith, Peter. 1994. "A Modern History Theory of Functions." *Noûs* 28 (3): 344–62.

Haimovici, Sabrina. 2013. "A Problem for the Mechanistic Account of Computation." *Journal of Cognitive Science* 14: 151–81.

Haugeland, John. 1985. *Artificial Intelligence: The Very Idea.* MIT Press.

Huneman, Philippe, ed. 2013. *Functions: selection and mechanisms.* Springer.

Illari, Phyllis McKay and Jon Williamson. 2012. "What is a Mechanism? Thinking About Mechanisms Across the Sciences." *European Journal for the Philosophy of Science* 2: 119–35.

Maley, Corey J. and Gualtiero Piccinini. 2017. "A Unified Mechanistic Account of Teleological Functions for Psychology and Neuroscience." In *Explanation and Integration in Mind and Brain Science*, edited by David M. Kaplan, 236–54. Oxford University Press.

Milkowski, Marcin. 2013. *Explaining the Computational Mind.* The MIT Press.

Millikan, Ruth Garrett. 1984. *Language, Thought, and Other Biological Categories: New Foundations for Realism.* MIT Press.

Neander, Karen. 1991. "Functions as Selected Effects: The Conceptual Analyst's Defence." *Philosophy of Science* 58 (2): 168–84.

Papineau, David. 1993. *Philosophical Naturalism.* Blackwell.

Papineau, David. 1994. "Mental Disorder, Illness, and Biological Disfunction." *Philosophy* 37: 73–82.

Piccinini, Gualtiero. 2015. *Physical Computation: a Mechanistic Account.* Oxford University Press.

Piccinini, Gualtiero and Corey J. Maley. 2014. "The Metaphysics of Mind and the Multiple Sources of Multiple Realizability." In *New Waves in Philosophy of Mind*, edited by Mark Sprevak and Jesper Kallestrup, 125–52. Palgrave Macmillan.

Putnam, Hilary. 1988. *Representation and Reality.* MIT Press.

Schweizer, Paul. 2016. "In What Sense Does the Brain Compute?" In *Computing and Philosophy*, edited by Vincent C. Müller, Vol. 375 of *Synthese Library*, 63–79. Springer.

Searle, John R. 1992. *The Rediscovery of the Mind*. MIT Press.

Shapiro, Lawrence A. 2000. “Multiple Realizations.” *Journal of Philosophy* 97 (12): 635–54.

Wright, Larry. 1973. “Functions.” *The Philosophical Review* 82 (2): 139–68.