

Computers in Abstraction/Representation Theory

Samuel C. Fletcher

*Department of Philosophy
University of Minnesota, Twin Cities*

June 21, 2018

Abstract

Recently, Horsman et al. (2014) have proposed a new framework, Abstraction/Representation (AR) theory, for understanding and evaluating claims about unconventional or non-standard computation. Among its attractive features, the theory in particular implies a novel account of what it means to be a computer. After expounding on this account, I compare it with other accounts of concrete computation, finding that it does not quite fit in the standard categorization: while it is most similar to some semantic accounts, it is not itself a semantic account. Then I evaluate it according to the six desiderata for accounts of concrete computation proposed by Piccinini (2015). Finding that it does not clearly satisfy some of them, I propose a modification, which I call Agential AR theory, that does, yielding an account that could be a serious competitor to other leading account of concrete computation.

References

- Horsman, C., Stepney, S., Wagner, R. C., and Kendon, V. (2014). When does a physical system compute? *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 470(2169).
- Piccinini, G. (2015). *Physical Computation: A Mechanistic Account*. Oxford University Press, Oxford.

Computers in Abstraction/Representation Theory

August 8, 2018

1 Introduction

Abstraction/Representation (AR) theory, recently introduced by Horsman et al. (2014) and developed subsequently by the same authors and collaborators (Horsman, 2015, 2017; Horsman et al., 2017b, 2018), is a general framework for understanding the relationships between physical systems and abstract models of those systems via representation relations. Besides yielding analyses of the similarities and differences between the activities of predictive and experimental science, engineering, and computation, it also provides accounts of their relations to simulation (Horsman et al., 2014), the status of computer science as a natural science (Horsman et al., 2017a) beyond its being a mathematical or engineering discipline (Tedre, 2011), and promises to clarify important concepts such as the distinction between hybrid and heterotic computing (Horsman, 2015). But its main goal has always been to answer the question of when and how a physical system is computing. In contrast to other accounts of concrete computation for which representation is central, the primary intended application of AR theory has not specifically been the human mind, but the assessment of putative unconventional computing devices: hybrid and heterotic computers such as the Galaxy Zoo social machine (Horsman, 2015), in which humans are enlisted as part of a galaxy classifying system (Lintott et al., 2008), biochemical systems such as bacterial chemotaxis and DNA (Horsman et al., 2017b), and non-sentient biological systems such as slime molds (Horsman et al., 2018).

Since AR theory has not heretofore received much sustained philosophical attention as a novel account of concrete computation, I expound it in Section 2, developing some details that have been ambiguous or only implicit in it, focusing on how it answers the question, “What is a computer?” This semi-formal exposition facilitates a clearer organization of AR theory’s conceptual structure besides suggesting further formal development, perhaps along categorical lines (as described in footnotes 1 and 5). One of the additional interesting features of AR theory is that, while it is related to semantic accounts of computation because of its central use of the concept of representation (Piccinini, 2017, §2), it is distinct from them in how it uses that concept. In Section 3 I compare them in more detail to show how AR theory is novel. I then evaluate it with respect to the six desiderata for accounts of concrete computation proposed by Piccinini (2015), finding that it does quite well on many of them, but could be faulted on some: the senses in which it does (not) make computation objective and the way it attempts to avoid collapsing into an uninteresting pan-computationalism. Thus, in Section 5 I propose a modification of AR theory—what I call Agential AR (AAR) theory—that solves these problems. The central way AAR theory modifies AR theory is the titular specification of a special role for agents and their capacities to delimit more tightly and make more objective which physical systems are actually computers. Finally, in Section 6 I raise some further open questions about (A)AR theory. Of especial interest are its pragmatic value as a tool for framing questions and facilitating progress in computer science, and its comparative evaluation with other accounts of computation, such as the mechanistic account (Piccinini, 2015, 2017, §2.5).

Before continuing, a note on my methodology: My goal in general, and in Section 2 in particular, is not finely exegetical—that is, to distinguish interpretively any slight differences in the overlapping formulations of AR theory given over the past few years (Horsman et al., 2014; Horsman, 2015, 2017; Horsman et al., 2017b, 2018). (In fact, I rely primarily on Horsman et al. (2014), Horsman (2015), and Horsman et al. (2018).) One should expect a bit a variation in the formulation of a theoretical framework in its first years of development as its central ideas are fixed by usage and application, and such differences should not necessarily be reified into variations of

doctrine. Rather, my goal is an *explication* of AR theory, more or less in the Carnapian sense, in the service of making it resilient to the kinds of demands philosophers of computation make of theories of concrete computation. Accordingly I allow that there may be different elaborations or explications; what follows is my own.

2 AR Theory and its Computers

AR theory grounds its account of concrete computation in a wider framework for scientific representation and mathematical modeling by abstractions. To do so, it first presupposes a distinction between two domains of objects: the *physical (states)*, \mathbf{P} , whose objects \mathbf{p} are represented notationally with lowercase boldface letters, and the *abstract*, M , whose objects m are represented notationally with lowercase italics letters. Concrete physical object states are also assumed to be related by certain unary functions $\mathbf{P} \rightarrow \mathbf{P}$, the class of which is denoted by \mathbf{F} , such as that which relates an object at a time to the same at another time (i.e., its temporally evolved counterpart). A *representation* or *modeling* relation is then a partial map \mathcal{R} that takes $\mathbf{P} \rightarrow M$ and $\mathbf{F} \rightarrow F$, where F is the class of unary functions on M .¹ (Following standard benign notational abuse, I will write $\mathcal{R}(m)$ and $\mathcal{R}(f)$, with $m \in M$ and $f \in F$, for what are really two distinct relations.) In other words, a representation maps (some) physical object states to abstract objects, and (some) physical function relations to abstract function relations.

Most major proposals for theories of scientific representation are not “radically naturalistic” (Suárez, 2003, p. 225) in the sense that “whether or not representation obtains depends on facts about the world and does not in any way answer to the personal purposes, views or interests of enquirers” (Suárez, 2003, pp. 226–7). In other words, which representation relation (if any) obtains in an actual case depends on the existence of a concrete representational entity \mathbf{e} intending that relation.² Since the representations of interest here relate abstracta with concreta, \mathbf{e} must therefore

¹Note the similarity between a representation and a “partial” functor between categories. Horsman (2015, p. 10) raises the possibility of giving a categorical interpretation of AR theory but does not pursue it. Similarly, I also urge its pursuit, but in future work. (See also footnote 5.)

²Horsman et al. (2014) use the terminology “computational entity,” superseded already in Horsman (2015).

have epistemological or even just pragmatic access to the relevant abstracta. Note however that “AR theory does not require the representational entity to be human, or even conscious” (Horsman et al., 2018, p. 128), only that they be capable of representing (i.e., modeling) concrete objects as abstract ones. The collection of physical object states in the domain of a representation thus does not determine the representation’s range; this is rather provided by a *theory* used explicitly or implicitly by the representational entity, which also specifies a collection of physical object states to which it applies.³ Importantly, different representation relations applied to a physical object state in their common domain may not yield the same abstract object. In other words, different theories can model the same objects in different ways.

Before continuing to how AR theory applies these ideas to concrete computation, it is helpful to discuss their potential presuppositions about ontology and representation. Although the foregoing suggests that AR theory is naturally interpreted as assuming the correctness of both mathematical platonism (Linnebo, 2018) and scientific realism (Chakravartty, 2017), it’s not clear that these are actually necessary presuppositions. One might argue, as nominalists have (Bueno, 2014), that the reference to the distinction between the abstract and concrete domains, *pace* Frege (1953) and Quine (1969), needn’t imply literal commitment to the existence of abstract mathematical objects—ditto for the concrete ontology postulated by the scientific theories one uses, as empiricists urge (Monton and Mohler, 2017). The reference to abstract objects in AR theory might just be taken to be a *façon de parler*, and the physical domain might ultimately concern just physical phenomena (Horsman et al., 2018, p. 132). Similarly, although Horsman et al. (2018, p. 130) seem to endorse the structuralist approach to representation of van Fraassen (2008),⁴ it’s not clear

³Horsman (2015, 2017) and Horsman et al. (2017b) essentially *define* a theory to be a set of representation relations, but this immediately leads to difficulties accounting for how theories can make claims about possible but not actual concrete situations: such situations are simply not in the domain of physical object states. But this problem can be avoided if one simply assumes that, whatever theories are, they *provide* representation relations for a wide range of domains, including possible but not actual physical states.

⁴In fact, it’s not clear that all aspects of AR theory are compatible with the structuralist accounts of scientific representation given by van Fraassen (2008), who takes representations to entail the proposal or assertion of an hypothesis that there is an isomorphic embedding of the abstract model into the concrete target of the representation. For AR theory, a representation relation is a map from the concrete to the abstract: it’s just the wrong kind of relation to be an isomorphic embedding. Perhaps a structuralist account of representation for AR theory would make the representation relation a homomorphism, but I won’t pursue this question here.

that they need to assume anything more about representation than that it allows for surrogate reasoning about concrete objects using abstract objects, a feature all viable accounts of scientific representation share (Frigg and Nguyen, 2016).

Indeed, the relationship between computation, on the one hand, and scientific modeling and engineering, on the other, that AR theory describes rests on how the basis for reasoning using the latter facilitates that for the former. To begin with, consider two concrete objects states, \mathbf{p} and \mathbf{p}' , related by some physical function \mathbf{H} : $\mathbf{H}(\mathbf{p}) = \mathbf{p}'$. \mathbf{H} could represent temporal evolution, so that \mathbf{p}' is just \mathbf{p} at a later time, although this isn't necessary. Whatever \mathbf{H} is, suppose that some theory \mathcal{T} provides a representation relation $\mathcal{R}_{\mathcal{T}}$ that has \mathbf{p} , \mathbf{p}' , and \mathbf{H} in its domain, and that there is a distance function d defined at least on the range of $\mathcal{R}_{\mathcal{T}}$ in M that encodes how relevantly similar two models are. Further, let $\mathcal{R}_{\mathcal{T}}(\mathbf{p}) = m_{\mathbf{p}}$, $\mathcal{R}_{\mathcal{T}}(\mathbf{p}') = m_{\mathbf{p}'}$, $\mathcal{R}_{\mathcal{T}}(\mathbf{H}) = C$, and the relation $m \approx_{\epsilon} m'$ hold for $m, m' \in M$ just when $d(m, m') < \epsilon$. Then \mathcal{T} is said to be ϵ -valid for \mathbf{p} , \mathbf{p}' and \mathbf{H} when

$$\mathcal{R}_{\mathcal{T}}[\mathbf{H}](\mathcal{R}_{\mathcal{T}}(\mathbf{p})) \approx_{\epsilon} \mathcal{R}_{\mathcal{T}}(\mathbf{H}(\mathbf{p})), \quad (1)$$

i.e., when $C(m_{\mathbf{p}}) \approx_{\epsilon} m_{\mathbf{p}'}$. So, supposing that there is some particular ϵ delimiting the range of acceptable error, this just means that the representation and the function \mathbf{H} “approximately commute” with one another.⁵ In particular, if \mathbf{H} represents the dynamical evolution of one object in a certain state to another in a certain state, then a theory of that dynamics and the objects to which it applies is (dropping the ϵ -) valid simpliciter when it can be used to predict with sufficient accuracy the abstract state of the object in question over time as represented in \mathcal{T} .⁶ Determining this for a specific theory is just the process of testing that theory—and that of any instruments used (Horsman et al., 2014, p. 7)—against its predictions within a contextually determined range of error, hence requires that measurements sufficiently determine according to \mathcal{T} the abstract properties needed to

⁵One might describe this with an approximately commuting diagram—see footnote 1 for more on the connection with category theory. Also cf. Corless and Fillion (2014, p. 30), who attribute the idea of an approximately commuting diagram to describe representation in the context of numerical computing to Robidoux (2002, Ch. 6).

⁶One could say in such cases that the function $C_{\mathcal{T}} = \mathcal{R}_{\mathcal{T}}(\mathbf{H})$ is the corresponding “abstract evolution” (Horsman, 2015, p. 4), but this should be understood metaphorically: since abstract objects do not exist in space and time they cannot literally evolve.

make this comparison. How one musters “confidence” (Horsman et al., 2014, p. 21) that a theory is valid for a certain range of objects and functional relations between them is thus a matter for confirmation theory and other criteria for theory selection in science.

Having a valid theory for a domain of object states and functional relations (which one might demand satisfies any other desired criteria besides predictive success) opens up the possibility of using it to control and engineer physical objects according to abstract design specifications. Thus, for a theory \mathcal{T} , one can define the instantiation relation as a partial map $\tilde{\mathcal{R}}_{\mathcal{T}}$ that takes $M \rightarrow \mathbf{P}$ and $F \rightarrow \mathbf{F}$. Then $\tilde{\mathcal{R}}_{\mathcal{T}}(\mathbf{p}') = m_{\mathbf{p}'}$ just when there exists some \mathbf{p} and \mathbf{H} in the domain of validity for \mathcal{T} for which Equation 1 holds. In such a case, the object state \mathbf{p}' has been “engineered” to the specification $m_{\mathbf{p}'}$, within the tolerance of error allowed by the approximation relation. One of the important consequences of basing the engineering specification on a valid theory \mathcal{T} that is predictive and not merely descriptive is that \mathcal{T} allows one to answer counterfactual questions about instantiation: “which physical system, when represented abstractly, would give [under evolution] the abstract representation that we are trying to instantiate?” (Horsman, 2015, p. 6).⁷ Thus, even if one does not in fact engineer the object state \mathbf{p}' according to the foregoing procedure, *it still instantiates the abstract model $m_{\mathbf{p}'}$.*

AR theory describes computing through a particular combination of representation, instantiation, and one further kind of relation: encoding and decoding. Formally, encodings E are just abstract partial functions $M \rightarrow M$ and $F \rightarrow F$, and decodings D the same but just on M . Typically, though, computation is a process by which one attempts to answer a specific question, i.e., compute the value $f(c)$ of a specific abstract function f , which may not occur “naturally” in the abstract models one has for valid theories of controllable physical objects: “there is though no a priori connection between the abstract specification of the problem, c , and the abstract specification of the computer, $m_{\mathbf{p}}$. This connection is to be found in the process of encoding” (Horsman et al., 2018, p. 137). So, suppose again that \mathcal{T} is a valid theory for a domain of object states, including \mathbf{p} and \mathbf{p}' , and functional relations between them, including \mathbf{H} , providing a representation relation

⁷See also Horsman (2017, p. 198).

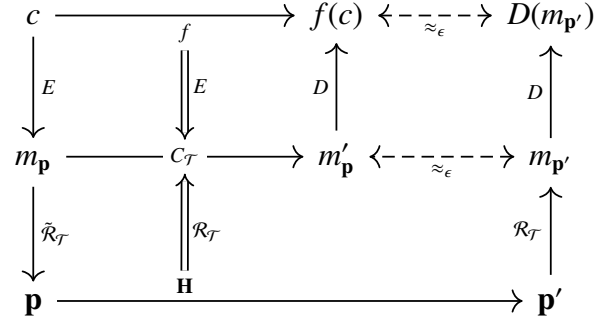


Figure 1: Depiction of a computation according to AR theory. The target output of the computation is the value $f(c)$ of the function f . The encoding E maps the input value c into a mathematical model $m_{\mathbf{p}}$ and the function f into an abstract dynamics on the model $C_{\mathcal{T}}$, representing some physical dynamics \mathbf{H} as described by some theory \mathcal{T} . According to that same theory, $m_{\mathbf{p}}$ instantiates a physical state \mathbf{p} and the physical state $\mathbf{p}' = \mathbf{H}(\mathbf{p})$ is represented by the abstract model $m_{\mathbf{p}'}$. Since the theory \mathcal{T} is ϵ -valid, $m_{\mathbf{p}'} \approx_{\epsilon} m'_{\mathbf{p}} = C_{\mathcal{T}}(m_{\mathbf{p}})$, as indicated by the dashed arrow. The decoding D applied to these models then returns the result of the computation. In general the computed result approximates the value $f(c)$, but this can often be made exact when f is a discrete function.

$\mathcal{R}_{\mathcal{T}}$ such that $\mathcal{R}_{\mathcal{T}}(\mathbf{p}) = m_{\mathbf{p}}$, $\mathcal{R}_{\mathcal{T}}(\mathbf{p}') = m_{\mathbf{p}'}$, and $\mathcal{R}_{\mathcal{T}}(\mathbf{H}) = C$. Then, for \mathcal{T} , an encoding/decoding pair for a function f with c in its domain satisfies $E(c) = m_{\mathbf{p}}$, $E(f) = C$, and $D(C(m_{\mathbf{p}})) = f(c)$. In a word, the encoding translates the input of the function and the function itself into the models of the physical object states and their relations, while the decoding translates back the value of the function. Often D can be the (perhaps weak) inverse of E in its action on M —this is when both c and c' are embedded in the same mathematical way in the models of the physical object states—but this is not at all necessary. In practice, the process of encoding can be concatenated in many levels, in which an abstract problem is encoded sequentially into models closer and closer to that used by a theory to represent a physical object, and similarly for decoding. But mathematically, the composition of these levels of encoding and decoding can be treated as a single operation.

Now all the components of AR theory's account of concrete computation are in place. Suppose once again that the assumptions of the previous paragraph regarding the valid theory \mathcal{T} and its domain are in place, as well as the existence of the encoding E and decoding D for a specific function evaluation $f(c)$ mentioned there. Further suppose that the same representational entity \mathbf{e} using \mathcal{T} is able to instantiate $m_{\mathbf{p}}$ in \mathbf{p} , i.e., $\tilde{\mathcal{R}}_{\mathcal{T}}(\mathbf{p}) = m_{\mathbf{p}}$. Then \mathbf{e} computes $f(c)$ using \mathbf{p} and \mathbf{p}' just

when \mathbf{e} actually performs this instantiation, measures \mathbf{p}' to determine $m_{\mathbf{p}'} \approx_{\epsilon} C(m_{\mathbf{p}})$ for sufficiently small ϵ , and decodes the result $D(m_{\mathbf{p}'}) \approx_{\epsilon} f(c) = D(C(m_{\mathbf{p}}))$, as depicted in Figure 1. (When the function computed is discrete, one can often demand that $m'_{\mathbf{p}} \approx_{\epsilon} m_{\mathbf{p}'}$ entail that $D(m_{\mathbf{p}'}) = f(c)$.) In this case, the *computer* is \mathbf{p} and \mathbf{p}' —typically the same object in different states at different times. In a word, “physical computing is the use of a physical system to predict the outcome of an abstract evolution” (Horsman et al., 2014, p. 1), i.e., the evaluation of a function. It is important that \mathbf{e} is *actually* representing, encoding, and decoding in order for concrete computation to occur: even if some \mathbf{e} in some sense could have done so, no computation has occurred if it did not. The representational entity selects which representation relation, encoding, and decoding are used, if any. Any physical object “may potentially be a computer, but without an encode and a decode step [with representation] it is just a physical system” (Horsman et al., 2014, p. 15).

3 AR Theory as an Account of Concrete Computation

AR theory notably does not fit within the usual division of positions about concrete computation into mapping, counterfactual, syntactic, mechanistic, and semantic accounts (Piccinini, 2017, §2). The mapping account, arising from a sketch by Putnam (1960), essentially takes any physical system that can be accurately described in computational terms to be a computer, i.e., performing a concrete computation. The best version of the mapping account, according to Piccinini (2015, pp. 17–18), states that

a physical system S performs computation C just in case:

- (i) there is a mapping from [a subset of] the states ascribed to S by a physical description to the states defined by computational description C , such that
- (ii) the state transitions between the physical states mirror the state transitions between the computational states.

Clause (ii) requires that for any computational state transition of the form $s_1 \rightarrow s_2$

(specified by the computational description C), if the system is in the physical state that maps onto s_1 , it then goes into the physical state that maps onto s_2 .

AR theory and the mapping account are not notably similar beyond their shared idea that computation involves physical and computational descriptions of some sort. In the mapping account, there is no invocation of anything analogous to representational entities or the role of their theories, for example.

Counterfactual accounts, and the related causal and dispositional accounts, augment clause (ii) in the mapping account with counterfactual, causal, or dispositional conditions, respectively. Since AR theory invokes neither causes nor dispositions, I will set those aside to focus on its comparison with counterfactual accounts. In such accounts, the gloss of clause (ii) by a material conditional is replaced by a subjunctive conditional: for any computational state transition of the form $s_1 \rightarrow s_2$ (specified by the computational description C), if the system were to be in the physical state that maps onto s_1 , it then would go into the physical state that maps onto s_2 . (See, e.g., Maudlin (1989, p. 415), Chalmers (1996, p. 312), and Copeland (1996, p. 341) for formulations along these lines.) This is closer to AR theory than the mapping account because AR theory also demands that, in order for physical computation to occur, the theory being used to represent the physical states must support counterfactual relations between states and their representing models. But, like with the mapping account, there again is, for example, no invocation of anything analogous to representational entities or the role of their theories.

Syntactic accounts (Fodor, 1975) restrict the acceptable mappings of the mapping account further than the counterfactual account by demanding that “only physical states that qualify as syntactic may be mapped onto computational descriptions, thereby qualifying as computational states” (Piccinini, 2015, pp. 44–5). By contrast, AR theory places no such restriction on the form of a theory’s abstract representation of a physical state. Indeed, part of the motivation for AR theory was to provide a framework for understanding for unconventional “new types of computing [that] are being developed which again do not sit cleanly with what has become the standard model of computing” (Horsman, 2017, p. 194) using Turing machines, the lambda calculus, etc.

The relatively new mechanistic account of concrete computation (Piccinini, 2007, 2015) describes a physical computing system as

a mechanism whose teleological function is ...the manipulation ...of a medium-independent vehicle according to a rule. A medium-independent vehicle is a physical variable defined solely in terms of its degrees of freedom ... as opposed to its specific composition. A rule is a mapping from input and/or internal states to internal states and/or outputs. (Piccinini, 2015, p. 10)

While AR theory shares with the mechanistic account an emphasis on “medium-independent vehicles”—it is irrelevant to the theory what particular physical system the abstract models in a computation represent—it does not demand that a physical computer be a mechanism with a teleological function. Indeed, even though most computers are designed with such a teleological function—namely, the one Piccinini describes—AR theory as mentioned before explicitly allows for the possibility of computers that were not designed as such. More importantly, however, the mechanistic account denies that “computation presupposes representation” (Piccinini, 2015, p. 3). By contrast, representation is key in AR theory: for example, Horsman (2015, fn. 3) explicitly endorses the dictum that there is “no computation without representation” (Fodor, 1981, p. 180).

Semantic accounts of computation share this commitment. They take inspiration from the practice of using computation to process meaningful information. As Piccinini (2017, §2.3) explains:

The semantic account of computation turns this practice into a metaphysical doctrine: computation is the processing of representations—or at least, the processing of appropriate representations in appropriate ways. ... Only physical states that qualify as representations may be mapped onto computational descriptions, thereby qualifying as computational states. If a state is not representational, it is not computational either.

Moreover, consider for example a pocket calculator being used to crunch some numbers. According to semantic accounts, its “computational states have their content essentially. If the calculator’s states represented something *other than* those numbers, they would be *different* computational

states. If the calculator's states did *not* represent *at all*, they *wouldn't* be computational states” (Piccinini, 2015, p. 26).

Construed appropriately, AR theory also takes computational states, such as c and c' discussed in Section 2 above, to be individuated essentially, but only by their *mathematical* content—that is, by whatever identity conditions are relevant for the type of abstract mathematical object they are (cf. Egan, 1999). Furthermore, without an encoding and decoding, the physical states of the putative computer, as represented by abstract models, would not be computational at all. A similar conclusion holds for computational processes, which are individuated by the mathematical function they implement, rather than any cognitive or external semantic meaning that function may have; without an encoding of the function into the physical relation or process used in the putative computation, there would be no computation at all. Whatever other semantic content (meaning) the input and output of the computation and the function computed have for the representational entity, if any—e.g., a certain division as the dollars owed by everyone in a party at a restaurant, or a particular minimization as the length of the shortest walking path in a city between two monuments—is irrelevant for its status as a computation. So, in this sense, a semantic account of computation, “as closely allied with cognition and a philosophy of logic and language, is the opposite of our starting-point for AR theory: a computation is abstract; a computer is physical” (Horsman et al., 2018, p. 131).

Another similarity between semantic accounts and AR theory is that only abstract models, and the physical systems they *represent*, can be a component of being a computation or computer, respectively: inputs to computation can be encoded only into models that instantiate in physical systems. In other words, without an abstract model being representational, and without a physical system being represented, computation cannot take place. But this does not entail for AR theory, as detailed in Section 2, that computation should be *identified* as the manipulation or processing of certain representations in certain ways. Some physical computations may involve such manipulations or processing, but this is not the case in general: all that is required is “the use of a computer (a physical system) to predict the outcome of a computation (an abstract evolution [i.e., the eval-

uation of a function]) through a compute cycle” (Horsman, 2015, p. 8), namely, the encoding, instantiation, and decoding of a computational problem by a representational entity according to some theory.

In sum, AR theory shares with semantic accounts of concrete computation a commitment to the centrality and indispensability of representation, but does not demand that this notion of representation “come pre-loaded with implications of intentionality or meaning; it is, at its most fundamental, a mapping from physical to abstract” (Horsman et al., 2018, p. 130).⁸ This is not to say that representational entities cannot have intentionality in their representations or that what they compute must be devoid of meaning; rather, whatever intention the representational entity has *beyond* the mapping from physical system to abstract mathematical model, and whatever meaning the representational entity attributes to the computation and its states *beyond* its mathematical structure, has no bearing on the existence and identity conditions for concrete computations.

4 The Evaluation of AR Theory along Six Desiderata

Even though AR theory does not fit neatly within the taxonomy of accounts of concrete computation provided by Piccinini (2015, 2017), one can still evaluate it according to the same sorts of desiderata (Piccinini, 2015, p. 11):⁹

An account of concrete computation that does justice to the science of computation should have the following features:

1. objectivity;
2. explanation;
3. the right things compute;

⁸They continue, “We almost never talk about ‘information’ or ‘knowledge’ or ‘meaning’ in using AR theory” but this seems to be an overstatement on at least one count: “information” is used in the context of discussing AR theory on several occasions in the same paper (Horsman et al., 2018, pp. 138, 142, 148) and elsewhere. Perhaps it is best to interpret these positive usages as information in the mathematical sense (Shannon and Weaver, 1949), devoid of semantic content.

⁹In the interests of facilitating reference throughout this section, I have separated the desiderata by line breaks in the following quote instead of listing them in-line.

4. the wrong things don't compute;
5. miscomputation is explained; and
6. an adequate taxonomy of computing systems is provided.

Indeed, many of these are also goals of the authors of AR theory: Horsman et al. (2014, pp. 15–16) take the objectivity of computation (desideratum 1) as an explicit goal, as is an account of computing allowing for the classification of non-standard or unconventional computers (desiderata 3 and 6) in addition to conventional ones while avoiding the endorsement of pancomputationalism (desideratum 4) (Horsman et al., 2014, p. 2)—see also Horsman (2015, p. 2) and Horsman et al. (2018, pp. 138–40, 145–6). Although these authors do not explicitly discuss desiderata 2 and 5, AR theory does quite well on them. I will discuss these briefly, as well as desideratum 6, before continuing to desiderata 1, 3, and 4, which require a longer discussion. I will argue that AR theory could be improved along the latter three, which will lead to Agential AR theory in Section 5.

To begin with desiderata 2 and 6: a good theory of concrete computation should both help explain the capacities, features, and limitations of particular computers, and provide the means for classifying them along these lines. AR theory allows one to do just this insofar as permits one to use the best scientific theories of the computers themselves to understand and classify particular kinds of computers. One can gain understanding and explanation of their types and capacities precisely because good scientific theories provide such understanding and explanation; the only additional step one must take is to interface the models of these theories with the mathematical models for computation via the process of encoding and decoding. By no means does AR theory make this routine or automatic, but it does show how it is possible, even practically.

Desideratum 5 demands that a good theory of concrete computation allow for a concrete computation to go wrong and explain how. For example, a concrete computation of $f(c)$ could yield a value c' not equal (or otherwise relevantly approximate) to it, could return an abstract model not in the domain of the decoding function, or even yield a physical system not within the domain of the theory \mathcal{T} being used. Just as with desiderata 2 and 6, AR theory succeeds in meeting this demand because it shows that the success of a concrete computation depends on the representational entity

using a valid theory, so miscomputation is a sign of either invalidity or misspecification. Moreover, AR theory indicates clearly several different aspects of validity that could fail: a valid theory may be inadvertently extended beyond its range of validity, for example, as often happens when a conventional silicon computer overheats, or the confidence in the validity of the theory may be unwarranted. In any case, the use of the physical computer as a means to predict must meet the relevant counterfactual conditions that the scientific theory provides.

The demand to make concrete computation an objective matter is, at least in part, to make it compatible with ideas about scientific objectivity and applicable to the sciences that use computational concepts—in a word, to make claims about concrete computation responsive to empirical evidence (Piccinini, 2015, pp. 11–12). Of course, there are many different senses in which science might be or aim at being objective (Reiss and Sprenger, 2017), the two most relevant of which here are *mind-independence* and *freedom from bias*.¹⁰ Mind-independence is the type of objectivity that demands that the products of scientific research do not essentially depend on our human perspective on the world. Freedom from bias, meanwhile, is a sort of intersubjectivity of the scientific process to correct the idiosyncratic perceptions and tendencies of individual scientists, typically through the use of measurement and quantification. Piccinini (2015, p. 12) seems to have in mind the latter, freedom from bias, as he requires that “When [scientists using computational descriptions] disagree, they address their opponents by mustering empirical evidence about the systems they study.” Horsman et al. (2014, p. 16) seem to have both senses in mind when they remark that whether something is a computing system “is not something that is subjective or a matter of opinion: it is a matter of fact about which hypotheses can be formed and tested” and that, regarding a putative case, the fact “that argument was required to settle the matter does not make it subjective”; but they certainly intend the mind-independence sense when they caution,

If we are now saying that computational processes cannot be described independently of computational entities (human or otherwise), then an immediate concern is that the act of computations then becomes wholly subjective, possibly subjected to the intent

¹⁰A third sense is the so-called “value-free ideal” of having no social, moral, or political value impinge on scientific method or product.

of the entity running the computer, and not something that can be dealt with by an objective scientific theory of computation. (Horsman et al., 2014, p. 15)

To address this concern, they state that “It is simply an objective fact of the matter whether or not a computational entity is part of the system. ... [I.e.,] the requirement is that an encoding and a decoding are present, an objective fact of the matter” (Horsman et al., 2014, p. 15). They give as an example a student working at a calculator: “By close observation of the student, you can determine whether information is being encoded into and decoded from the calculator. You as the observer can formulate and test the hypothesis that the student and calculator form a computing system” (Horsman et al., 2014, p. 15). However, it’s not at all clear just what the objective criteria for “observing” or “testing” for an encoding or decoding are in the general case—these are abstract processes, after all, in AR theory—and the example of the student depends on our plausible imputation of intent to the student’s behavior. Horsman et al. (2017b, p. 105) write that “if a certain process can or could occur in multiple different ways, all of which instantiate a single abstract object or evolution, then this is the signature that it is the abstract and not the physical operation that is important,” which is a sign of (but not sufficient for) representation. But important for what (or for whom)? And what kinds of multiply realized processes count? Despite statements to the contrary, methods in general by which researchers could come to intersubjective agreement about whether an object is a representational entity and what theory it is using, independently of any attributions by human observers, are yet forthcoming. This is the *objectivity problem for representational entities*.

Another, more localized but still important problem for AR theory to realize the objectivity desideratum comes from the definition of (ϵ -)validity for theories, which requires that the theoretically predicted model only be *approximately* the same as the measured one (equation 1). The conflict for objectivity is that, as things currently stand, there seems to be no fact of the matter regarding what the relevant notion of approximation should be—i.e., how to determine the distance function d on abstract models—and even after this is determined, how close the approximation should be in order to count a theory as valid. Another conflict for AR theory as currently de-

veloped is that the representational entity is required to have “confidence” (Horsman et al., 2014, p. 21) in the validity of the theory, or that the theory be “well understood” (Horsman, 2015, p. 7), especially as it applies to the physical functional relations in its domain (Horsman et al., 2018, p. 134), but it is not clear that these concepts are applicable to general representational entities; hence, what objectively determines whether they obtain? This is the *objectivity problem for the validity of theories*.

I will propose a slight modification to AR theory in Section 5 to solve the objectivity problems for computational entities and theory validity, but first I will evaluate AR theory along desiderata 3 and 4. These two require a theory of concrete computation to rule correctly on intuitively evaluable cases, both of computation and non-computation, respectively. Historically, accounts of concrete computation have not had trouble describing what are intuitively computers as such, but have had more trouble ruling out non-computers. In particular, mapping, counterfactual, and some semantic accounts have entailed a version of pancomputationalism, the thesis that all physical systems are computers (Piccinini, 2017, §3). *Unlimited* pancomputationalism holds in particular that each physical system performs each possible computation, or at least any sufficiently complex physical system performs a large number of inequivalent computations. Its weaker version, *limited* pancomputationalism, maintains only that each physical system performs at least one computation.

Now, as an account designed to accommodate both conventional and unconventional computers, AR theory seems to be entirely successful in describing classical digital computers, the Babbage difference engine, even slime mold maze solvers as computers (Horsman et al., 2018, §3).¹¹ Additionally, Horsman et al. (2014, p. 2) were originally motivated on practical grounds to provide an account of concrete computation that rules out pancomputationalism—any account that allows it, they contend, will not be of much use to science, and this is surely correct (at least for unlimited versions). But it is not yet clear how AR theory on its own rules it out, even the unlimited version, in part for reasons related to the objectivity problem for representational entities.

To see how this problem arises, consider any physical system state \mathbf{p} as it arises within a

¹¹Horsman et al. (2017b) also do so for chemotaxis in bacteria and DNA, but these cases less plausibly fall under the category of intuitively evaluable computations.

sufficiently large (i.e., infinite) state space and suppose that it is its own representational entity. Given any theory valid for the states of this system and its dynamics \mathbf{H} ¹² and some computational problem—i.e., a function f to evaluate—there will be some encoding E that maps the modeled dynamics C to the function f , an initial state $m_{\mathbf{p}} = \mathcal{R}_{\mathcal{T}}(\mathbf{p})$ to any desired input c to the function f , and hence the final state $m_{\mathbf{p}'} = \mathbf{H}(\mathbf{p})$ to the desired output $f(c)$. Thus, without clear restrictions on what can count as a representational entity—that is, what physical objects can represent with theories and encode and decode—just about anything (sufficiently complex) could be computing anything.¹³ This is the same pancomputational problem afflicting semantic accounts when they put no restriction on what can be representing (Shagrir, 2006, p. 396).

Now, Horsman et al. (2018, p. 145) respond to this problem by reference to Horsman et al. (2017b), who they claim show that representation “is a highly non-trivial process requiring the active and explicit use of representation intrinsic to a system’s processes.” Horsman et al. (2017b, §4) analyze cases of putative computation in biological systems, arguing that representation and signaling—a trivial computation—occur in chemotaxis for bacteria and in DNA transcription. Suppose one accepts these analyses; it’s then true that these examples are non-trivial, but why should all examples be so in order to be legitimate? There’s nothing in AR theory itself which establishes what counts as “non-trivial,” “active,” or “explicit” representation, much less why these features should be at all relevant for restricting what can count as a representation or representational entity. This is the *pancomputationalist problem for representational entities*.

AR theory faces another pancomputationalist problem, implicit in the above scenario, even if that for representational entities is solved. For suppose some restriction is placed on which representations and representational entities are legitimate, and that some person—clearly a viable representational entity—uses a theory to represent legitimately some physical object, say, a rock.

¹²There is always some valid theory for any domain, namely, the one whose representation maps each physical system to the same abstract object—this is the *constant* representation. However, we must assume here some theory whose representation relation has a range with an infinite cardinality. Unless the physical domain \mathbf{P} is much simpler than it appears, this will be easily satisfied.

¹³Another option would be to restrict the admissible representations or encodings directly, although to avoid being ad hoc this would move AR theory in the direction of a syntactic account of concrete computation. I won’t pursue that option here, since it is inimical to the motivations for AR theory for understanding unconventional computation, as discussed in Section 3.

Again, if its state space is sufficiently large, then there will be some encoding of an arbitrary problem into the abstract model for the physical object and its evolution. At the very least, with the identity encoding they can compute $E(C) = C$, and with the constant encoding they can compute the identity function $E(C) = \text{id}$. Indeed, why cannot the encoding and decoding simply be composed with the representation to provide a new representation? This is the *pancomputationalist problem for encoding/decoding*.

Horsman et al. (2018, p. 146) have also responded to this sort of issue: “All the ‘computation’ of establishing the mapping from rock states to table entries [of values of f] is being done in the representation stage: the rock *itself* has computed none of this.” This is certainly true, intuitively, but there is no restriction on what theories, representations, and encodings and decodings can be used in AR theory as currently formulated. One might be tempted to say that the computational work done constructing the table entries should be included as a necessary part of the computation, but this quickly leads to a regress: should one include the computational work needed to construct the computer used to determine the table entries? Just how “much” computation must have occurred to make inclusion necessary? Even if this response were a tenable solution to the pancomputationalist problem for encoding/decoding, it would generate a new problem for desideratum 3, for what one thought were *standard* computers are *not* because one has left out all the computers used to design them, etc.

5 Agential Abstraction/Representation Theory

In Section 4, I showed that AR theory does well along four of the six desiderata for accounts of concrete computation proposed by Piccinini (2015, p. 11). As it is currently formulated, however, it has problems with desideratum 1, objectivity of computation, and desideratum 4, that the wrong things don’t compute. There were in fact two distinct problems for each: respectively, the objectivity problems for representational entities and the validity of theories, and the pancomputationalist problems for representational entities and encoding/decoding. However, as I will argue in

this section, these problems can be overcome by some modifications to AR theory, in particular an appropriate restriction on representational entities and how their capacities relate to representation and encoding. Such representational entities can be called *agents*, and the result, Agential AR (AAR) theory.

In their discussion of how to characterize a physical process as information processing, Maroney and Timpson (2018, p. 124) emphasize that

the question of how information-processing tasks can be physically instantiated cannot be separated from the question of the existence of physically embodied agents who are informing and who might be being informed by the task. ... It is their physical attributes and abilities that ultimately fix the starting and ending points of the process.

Although my goal here is not a characterization of information-processing tasks per se, this is still a valuable insight that applies *mutatis mutandis* to computation. Since the input and output of a computation are abstract objects, any agent qua representational entity can be characterized as

- having (in principle) access to or understanding of these abstracta, including encodings, decodings, and representations,
- being able to observe or measure the outcome of a physical evolution of a putative concrete computation to compare its result with that abstractly predicted, and
- having the ability (in principle) to gain evidence, hence confidence, about the validity of a theory concerning the objects involved in the putative concrete computation.

Certainly humans can thus be agents, while equally certainly rocks cannot. Although we are not certain about exactly which objects have these capabilities—i.e., are agents—the investigation of this boundary is part of the subject of cognitive science, including animal cognition and artificial intelligence.¹⁴ Thus, although there may not be agreement regarding which objects have

¹⁴I am writing as if there could be only one agential community—a group of objects considered as agents simpliciter—presumably the one in which we are included. There could be, however, as many distinct such communities as there are groups with different shared capabilities for abstraction and representation, observation and

these properties, there is some confidence that over time intersubjective agreement will form—thus which objects are agents is objective in the sense of being free from bias.¹⁵ Further, although having the capacity to be an agent may involve having a mind (of a sort), this is a property that does not depend on any individual’s opinion. Thus, this characterization of representational entities solves their objectivity problem.

I also claim that the same characterization solves the pancomputationalist problems for representational entities and encoding/decoding. Regarding the former, an arbitrary physical object just won’t have the right capacities to be an agent, so not all objects compute in virtue of representing themselves as such. Regarding the latter, the only encodings, decodings, and representations that can be used in a legitimate concrete computation are those that are directly accessible, or comprehensible, to the agent. This rules out the many arbitrary encodings and decodings, as well as ones that, though not arbitrary, are too complicated to be immediately accessed, and similarly for representations. Moreover, the agent must be able to observe or measure the physical object used as a computer, and actually do so for a concrete computation to occur. Now, it’s true that most users of modern computers don’t actually have accessible abstract representations of them as physical devices, nor for the encodings they effect when they alter the device’s state by typing in instructions through a keyboard. But representations of all of these have been constructed by the designers of the computer, who are part of our epistemic community (Maroney and Timpson, 2018, §6.1), so in principle these representations are available to human users. By contrast, if one encountered an alien putative computer, one would have to develop a theory about its behavior in enough circumstances before one could actually use it in a concrete computation. Thus, the above characterization of representational entities as agents and the encodings, decodings, and representations they allow also seems to solve their pancomputationalist problem.

prediction, and theorizing and confirmation. Thus one can think about agential communities as a sort of epistemic community (van Fraassen, 1980, pp. 18–9), of which there can be many and which can in principle change over time (van Fraassen, 2005). But delineating the extent and multitude of these boundaries is not necessary for AAR theory: it is always applied relative to some such community, of which we can at least identify a prototypical or otherwise representative member.

¹⁵This would follow from one view of cognitive science’s status as a science. Failing that, however, one could always conservatively restrict to humans.

The only problem that remains is that of the objectivity of the (ϵ -)validity of theories. Recall that this problem was to find an objective way of characterizing how abstract models should approximate one another, and what that degree of approximation should be, in order that a theory be counted as valid, and how a representational entity \mathbf{e} can have confidence in that validity. Regarding the latter part of the problem, agents are just the sort of objects that can gain evidence, hence confidence, about the validity of a theory, whose relevant objectivity is underlaid by the science and investigation used to produce that evidence. Regarding the former part, one should understand the distance function $d_{\mathbf{e}}$ (defined at least on the range of $\mathcal{R}_{\mathcal{T}}$ in M and the domain and range of functions f to compute) as formalizing how relevantly similar two models are *according to the purposes and interests of \mathbf{e}* . These purposes and interests are contextual to encodings E and decodings D actually used (or at least understandable) by \mathbf{e} , and functions f that \mathbf{e} intends to compute. Relative to these, one can simply say that theory \mathcal{T} is ϵ -valid for a range of physical objects \mathbf{p} and functional relations among them \mathbf{H} just when in fact for every c in the domain of f ,

$$D(\mathcal{R}_{\mathcal{T}}(\mathbf{H}(\tilde{\mathcal{R}}_{\mathcal{T}}(E(c)))))) \approx_{\epsilon} f(c), \quad (2)$$

i.e., the process of encoding c into a model, instantiating that model in a physical system, letting the system evolve (say), then measuring it to determine its representing model and decoding this result is approximately the same as computing the value of the function $f(c)$. When an *equality* is achieved, validity (simpliciter) obtains, but otherwise the relevant threshold ϵ for which the approximation is good enough is determined again according to the purposes and interests of \mathbf{e} .

This answer might at first seem circular: how can one define validity using instantiation if the latter requires a valid theory? It is not circular because an agent can apply the representations of a theory and attempt to instantiate an abstract model in a physical system without the secure grasp that the putative computation will succeed. In this sense, unsuccessful computations are in fact non-computations. Validity in this sense is established in practice for lots of very simple functions, which are then composed together in the manner of recursion theory to allow for the computation

of more complicated functions. Thus, on the picture provided by AAR theory, the modularity of computation is more of a practical demand than an essential feature.

6 Conclusions and Further Questions

While the problem of the objectivity of the validity of theories was solved largely by filling in details perhaps compatible with AR theory, the other three problems—of the objectivity of representational entities, and both pancomputationalist ones—were solved only by explicitly restricting representational entities to agents, and encoding, decoding, and representation to those intelligible by agents. One of the possible costs of this transition to AAR theory is that it is less clear how to vindicate biological systems such as bacteria and DNA as computers, with themselves as representational entities (agents), contra Horsman et al. (2017b). However, perhaps this just shows that bacteria and DNA really aren't by and for themselves computers at all. AAR can at least explain how we can find it fruitful to describe the physical behavior of such systems (and others) in terms of computation: although we, as agents, are the representational entities, the theories we develop to predict their behavior can be framed in a simple computational language that, if they were accessible to the biological systems, would allow them to be agents, too, and compute for themselves.

Solving these problems, though, comes with more direct benefits to theories of computation. Within AAR theory, it allows one to pursue further work connecting it with mathematical theories of computation and with computational complexity theory. Regarding the former, one has with AAR theory a better grip on which mathematical theories of computation can be considered more “physics-like” by understanding how they require simpler embeddings into the models of the physical computer (Sutner, 2018); when one considers a variety of theories besides classical mechanics, perhaps some mathematical theories will be more like specific physical theories than others. Regarding the latter, having an objective theory of when physical systems are computing opens the possibility for an objective and quantitative theory of physical computational complex-

ity and information processing, i.e., the physical “costs” of performing computation or processing information (Maroney and Timpson, 2018).

Another line of investigation concerns a deeper evaluation and appreciations of AAR theory within the philosophical debates concerning the proper account of concrete computation, especially since I have shown (in Section 3) how it does not fit neatly into the standard categorization of these accounts. Given its similarities and differences with semantic accounts discussed in Section 3, it deserves a more direct evaluation of whether it avoids the principal criticisms of these accounts (Piccinini, 2015, Ch. 3), as well as a thorough comparison with mechanistic accounts of computation for their comparative strengths and weaknesses. Since AAR theory seems to satisfy the desiderata for a concrete account of computation given in Section 4, it would be interesting to investigate on what other grounds it could be normatively compared with the mechanistic account.

References

- Bueno, O. (2014). Nominalism in the philosophy of mathematics. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2014 edition.
- Chakravartty, A. (2017). Scientific realism. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2017 edition.
- Chalmers, D. J. (1996). Does a rock implement every finite-state automaton? *Synthese*, 108(3):310–333.
- Copeland, B. J. (1996). What is computation? *Synthese*, 108(3):335–359.
- Corless, R. and Fillion, N. (2014). *A Graduate Introduction to Numerical Methods*. Springer, New York.
- Egan, F. (1999). In defence of narrow mindedness. *Mind and Language*, 14(2):177–94.

- Fodor, J. A. (1975). *The Language of Thought*. Harvard University Press, Cambridge, MA.
- Fodor, J. A. (1981). The mind-body problem. *Scientific American*, 244:114–125.
- Frege, G. (1953). *Foundations of Arithmetic*. Blackwell, Oxford. Trans. J.L. Austin.
- Frigg, R. and Nguyen, J. (2016). Scientific representation. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2016 edition.
- Horsman, C., Stepney, S., Wagner, R. C., and Kendon, V. (2014). When does a physical system compute? *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 470(2169).
- Horsman, D. (2017). The representation of computation in physical systems. In Massimi, M., Romeijn, J.-W., and Schurz, G., editors, *EPSA15 Selected Papers*, pages 191–204, Cham. Springer International Publishing.
- Horsman, D., Kendon, V., and Stepney, S. (2017a). The natural science of computing. *Commun. ACM*, 60(8):31–34.
- Horsman, D., Kendon, V., and Stepney, S. (2018). Abstraction/representation theory and the natural science of computation. In Cuffaro, M. E. and Fletcher, S. C., editors, *Physical Perspectives on Computation, Computational Perspectives on Physics*, pages 127–149. Cambridge University Press, Cambridge.
- Horsman, D., Kendon, V., Stepney, S., and Young, J. P. W. (2017b). Abstraction and representation in living organisms: When does a biological system compute? In Dodig-Crnkovic, G. and Giovagnoli, R., editors, *Representation and Reality in Humans, Other Living Organisms and Intelligent Machines*, pages 91–116. Springer International Publishing, Cham.
- Horsman, D. C. (2015). Abstraction/representation theory for heterotic physical computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 373(2046).

- Linnebo, Ø. (2018). Platonism in the philosophy of mathematics. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2018 edition.
- Lintott, C. J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M. J., Nichol, R. C., Szalay, A., Andreescu, D., Murray, P., and Vandenberg, J. (2008). Galaxy zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, 389(3):1179–1189.
- Maroney, O. J. E. and Timpson, C. G. (2018). How is there a physics of information? On characterizing physical evolution as information processing. In Cuffaro, M. E. and Fletcher, S. C., editors, *Physical Perspectives on Computation, Computational Perspectives on Physics*, pages 103–126. Cambridge University Press, Cambridge.
- Maudlin, T. (1989). Computation and consciousness. *Journal of Philosophy*, 86(8):407–432.
- Monton, B. and Mohler, C. (2017). Constructive empiricism. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2017 edition.
- Piccinini, G. (2007). Computing mechanisms. *Philosophy of Science*, 74(4):501–526.
- Piccinini, G. (2015). *Physical Computation: A Mechanistic Account*. Oxford University Press, Oxford.
- Piccinini, G. (2017). Computation in physical systems. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2017 edition.
- Putnam, H. (1960). Minds and machines. In Hook, S., editor, *Dimensions of Mind: A Symposium*, pages 138–164. Collier, New York.

- Quine, W. (1969). Existence and quantification. In *Ontological Relativity and Other Essays*, pages 91–113. Columbia University Press, New York.
- Reiss, J. and Sprenger, J. (2017). Scientific objectivity. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2017 edition.
- Robidoux, N. (2002). *Numerical solution of the steady diffusion equation with discontinuous coefficients*. PhD thesis, University of New Mexico.
- Shagrir, O. (2006). Why we view the brain as a computer. *Synthese*, 153(3):393–416.
- Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, IL.
- Suárez, M. (2003). Scientific representation: Against similarity and isomorphism. *International Studies in the Philosophy of Science*, 17(3):225–44.
- Sutner, K. (2018). Physics-like models of computation. In Cuffaro, M. E. and Fletcher, S. C., editors, *Physical Perspectives on Computation, Computational Perspectives on Physics*, pages 151–171. Cambridge University Press, Cambridge.
- Tedre, M. (2011). Computing as a science: A survey of competing viewpoints. *Minds and Machines*, 21(3):361–387.
- van Fraassen, B. C. (1980). *The Scientific Image*. Oxford University Press, Oxford.
- van Fraassen, B. C. (2005). The day of the dolphins: Puzzling over epistemic partnership. In Irvine, A. and Peacock, K., editors, *Mistakes of Reason: Essays in Honour of John Woods*, pages 111–133. University of Toronto Press, Toronto.
- van Fraassen, B. C. (2008). *Scientific Representation: Paradoxes of Perspective*. Oxford University Press, Oxford.