Levels of Computational Explanation

Michael Rescorla

Abstract: It is widely agreed that one can fruitfully describe a computing system at various levels. Discussion typically centers on three levels: *the representational level*, *the syntactic level*, and *the hardware level*. I will argue that the three-level picture works well for *artificial* computing systems (i.e. computing systems designed and built by intelligent agents) but less well for *natural* computing systems (i.e. computing systems that arise in nature without design or construction by intelligent agents). Philosophers and cognitive scientists have been too hasty to extrapolate lessons drawn from artificial computation to the much different case of natural computation.

§1. Representation, syntax, and hardware

It is widely agreed that one can fruitfully describe a computing system at various levels. Discussion typically centers on three levels that I will call *the representational level*, *the syntactic level*, and *the hardware level*. To illustrate, consider a computer programmed to perform elementary arithmetical operations such as addition, multiplication, and division:

- At the *representational level*, we individuate computational states through their representational properties. For instance, we might say that our computer divides the number 2 into the number 5 to yield remainder 1. This description implicitly presupposes that the computer's states represent specific numbers.

- At the *syntactic level*, we individuate computational states non-representationally. We describe our computer as manipulating *numerals*, rather than performing arithmetical operations over *numbers*. For example, we might say that the computer performs certain syntactic operations over the numerals "2" and "5" and then outputs the numeral "1." When offering this description, we do not presuppose that the computer's states represent numbers.
- At the *hardware level*, we describe the physical realization of computational states. We specify our computer's components, how those components are assembled, and how the computer's physical state evolves according to well-understood physical laws.

A three-level picture along these lines figures prominently in many philosophical and scientific discussions (Chalmers, 2011; 2012), (Fodor, 1981; 1987; 1994; 2008), (Haugeland, 1985), (Pylyshyn, 1984).

I will argue that the three-level picture works well for *artificial* computing systems (i.e. computing systems *designed* and *built* by intelligent agents) but less well for *natural* computing systems (i.e. computing systems that arise in nature without design or construction by intelligent agents). Philosophers and cognitive scientists have been too hasty to extrapolate lessons drawn from artificial computation to the much different case of natural computation. I discuss artificial computation in §§2-3 and natural computation in §4. I compare the two cases in §5.

§2. Representational description of artificial computation

Hardware description figures indispensably within computing practice. Ultimately, we must describe the materials from which a machine is to be built, the way those materials are to be combined, the intended physical evolution of the machine, and so on. Only then can we build the machine. A good hardware description serves as a blueprint, specifying how to construct a physical system with desired properties. Suitable hardware description is also needed for various modifications or repairs we might make. These points are evident, so I will not discuss them further.

I focus on the representational and syntactic levels. I will argue that representational description illuminates a wide range of artificial computations (§§2.1-2.2). I will then argue that syntactic description plays a key role in *mediating* between representational description and physical construction of artificial systems (§3).

§2.1 Representation elucidated

Researchers across philosophy, computer science (CS), and cognitive science use the phrase "representation" in various ways. Following common philosophical usage (e.g. Burge, 2010, p. 9), I tie representation to *veridicality-conditions*. To illustrate:

- Beliefs are evaluable as true or false. My belief *that Barack Obama is president* is true if Barack Obama is president, false if he is not.
- Declarative sentences (e.g. "Barack Obama is president") as uttered in specific conversational contexts are likewise evaluable as true or false.
- Perceptual states are evaluable as accurate or inaccurate. A perceptual state that represents presence of a red sphere is accurate only if a red sphere is before me.
- Intentions are evaluable as fulfilled or thwarted. My intention *to eat chocolate* is fulfilled if I eat chocolate, thwarted if I do not eat chocolate.

Truth-conditions, accuracy-conditions, and fulfillment-conditions are species of veridicality-conditions. Complex representations decompose into parts whose representational properties contribute to veridicality-conditions. For example, the truth-condition of "John loves Mary" is determined by the denotation of "John," the denotation of "Mary," and the satisfaction-condition of "loves." *Representational description* invokes veridicality-conditions or representational properties that contribute to veridicality-conditions.

I distinguish two ways that a system may come to have representational properties: it may have representational properties at least partly by virtue of its own activity; or it may have representational properties entirely because some other system has imposed those properties upon it. For example, the human mind has representational properties at least partly due to its own activity. In contrast, words in a book represent entirely by virtue of their connection to our linguistic conventions. The book does not contribute to representational properties of its component words.

Philosophers commonly evoke this distinction using the labels *original* versus *derived intentionality* (Haugeland, 1985) or *intrinsic* versus *observer relative meanings* (Searle, 1980). To my ear, these labels suggest that the main contrast concerns whether a system is *solely responsible* for generating its own representational properties. Yet Burge (1982) and Putnam (1975) have argued convincingly that the external physical and social environment plays a large role in determining representational properties of mental states, so that not even the mind is *solely* responsible for generating its own representational properties. I prefer the labels *indigenous* versus *inherited*, which seem to me to carry fewer misleading connotations. Representational properties of human mental states are indigenous, because human mental activity plays at least *some* role in generating representational properties of mental states.

Representational properties of words in a book are inherited, because the book plays *no* role in generating those properties.

Are representational properties of artificial computing systems inherited or indigenous? For the artificial computing systems employed in our own society, the answer is usually "inherited." For example, a simple pocket calculator only represents numbers by virtue of our linguistic conventions regarding numerals. A similar diagnosis applies to many far more sophisticated systems. Some philosophers maintain that artificial computing machines *in principle* cannot have indigenous representational properties (Searle, 1980). I think that this position is implausible and that existing arguments for it are flawed. I see no reason why a sufficiently sophisticated robot could not confer representational properties upon its own internal states. We could equip the robot with sensors or motor organs, so that it causally interacts with the external world in a suitably sophisticated way. So equipped, I see no reason why the robot could not achieve indigenous representation of its external environment. Whether any *actual existing* artificial computing systems have indigenous representational properties is a trickier question that I set aside.

§2.2 The value of representational description

To what extent do we illuminate an artificial computing system by citing its representational properties (whether those properties are inherited or indigenous)?

We often want to compute over a *non-linguistic domain*. Consider the Euclidean algorithm for computing greatest common divisors (gcds). The algorithm features arithmetical operations over natural numbers (Rescorla, 2013a). Numbers cannot be directly instantiated inside a computing system. Rather, the computing system must instantiate *numerals* that denote natural numbers. More generally, a system can compute over non-linguistic items only if the system represents those items (Rescorla, 2015c). When we describe a machine as computing over a non-linguistic domain, we presuppose (at least implicitly) that the machine's states represent elements of the domain.

Non-linguistic domains studied within computability theory typically contain *mathematical entities* (e.g. natural numbers). Real-world applications just as often involve computation over a non-linguistic, non-mathematical domain. We might want a smartphone that computes the fastest route from one location to another; or a library catalog system that allows users to recall library books; or a robot that estimates current position based on sonar and odometer readings; or a program that retrieves an individual's city of birth from a database; and so on. Each computation is defined at least partly over a non-linguistic, non-mathematical domain. To understand the computation, we must describe it as representing locations, books, people, cities, and so on. Representational description helps us articulate why we built the machine and what function it serves in our broader practices.

Some philosophers assert or intimate that representational description is relatively unimportant to computing practice (Chalmers, 2011), (Piccinini, 2008; 2009). Why not instead employ syntactic descriptions? Rather than say that a machine computes the gcd of two *numbers*, why not say that it executes a syntactic operation over *numerals*? Rather than say that a machine retrieves an individual's city of birth, why not say that the machine executes appropriate operations over *names*? Doesn't the representational level become superfluous once we isolate an underlying syntactic description?

I find such proposals jarringly divergent from actual practice within computability theory, CS, robotics, industry, and everyday life. In all these spheres, we are often primarily concerned with representational aspects of computation. The representational level is not superfluous, because it captures the primary purpose served by underlying syntactic operations. Representational description is needed to articulate the main reason why we built the machine in the first place: computation over a non-linguistic domain.

Accordingly, representational relations between computing systems and represented domains figure crucially within scientific inquiry. Three examples:

- *Computability theory* studies computational properties of notations for various non-linguistic domains (Rescorla, 2015c). Different notations embody different ways of representing the domain. For instance, computable analysis studies which operations over real numbers are computable relative to decimal notation and which are computable relative to alternative notations (Weihrauch, 2000).
- *Computer science* offers rigorous computational models that describe computations in representational terms (Rescorla, 2013a; 2014c). For example, one can codify the Euclidean algorithm as a LISP program or as a register machine program (Abelson, Sussman, and Sussman, 1996, p. 49, p. 497). The resulting programs describe computation over natural numbers, just as the pre-theoretic Euclidean algorithm does. So the programs individuate computational states partly through their representational properties.
- *Probabilistic robotics* delineates Bayesian algorithms describing how a robot navigates through its environment (Thrun, Burgard, and Fox, 2005). These algorithms presuppose that the robot maintains an internal *map* that represents spatial aspects of the environment (Rescorla, 2009). The algorithms dictate how to update probabilities over maps in light of sensor measurements and odometer readings.

We could not retain computability theory, computer science, or probabilistic robotics in anything resembling their current forms if we were to jettison representational description in favor of syntactic description.

Representational description usefully characterizes not just *inputs and outputs* to computation but also *internal states*. Suppose we describe a machine as executing the Euclidean algorithm. In doing so, we describe the machine as repeatedly executing the division operation and hence as representing a series of numbers intermediate between input and output (Rescorla, 2013a). Or suppose we describe a robot as updating probabilities over maps. The robot's updates involve internal representational states --- probability assignments to maps that represent the environment. In these examples, and many others, we want our machine to transit appropriately between internal computational states with suitable representational properties.

I do not say that representational description illuminates *all* artificial computing systems. I say only that it illuminates *many* artificial computing systems that figure prominently in science and everyday life.

§3. Syntactic description of artificial computation

There has been considerable philosophical discussion surrounding the notion of "syntax." I will assume that syntactic description has at least two properties:

- Syntactic description is *non-representational*, i.e. it does not individuate computational states through their representational properties. If we say that a computer stores the numeral "13" in some memory register, then we have offered a non-representational description, because the numeral "13" might have had different representational properties (or no representational properties at all) depending on the surrounding computational, physical, or social environment.

- Syntactic description is *multiply realizable* in Putnam’s (1975) sense: physical systems with wildly heterogeneous physical properties may satisfy a given syntactic description. Because syntactic description is multiply realizable, it is much more abstract than hardware description.

Syntactic descriptions with these two properties figure prominently in computing practice. Consider a Turing machine that manipulates stroke marks on a machine tape; or a system of logic gates that manipulate “0”s and “1”s; or a desktop computer that compiles a high-level programming language into machine code. In each case, we can describe how the machine manipulates syntactic items while ignoring representational and physical details.

Philosophers commonly gloss syntax in *functionalist* terms (Chalmers, 2011), (Field, 2001, pp. 56-57), (Fodor, 1994, pp. 108-110), (Stich, 1983, pp. 149-151),: syntactic states are individuated through their characteristic relations to one another and to computational inputs and outputs. Chalmers develops the functionalist conception using the notion of *causal topology*: “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” (2011, p. 337). On Chalmers’s view, syntactic description specifies a causal topology. It thereby constrains abstract causal structure but not physical details. A variant functionalist view allows syntactic description to constrain both abstract causal structure *and* physical aspects of inputs and outputs (e.g. geometric shapes of a desktop computer’s inputs and outputs; physical properties of a robot’s sensor inputs and motor outputs).

In everyday computing practice, we are often primarily concerned with computation over syntactic entities. We describe a computation by decomposing it into elementary syntactic operations (e.g. moving a word from one memory register to another) that transform syntactic inputs into syntactic outputs. For example, a compiler carries syntactic inputs (source code in a high-level programming language) into syntactic outputs (machine language code) through a series of syntactic manipulations. Typically, an artificial computing system falls under multiple levels of syntactic description. Syntactic description of a desktop computer is organized in a hierarchy, ranging from logic gates to machine code to assembly code to a high-level programming language. As we ascend the hierarchy, we describe progressively more abstract aspects of syntactic processing.

Even when computer scientists are most fundamentally interested in representational aspects of computation, syntactic description plays a pivotal role. Suppose we want to build a machine that executes some computation *as described in representational terms*. Representational description, taken on its own, does not specify how to build such a machine. Even if we know how we *want* our machine to transit between representational states, we may have little idea how to build a machine that so transits. As Chalmers (2012, p. 245) puts it, "[o]ne cannot go straight from representational explanation to building a mechanism; one has some hard working to do in figuring out the right mechanism." How do we ensure that our machine transits as desired between representational states? How do we build a machine that reliably transits from a computational state *that bears some relation to the represented domain* to a computational state *that bears some other desired relation to the represented domain*? For example, suppose we want a machine that executes the Euclidean algorithm. How do we ensure that our machine divides numbers into one another as the algorithm requires? Representational description by itself does not supply anything like a workable blueprint for a physical machine.

Turing's (1936) brilliant solution: supplement representational description with syntactic description. To build a machine that transits appropriately between *representational* states, we build a machine that manipulates *syntactic* items. Suitable manipulation of syntactic items endowed with suitable representational properties ensures that the machine satisfies our desired representational description. To a build a machine that executes the Euclidean algorithm, we isolate an algorithm for manipulating *numerals*. Assuming that the numerals denote suitable numbers, the machine thereby computes gcds.

Syntactic description carries us much closer than representational description to a workable blueprint for a physical machine. In principle, we know how to build a machine that executes iterated elementary syntactic operations over syntactic items. This is especially true for low-level syntactic descriptions, such as logic gate descriptions or machine language descriptions. It is also true for more abstract syntactic descriptions, such as a LISP program that specifies manipulation of list structures. Syntactic description helps us design and construct machines in a way that representational description does not.

Why exactly does syntactic description carry us closer than representational description to a workable blueprint? Because we do not know helpful sufficient conditions for a machine to instantiate desired representational properties:

- Indigenous representational properties typically depend upon complex causal interactions between the physical system and its surrounding environment --- causal interactions that we are currently unable to specify in an informative way. Even when the represented domain is mathematical, we do not have a good theory describing what it takes for the system to bear appropriate relations to the represented domain. We have nothing like a useful blueprint for ensuring that a machine has suitable *indigenous* representational properties.

- Inherited representational properties may initially seem less problematic, since it is easy enough to stipulate that a machine's internal states have certain representational properties. However, we cannot ensure that *other users* of some machine will make the same stipulations. Once we release a machine into the wild, we have little control over which representational properties other people bestow upon it. Thus, we are not able to provide anything like a useful blueprint for ensuring that a machine has suitable *inherited* representational properties.

Syntactic description avoids these problems. By focusing solely on "intrinsic" aspects of computation, without seeking to ensure that computational states bear appropriate relations to represented entities, syntactic description carries us much closer to a workable blueprint for a physical system.

Hardware description likewise supplies a workable blueprint. As Chalmers (2012) emphasizes, though, it includes numerous details that are irrelevant for many purposes. When designing or modifying a computing machine, we often do not care about the exact physical substrate that implements, say, memory registers. We would like a workable blueprint that prescinds from irrelevant hardware details. Syntactic description fulfills this desideratum. As Chalmers (2012, p. 245) puts it, syntactic description "yields a sweet spot of being detailed enough that a fully specified mechanism is provided, while at the same time providing the minimal level of detail needed for such a mechanism."

Chalmers’s analysis illuminates why syntactic description figures so centrally within computing practice. Even when we are mainly concerned with representational aspects of computation, syntactic description helps us build a physical system that transits appropriately between representational states. Syntactic description helps because it is non-representational (so that it furnishes a workable blueprint) *and* multiply realizable (so that it suppresses irrelevant hardware details).

Chalmers’s analysis is closely related to *abstraction*, a common technique in computer science (Abelson, Sussman, and Sussman, 1996, pp. 83-89). Abstraction is suppression of low-level implementation detail. For example, one might model manipulation of list structures without specifying how list structures are implemented by memory registers. Abstraction has several virtues:

- Abstraction helps us manage the enormous *complexity* of typical computing systems. Designing and modifying complex systems is much easier when we ignore details that do not bear upon our current design goals.
- Abstraction increases *flexibility*, allowing us to remain non-committal about how exactly we will implement our high-level description. Flexibility is important if we are not sure which low-level implementation is best, or if we want to permit different implementation details at some future date.

The advantages of syntactic description over hardware description are a special case of the general pressure towards abstraction. Good computer design manages complexity and promotes flexibility by suppressing irrelevant hardware details whenever possible.

I conclude that syntactic description advances our pragmatic computing goals in a distinctive way that representational description and hardware description do not. Syntactic description helps us design and build physical machines that implement representationally-specified computations. It plays a crucial role in mediating between representational description and physical construction of artificial computing machines.[1]

§4. Natural computing systems

By a "natural system," I mean one that arises in nature without design or oversight by intelligent agents. Whether a system counts as "natural" is a matter of its etiology, not its material constitution. A computing system constructed by humans from DNA or other biochemical material is not "natural," because it is an artifact. A silicon-based creature that evolved through natural selection on another planet counts as "natural," even though it is not constructed from terrestrial biochemical materials.

According to *the computational theory of mind* (CTM), the mind is a computing system. *Classical CTM* holds that the mind executes computations similar to those executed by Turing machines (Fodor, 1975; 1987; 1994; 2008), (Gallistel and King, 2009), (Putnam, 1975), (Pylyshyn, 1994). *Connectionist CTM* models mental activity using *neural networks* (Horgan and Tienson, 1996), (Ramsey, 2007), (Rumelhart, McClelland, and the PDP Research Group, 1986). Both classical and connectionist CTM trace back to seminal work of McCulloch and Pitts (1943). In (Rescorla, 2015b), I surveyed classical, connectionist, and other versions of CTM. For present purposes, I do not assume any particular version of CTM. I simply assume that the mind *in some sense* computes. Under that assumption, it makes sense to talk about "natural computing

[1] When representational properties are inherited rather than indigenous, syntactic description offers further advantages over representational description. I argue in (Rescorla, 2014b) that inherited representational properties of computational states are *causally irrelevant*: one can freely vary inherited representational properties without altering the underlying syntactic manipulations, so representational properties do not make a difference to the computation. Representational description does not furnish genuinely causal explanations of a system whose representational properties are all inherited. No such analysis applies to a computational system whose representational properties are indigenous. In that case, I claim, representational properties can be causally relevant (Rescorla, 2014b).

systems." We may therefore ask how §1's levels of description apply to natural computation --- specifically, mental computation.[2]

Hardware description is vital to the study of mental computation. Ultimately, we want to know how neural tissue physically realizes mental computations. Everyone agrees that a complete cognitive science will include detailed hardware descriptions that characterize how neural processes implement mental activity. Unfortunately, satisfactory hardware descriptions are not yet available. Although we know quite a lot about the brain, we still do not know how exactly neural processing physically realizes basic mental activities such as perception, motor control, navigation, reasoning, decision-making, and so on.

What about representational and syntactic description? Will these also figure in any complete cognitive science? I discuss representation in §4.1 and syntax in §4.2.

§4.1 Representational description of mental computation

Traditionally, philosophers have emphasized the mind's representational capacity as one of its most important features. Perception, motor control, navigation, decision-making, language acquisition, problem solving, and many other core mental activities crucially involve representational mental states. For example:

- *Perception*. Perceptual states represent the environment as being a certain way. They represent shapes, sizes, colors, locations, and other properties of distal objects. They are evaluable as accurate or inaccurate, depending on whether perceived objects actually have the represented distal properties (Burge, 2010), (Peacocke, 1992).

[2] For purposes of this paper, "mental computation" indicates computation by a *natural* system with a mind. I leave open the possibility that an artificial system (such as a sophisticated robot) might also have a mind.

- *Motor control*. The motor system transforms intentions into motor commands. When all goes well, the resulting motor commands promote fulfillment of the operative intention. For example, if I form an intention to pick up a nearby ball, then my motor system issues motor commands that hopefully result in my picking up the ball.
- *Navigation*. We routinely navigate through the physical environment. In many cases, we do so by estimating the environment's spatial layout and by planning a route to some destination (Evans, 1982). Estimates of spatial layout are evaluable as accurate or inaccurate. Representations of my desired destination are evaluable as fulfilled or thwarted, depending on whether I reach the destination.

Perception, motor control, and navigation crucially involve mental states with veridicality-conditions. So do numerous other core mental processes.

Cognitive scientists offer explanatorily successful theories that describe mental activity in representational terms:

- *Perceptual psychology* studies how the perceptual system transits from proximal sensory stimulations (e.g. retinal stimulations) to perceptual states that estimate shapes, sizes, colors, locations, and other distal properties (Palmer, 1999). Perceptual modeling individuates perceptual states through their representational properties --- *as* estimates of specific distal shapes, sizes, locations, and so on (Burge, 2010), (Rescorla, 2015a).
- *Sensorimotor psychology* studies how the motor system converts intentions into motor commands that promote fulfillment of those intentions (Rosenbaum, 2001), (Shadmehr and Mussa-Ivaldi, 2012). The science presupposes that individuals form

intentions with fulfillment-conditions (Jeannerod, 2006), (Pacherie, 2006), (Rescorla, 2016a).

- Beginning with Tolman (1948), many cognitive psychologists have postulated that mammals navigate using *cognitive maps* (Gallistel, 1990), (O'Keefe and Nadel, 1978). Mammals update their cognitive maps based on sensory input and self-motion cues. Cognitive maps represent spatial aspects of the environment, including landmark locations as well as the individual's own current location (Rescorla, 2009; forthcoming b).

In these areas, and many others, cognitive science describes how representational mental states interact with one another, with sensory inputs, and with motor outputs. A psychological theory that cites representational aspects of mentality is often called *intentional psychology*.

Recently, *Bayesian cognitive science* has elevated intentional psychology to new heights of mathematical rigor, precision, and explanatory power. The basic idea is to model mental activity using tools of Bayesian decision theory:

- According to *Bayesian perceptual psychology*, the perceptual system executes an unconscious Bayesian inference from proximal sensory stimulations to perceptual states that estimate distal conditions (Feldman, 2015), (Knill and Richard, 1996), (Rescorla, 2015a).
- According to *Bayesian sensorimotor psychology*, the sensorimotor system selects motor commands through unconscious Bayesian inference and decision-making (Bays and Wolpert, 2007), (Rescorla, 2016a), (Wolpert, 2007).
- *Bayesian models of navigation* posit Bayesian updating over cognitive maps that represent the spatial environment (Cheng, Shuttleworth, Huttenlocher, and Rieser,

2007), (Madl, Franklin, Chen, Montaldi, and Trappl, 2014; 2016), (Penny, Zeidman, and Burgess, 2013), (Rescorla, 2009).

On a Bayesian approach, the individual (or her subsystems) assigns probabilities to "hypotheses" drawn from a hypothesis space. Bayesian models typically individuate hypotheses in representational terms --- *as* representations of specific distal shapes, sizes, colors, locations, and so on. Bayesian cognitive science describes how representational mental states (probability assignments to hypotheses that represent the world) interact with one another, with sensory inputs, and with motor outputs.

The past century has witnessed successive waves of anti-representationalist sentiment. Advocates of *behaviorism* (Skinner, 1938), *connectionism* (Ramsey, 2007), *eliminative materialism* (Churchland, 1981), (Quine, 1960), (Stich, 1983), *interpretivism* (Davidson, 1980), (Dennett, 1971), *embodied cognition* (van Gelder, 1992), and *dynamical systems theory* (Chemero, 1990) frequently reject intentional psychology as unscientific, unconfirmed, unexplanatory, vacuous, or otherwise problematic. Anti-representationalists recommend that scientific psychology eschew representational discourse, relying instead upon stimulus-response psychology, or neuroscience, or some other non-representational scientific framework. In many cases, anti-representationalists launch highly abstract philosophical critiques of intentional psychology (Dennett, 1971), (Quine, 1960), (Stich, 1983). I think that anti-representationalism has dramatically failed. Anti-representational theories have repeatedly shown themselves unequipped to explain even very basic mental phenomena that intentional psychology readily explains. Abstract philosophical critiques of intentional psychology tend to be much less convincing than the representationalist theorizing they purportedly undermine.

I henceforth assume that intentional psychology illuminates perception, motor control, navigation, decision-making, and many other core mental phenomena. We reap substantial explanatory benefits by describing these phenomena in representational terms.

§4.2 Syntactic description of mental computation

Should syntactic description of mental activity likewise play an important role in cognitive science?

Fodor (1975; 1987; 1994; 2008) holds that mental computation manipulates items drawn from the *language of thought* --- an internal system of mental representations. Mental representations have *formal syntactic* properties, i.e. properties individuated without regard to representational import. Mental computation is sensitive to formal syntactic properties but not representational properties. Fodor holds that a complete scientific psychology should delineate *intentional laws*, which describe how mental states *as individuated representationally* interact with one another, with sensory inputs, and with motor outputs. Intentional laws are implemented by computations describable in syntactic terms. On Fodor's picture, syntactic manipulation of mental representations ensures that mental computation transits appropriately between representational mental states. Fodor also recognizes that any compete cognitive science will assign a prominent role to neuroscientific description. In this way, he applies §1's three-level picture to mental computation. Representational mental activity is implemented by syntactic manipulations, which are physically realized by neurophysiological processes.

Chalmers (2011; 2012) espouses a similar three-level picture of mental activity, although he places less emphasis than Fodor on representational aspects of psychological explanation. Field (2001) and Stich (1983) embrace the syntactic and hardware levels while rejecting the representational level. They hold that cognitive science should describe mental computation syntactically while ignoring representational aspects of mentality. Fodor, Chalmers, Field, and Stich all agree that syntactic description of mental computation should figure crucially within any complete cognitive science.[3]

Fodor (1981; 2008) maintains that cognitive science *already* prioritizes syntactic description of mental activity. I disagree. Contrary to what Fodor suggests, formal syntactic description does not figure in current scientific theorizing about numerous mental phenomena, including perception, motor control, deductive inference, decision-making, and so on (Rescorla, 2012; 2014b; forthcoming a). Bayesian perceptual psychology describes perceptual inference in representational terms *rather than* formal syntactic terms (Rescorla, 2015a). Bayesian sensorimotor psychology describes motor control in representational terms *rather than* formal syntactic terms (Rescorla, 2016a). There may be *some* areas where cognitive science offers syntactic explanations. For example, certain computational models of low-level insect navigation look both non-neural and non-representational (Rescorla, 2013b). But formal syntactic description is entirely absent from many core areas of cognitive science.

Plausibly, one always *can* describe mental activity in syntactic terms. The question is whether one thereby gains any explanatory benefits. There are innumerable possible ways of taxonomizing mental states. Most taxonomic schemes offer no explanatory value. For instance, we can introduce a predicate true of precisely those individuals who believe that snow is white *or* who want to drink water. However, it seems unlikely that this disjunctive predicate will play any

[3] Piccinini (2015) assigns a central role to non-representational, multiply realizable descriptions of artificial and natural computation, including mental computation. He declines to call these descriptions "syntactic." Nevertheless, the worries developed below regarding syntactic description of mental computation also apply to Piccinini's approach. For further discussion, see (Rescorla, 2016b).

significant explanatory role within a finished cognitive science. Why expect that syntactic taxonomization will play any more significant a role?

To focus the discussion, consider Chalmers's functionalist conception of syntax. Given a true representational or neurophysiological theory of a mental process, we can abstract away from representational and neural details to specify a causal topology instantiated by the process. But why suspect that we thereby gain any explanatory benefits? We can abstract away from a true scientific theory of *any* phenomenon to specify a causal topology instantiated by the phenomenon. In most cases, we do not thereby improve our explanations of the target phenomenon.

Here is a non-psychological example. The *Lotka-Volterra equations* are first-order nonlinear differential equations used in ecology to model simple predator-prey systems (Nowak, 2006):

(LV)

$$\frac{dx}{dt} = x(a - by)$$

$$\frac{dy}{dt} = y(dx - c)$$

where x is prey population level, y is predator population level, t is time, ax is prey reproduction rate, bxy is the rate at which predators eat prey, cy is predator death rate, and dxy is predator reproduction rate. Lotka (1910) introduced LV in order to model oscillating chemical reactions. Researchers have subsequently used LV to model epidemics (Kermack and McKendrick, 1927), economic interaction (Goodwin, 1967), combustion (Semenov, 1935), and other unrelated phenomena. So LV applies not just to ecological systems but also to diverse non-ecological systems *provided that we reinterpret x and y as suitable non-ecological variables*. These diverse systems instantiate the same causal topology. We can specify their shared causal topology more explicitly by taking LV's Ramsey sentence, thereby suppressing all ecological details.

Ecologists explain predator/prey population levels by using LV *where x and y are interpreted as prey and predator population levels*. We do not improve ecological explanation by noting that LV describes some chemical or economic system *when x and y are reinterpreted as chemical or economic variables*, or by supplementing LV with LV's Ramsey sentence.[4] What matters for ecological explanation are the ecological interactions described by LV, not the causal topology obtained by suppressing ecological details. That some ecological system shares a causal topology with certain chemical or economic systems is an interesting coincidence, not an explanatory significant fact that illuminates population levels. The causal topology determined by LV is not itself explanatory. It is just a byproduct of underlying ecological interactions described by LV *when x and y are interpreted as prey and predator population levels*.

Cognitive science describes causal interactions among representational mental states. By suppressing representational and neural properties, we can specify a causal topology instantiated by mental computation. But this causal topology looks like a mere byproduct of causal interactions among representational mental states. In itself, it does not seem explanatory. Certainly, actual cognitive science practice does not assign an explanatorily significant role to abstract descriptions of the causal topology instantiated by perception, motor control, or numerous other mental phenomena.

Philosophers have offered various arguments why cognitive science requires syntactic description of mental activity. I will quickly address a few prominent arguments. I critique these and other arguments more thoroughly in (Rescorla, forthcoming a).

[4] See (Morrison, 2000) for further examples along similar lines.

Argument from computational formalism (Fodor, 1981, p. 241), (Gallistel and King, 2009, p. 107), (Haugeland, 1985, p. 106): Standard computational formalisms found in computability theory operate at the syntactic level. We can model the mind as a computational system only if we postulate formal syntactic items manipulated during mental computation.

Reply: The argument misdescribes standard computational formalisms. Contrary to what the argument maintains, many standard formalisms are couched at an abstract level that remains neutral regarding the existence of formal syntactic items. We can construe many standard computational models as defined over states that are individuated representationally *rather than* syntactically. Computational modeling *per se* does not require syntactic description. My previous writings have expounded this viewpoint as applied to Turing machines (Rescorla, forthcoming a), the lambda calculus (Rescorla, 2012), and register machines (Rescorla, 2013a).

Argument from causation (Egan, 2003), (Haugeland, 1985, pp. 39-44): Representational properties are *causally irrelevant* to mental activity. Thus, intentional psychology cannot furnish causal explanations. We should replace or supplement intentional psychology with suitable non-representational descriptions, thereby attaining genuinely causal explanations of mental and behavioral outcomes.

Reply: The argument assumes that representational properties are causally irrelevant to mental activity. This assumption conflicts with pre-theoretic intuition *and* with scientific psychology (Burge, 2007, pp. 344-362), which both assign representational aspects of mentality a crucial causal role in mental activity. We have no good reason to doubt that representational properties are causally relevant to mental activity. In (Rescorla, 2014a), I argue that indigenous representational properties of a computing system can be causally relevant to the system's computations. Since mental states have indigenous representational properties, it follows that representational properties can be causally relevant to mental computation.

Argument from implementation mechanisms (Chalmers, 2012), (Fodor, 1987, pp. 18-19): We would like to describe in non-representational terms how the mind reliably transits between representational mental states. In other words, we would like to isolate *non-intentional implementation mechanisms* for intentional psychology. We should delineate a syntactic theory of mental computation, thereby specifying non-intentional mechanisms that implement transitions among representational mental states.

Reply: I agree that we should isolate non-intentional implementation mechanisms for intentional psychology. However, we can take the implementation mechanisms to be *neural* rather than *syntactic* (Rescorla, forthcoming a). We can correlate representational mental states with neural states, and we can describe how transitions among neural states track transitions among representationally-specified states. As indicated above, we do not yet know how to do this. We do not yet know the precise neural mechanisms that implement intentional mental activity. In principle, though, we should be able to isolate those mechanisms. Indeed, discovering the neural mechanisms of cognition is widely considered a holy grail for cognitive science. What value would mental syntax add to an eventual neural theory of implementation mechanisms?

Argument from explanatory generality (Chalmers, 2012): Syntactic description prescinds from both representational and neural properties. Thus, it offers a degree of generality distinct from intentional psychology and neuroscience. This distinctive generality provides us with reason to employ syntactic description. In particular, a syntactic theory of implementation mechanisms offers advantages over a neural theory of implementation mechanisms by supplying a different degree of generality.

Reply: The argument relies on a crucial premise: that generality is always an explanatory virtue. One can disambiguate this premise in various ways, using different notions of "generality." I doubt that *any* disambiguation of the premise will prove compelling. As Potochnik (2010, p. 66) notes, "Generality may be of explanatory worth, but explanations can be too general or general in the wrong way." One can boast generality through disjunctive or gerrymandered descriptions that add no explanatory value to one's theorizing (Rescorla, forthcoming a), (Williamson, 2000). To illustrate, suppose we want to explain why John failed the test. We might note that

John did not study all semester.

Alternatively, we might note that

John did not study all semester *or* John was seriously ill.

There is a clear sense in which the second explanation is more general than first. Nevertheless, it does not seem superior. One might try to disbar such counterexamples by saying that generality is a virtue *when achieved in a non-disjunctive or non-gerrymandered way*.[5] But then one would need to show that syntactic description is itself non-disjunctive and non-gerrymandered, which carries us back to the question whether syntactic description is explanatorily valuable. Thus, I doubt that generic methodological appeals to explanatory generality support syntactic modeling

[5] Strevens (2008) offers a detailed theory of explanation based on the core idea that good explanation abstracts away from as many details as possible. However, his finished theory significantly compromises that core idea, precisely so as to impugn disjunctive explanations. Strevens seeks to eliminate disjunctive explanations through a *causal contiguity* condition on good explanation (2008, pp. 101-109): when we explain some phenomenon through a causal model, all the model's realizers should form a "contiguous set" in "causal similarity space." He says that we should pursue greater abstraction only to the extent that we preserve cohesion. He says that overly disjunctive explanantia violate cohesion, because they have non-cohesive realizers. Strevens's causal contiguity condition has dramatic consequences for scientific psychology. Psychological properties are multiply realizable, so psychological explanations are apparently realized by processes that form a "non-contiguous set" in "causal similarity space." Hence, as Strevens admits (pp. 155-165, p. 167), the cohesion requirement prohibits causal models from citing psychological properties. This prohibition applies just as readily to syntactic description as to representational description. So Strevens's treatment does not provide any support for syntactic explanation of mental activity. He castigates both syntactic explanation and intentional explanation as non-cohesive.

of the mind.[6]

Overall, philosophical discussion of mental computation has vastly overemphasized formal mental syntax. Certain areas of cognitive science may posit formal syntactic mental items, but there is no clear reason to believe that mental computation *in general* is fruitfully described in syntactic terms.

§4.3 A case study: mammalian cognitive maps

To illustrate the themes of this section, let us consider *mammalian cognitive maps*. These have veridicality-conditions. For example, a cognitive map that represents a landmark as present at some physical location is veridical only if the landmark is indeed present at that location. Detailed, empirically fruitful theories describe how mammalian cognitive maps interface with sensory inputs, motor commands, and self-motion cues. The theories describe computations through which mammals form, update, and deploy cognitive maps. In describing the computations, researchers cite representational properties that contribute to veridicality-conditions --- e.g. they cite the physical location that a cognitive map attributes to a landmark. Thus, representational description plays a central role within current theories of mammalian navigation (Rescorla, forthcoming b).

[6] Potochnik (2010) argues that generality is an explanatory virtue only when it advances the research program to which an explanation contributes. Theoretical context heavily shapes whether it is explanatorily beneficial to abstract away from certain details. On this conception, one cannot motivate syntactic description through blanket appeal to the virtues of explanatory generality. One would instead need to cite specific details of psychological inquiry, arguing that the generality afforded by syntactic description promotes psychology's goals. I doubt that any such argument will prove compelling.

Neurophysiological description also plays a central role. In comparison with other areas of cognitive science, we know a fair amount about the neural underpinnings of map-based navigation. For example:

- The rat hippocampus contains *place cells*, each responding selectively to a specific spatial location.
- The rat entorhinal cortex contains *grid cells*, each responding selectively to multiple spatial locations in the available environment. They are called "grid cells" because the locations where a given cell fires form a periodic grid that covers the environment.

Neuroscientists have developed mathematical models describing how place cells, grid cells, and other such cells support mammalian navigation (Evans, Bicanski, Bush, and Burgess, forthcoming), (Giacomo, Moser, and Moser, 2011). The models aim to illuminate the neurophysiological mechanisms that underlie formation, updating, and deployment of cognitive maps. To be sure, we are still a long way from completely understanding those mechanisms.

Conspicuously lacking from current scientific research into mammalian navigation: anything resembling syntactic description. The science describes navigational computations in representational terms, and it explores the neural mechanisms that implement those representationally-described computations. It does not describe the mechanisms in multiply realizable, non-representational terms. It does not abstract away from neural details of the mechanisms. On the contrary, neural details are precisely what researchers want to illuminate. Of course, one might propose that we *supplement* representational and neurophysiological description of mammalian navigation with syntactic description. For example, one might articulate a causal topology that prescinds from representational and neural details. But we have yet to identify any clear rationale for the proposed supplementation. Certainly, current scientific practice provides no such rationale. Taking current science as our guide, syntactic description of mammalian navigation looks like an explanatorily idle abstraction from genuinely explanatory representational and neural descriptions.

§5. Contrast between artificial and natural computation

I have drawn a sharp distinction between artificial and natural computing systems. Syntactic description plays a vital role in mediating between representational description and physical construction of artificial computing systems. In contrast, many mental computations are usefully described in representational terms *rather than* syntactic terms. Why the disparity? Why is syntactic description so much more important for artificial computation than natural computation?

§3 emphasized the crucial *pragmatic* role that syntax plays within computing practice. By abstracting away from representational properties, syntactic description offers a workable blueprint for a physical machine. By abstracting away from physical properties, syntactic description ignores hardware details that are irrelevant for many purposes. These are *practical* advantages that immeasurably advance a *practical* task: design and construction of physical machines.

Admittedly, we can imagine a computing practice that eschews syntactic description. However, our own reliance on syntactic description secures important advantages over any such hypothetical practice. To illustrate, suppose an agent designs and builds a machine to execute the Euclidean algorithm. Suppose the agent describes his machine in representational terms and hardware terms but *not* syntactic terms. Now consider a second machine that has very different hardware but instantiates the same causal topology. Both duplicates satisfy a common abstract causal blueprint. This commonality is notable even if the agent does not register it. The agent could have achieved his computing goals by building the second machine rather than first. In eschewing talk about syntax, the agent foregoes valuable descriptions that promote his own computing ends. He does not employ syntactic descriptions, but he *should*.

Thus, norms of good computing design ensure a key role for syntactic description of artificial computing systems. Syntactic description enables pragmatically fruitful suppression of representational and hardware properties.

No such rationale applies to the scientific study of mental computation. Psychology is not a practical enterprise. Cognitive scientists are not trying build a computing system. Instead, they seek to explain activity in pre-given computing systems. *Constructing* an artificial computing system is a very different enterprise than *understanding* a pre-given computing system. That formal syntactic description advances the *practical* task of designing and constructing artificial computers does not establish that it advances the *explanatory* task of understanding a pre-given computational system. We have seen no reason to think that suppressing representational and hardware properties of natural computing systems advances our study of those systems. We have seen no reason to think that formal syntactic description adds explanatory value to representational and neural description of mental computation.

Any artificial computing machine was designed by intelligent agents. Good design practice dictates that those agents sometimes adopt a syntactic viewpoint even when they are mainly concerned with representational aspects of computation. No such rationale applies to natural systems, which are not designed by intelligent agents. That is why syntactic description is central to our understanding of artificial computing systems but much less central to our understanding of natural computing systems.

Consider a concrete example: perception. If Bayesian perceptual psychology is even remotely on the right track, then a finished perceptual psychology will treat the perceptual system as approximately implementing Bayesian inferences over hypotheses that represent distal properties (e.g. shapes, sizes, color, etc.). A finished perceptual psychology will also identify the neural underpinnings of Bayesian perceptual inference. It will reveal how populations of neurons approximately encode probability distributions and how neural activity approximately implements Bayesian inference. Current science already offers tentative neuroscientific conjectures in that vein (Pouget, Beck, Ma, and Latham, 2013). Will a finished perceptual psychology *also* offer formal syntactic descriptions? There is no evident reason to expect so. Formal syntactic description would suppress the two aspects of perceptual activity that figure most prominently in contemporary science: representational relations to distal properties; and neural underpinnings. Ignoring perception's most scientifically important features does not seem like a promising strategy for good scientific explanation of perception.

Now suppose we want to build an artificial perceiver that *replicates* Bayesian computations executed by the human perceptual system. We connect our artificial perceiver to artificial sensory organs that suitably resemble human sensory organs (e.g. the retina). We want to ensure that our artificial perceiver transits from sensory input to perceptual states through the same representationally-specified computations as the human perceptual system. More specifically, our artificial perceiver should execute the same approximate Bayesian inferences specified by a finished Bayesian model of human perception. As we try to build a machine that executes these Bayesian computations, the design considerations emphasized in §2.2 apply.

Syntactic description plays a valuable mediating role, helping us convert our representational description into an actual physical machine. So syntactic description greatly facilitates design and construction of our artificial perceiver, whereas syntactic description does not make any evident contribution to scientific theorizing about the human perceptual system itself.

My analysis may remind some readers of Dennett's (1987) famous distinction between the *design stance* and the *intentional stance*. When we adopt the design stance towards a system, we take the perspective of a designer trying to satisfy certain constraints, optimize certain factors, and so on. When we adopt the intentional stance, we view the system as an agent whose representational mental states interact in approximate accord with rational norms. Doesn't my position amount to saying that we should adopt the design stance towards artificial computation and the intentional stance towards mental computation?

No. First, I have repeatedly stressed that representational description is crucial for understanding many artificial computing systems. Second, representational description does not necessarily implement Dennett's "intentional stance," because representationally described activity need not conform even approximately to rational norms. Third, one might describe mental computation in syntactic terms without adopting the design stance. Fourth, I have allowed that *certain* mental computations (e.g. low-level navigational computations) may be fruitfully described in syntactic rather than representational terms.

The "design stance" is not just one perspective we happen to adopt when discussing artificial computation. Any artificial computing machine was in fact designed. Actual computer designers adopted the design stance towards it. If they had not done so, the machine would not have come into existence. In contrast, a natural computing system was not designed by anyone. For that reason, norms of good design practice do not transfer from artificial computing systems to natural computing systems. One cannot legitimately deploy design considerations to motivate syntactic description of mental computation.[7]

We have seen that proponents of §1's three-level picture often motivate syntactic description by invoking *implementation mechanisms* (Chalmers, 2012), (Fodor, 1987). The basic idea is to ground representational description in syntactic description, thereby clarifying how representational activity is physically realized. I submit that we must distinguish two endeavors, both involving "implementation mechanisms." First, one might want to *design and construct* a physical machine that realizes representational description *R*. Second, one might want to explain how *a given physical system* (e.g. the human brain) realizes representational description *R*. When we engage in the first endeavor, hardware details are fairly inessential. Good design practice dictates that we suppress hardware details whenever possible, articulating an abstract syntactic blueprint compatible with diverse physical realizations. When we engage in the second endeavor, hardware details become much more central. We want to understand how a *specific system* with *specific fixed hardware* succeeds in transiting between representational states as *R* dictates. In the first context, suppressing hardware details promotes good design. In the second context, suppressing hardware details offers no comparable advantages. On the contrary, hardware details are precisely what we want to illuminate! The first endeavor mandates an abstract syntactic viewpoint in a way that the second does not.

§6. Minds and machines

[7] Even if a terrestrial biological computing system wasn't designed by an intelligent agent, wasn't it "designed" by Mother Nature? And doesn't this show the norms of good design still apply to biological computing systems, thereby motivating an important role for formal mental syntax? This is a suspect line of argument. The design stance towards biological creatures may be useful for certain heuristic or pedagogical purposes. Strictly speaking, though, biological creatures were *not* designed. They evolved through natural selection. All legitimate talk within evolutionary theory about design is eliminable. Thus, any legitimate arguments based upon evolutionary theory should be statable without any talk about design, intentions, or the like. I doubt that, once we eliminate all such talk, we will be able to motivate syntactic description by citing anything like norms of good design.

The stunning success of the computer revolution has inspired many scientists and philosophers to pursue computational models of mental activity. Unfortunately, researchers have been too quick to extrapolate from artificial computing systems to natural computing systems. Attempts to transfer §1's three-level picture from artificial computation (where it seems quite apt) to natural computation (where it seems much less apt) are particularly suspect. Syntax plays a valuable pragmatic role in the design and construction of artificial computing systems: it helps us convert desired representational descriptions into actual physical machines. Syntax plays no comparable role in mediating between representational description and physical realization of mental computation. In many cases, syntactic description of mental activity seems like an explanatorily idle abstraction from what really matters: representational mental activity and the neural processing that implements it.

Philosophers commonly cite the computer revolution as evidence for a formal syntactic conception of mental activity. Chalmers (2012), Fodor (1987), Haugeland (1985), and others emphasize the key role that syntactic manipulation plays within *artificial* computation, arguing on that basis that cognitive science should postulate *mental* syntactic manipulation. They usually add that syntactic description of mental computation enjoys some kind of causal, explanatory, or metaphysical priority over representational description. I think that these authors distort explanatory practice within actual cognitive science, which evinces no tendency to ground representational description in syntactic description. They also neglect the essentially *pragmatic* nature of the advantages that syntactic description affords. By heeding the notable differences between artificial and natural computing systems, we may yet articulate more compelling computational theories of mind.

Acknowledgments: I presented an earlier version of this material at the 2015 annual meeting of the International Association for Computing and Philosophy, held at the University of Delaware. I am grateful to all participants for their feedback, especially Gualtiero Piccinini, Thomas Powers, and William Rapaport. Thanks also to Tyler Burge and Mark Greenberg for many helpful discussions of these ideas.

Works Cited

Abelson, H. and Sussman, G., and Sussman, J. 1996. *The Structure and Interpretation of Computer Programs*. Cambridge: MIT Press.

Bays, P., and Wolpert, D. 2007. "Computational Principles of Sensorimotor Control that Minimize Uncertainty and Variability." *Journal of Physiology* 578: 387-396.

Burge, T. 1982. "Other Bodies." In *Thought and Object*, ed. A. Woodfield. Oxford: Oxford University Press.

---. 2007. *Foundations of Mind*. Oxford: Oxford University Press.

---. 2010. *Origins of Objectivity*. Oxford: Oxford University Press.

Chalmers, D. 2011. "A Computational Foundation for the Study of Cognition." *The Journal of Cognitive Science* 12: 323-357.

---. 2012. "The Varieties of Computation: A Reply." *The Journal of Cognitive Science* 13: 213-248.

Chemero, A. 2009. *Radical Embodied Cognitive Science*. Cambridge: MIT Press.

Cheng, K., Shuttleworth, S., Huttenlocher, J., and Rieser, J. 2007. "Bayesian Integration of Spatial Information." *Psychological Bulletin* 13: 625-637.

Churchland, P. M. 1981. "Eliminative Materialism and the Propositional Attitudes." *Journal of Philosophy* 78: 67-90.

Davidson, D. 1980. *Essays on Actions and Events*. Oxford: Clarendon Press.

Dennett, D. 1971. "Intentional Systems." *Journal of Philosophy* 68: 87-106.

---. 1987. *The Intentional Stance*. Cambridge: MIT Press.

Egan, F. 2003. "Naturalistic Inquiry: Where Does Mental Representation Fit In?." In *Chomsky and His Critics*, eds. L. Antony and N. Hornstein. Malden: Blackwell.

Evans, G. 1982. *The Varieties of Reference*. Oxford: Clarendon Press.

Evans, T., Bicanski, A., Bush, D., and Burgess. N. Forthcoming. "How Environment and Self Motion Combine in Neural Representations of Space." *The Journal of Physiology*.

Feldman, J. 2015. "Bayesian Models of Perceptual Organization." In *The Oxford Handbook of Perceptual Organization*, ed. J. Wagemans. Oxford: Oxford University Press.

Field, H. 2001. *Truth and the Absence of Fact*. Oxford: Clarendon Press.

Fodor, J. 1975. *The Language of Thought*. New York: Thomas Y. Crowell.

---. 1981. *Representations*. Cambridge: MIT Press.

---. 1987. *Psychosemantics*. Cambridge: MIT Press.

---. 1994. *The Elm and the Expert*. Cambridge: MIT Press.

---. 2008. *LOT2*. Oxford: Clarendon Press.

Gallistel, C. R. 1990. *The Organization of Learning*. Cambridge: MIT Press.
Gallistel, C. R. and King, A. 2009. *Memory and the Computational Brain*. Malden: Wiley-Blackwell.
Giacomo, L., Moser, M.-B., and Moser, E. 2011. "Computational Models of Grid Cells." *Neuron* 71: 589-603.
Goodwin, R. 1967. "A Growth Cycle." In *Socialism, Capitalism and Economic Growth*, ed. C. Feinstein. Cambridge: Cambridge University Press.
Haugeland, J. 1985. *Artificial Intelligence: The Very Idea*. Cambridge: MIT Press.
Horgan, T., and Tienson, J. 1996. *Connectionism and the Philosophy of Psychology*. Cambridge: MIT Press.
Jeanerrod, M. 2006. *Motor Cognition*. Oxford: Oxford University Press.
Kermack, W., and McKendrick, A. 1927. "A Contribution to the Mathematical Theory of Epidemics." *Proceedings of the Royal Society of London* 115: 700-721.
Knill, D. and Richards, W., eds. 1996. *Perception as Bayesian Inference*. Cambridge: Cambridge University Press.
Lotka, A. J. 1910. "Contribution to the Theory of Periodic Reaction." *Journal of Physical Chemistry* 14: 271-274.
McCulloch, W. and Pitts, W. 1943. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics* 7: 115–133.
Madl, T., Franklin, S., Chen, K., Montaldi, D., and Trappl, R. 2014. "Bayesian Integration of Information in Hippocampal Place Cells." *PloS One* 9: pp. e89762.
---. 2016. "Towards Real-World Capable Spatial Memory in the LIDA Architecture." *Biologically Inspired Cognitive Architectures* 16: pp. 87-104.
Morrison, M. 2000. *Unifying Scientific Theories*. Cambridge: Cambridge University Press.
Nowak, M. 2006. *Evolutionary Dynamics: Exploring the Equations of Life*. Harvard: Belknap Press.
O'Keefe, J., and Nadel, L. 1978. *The Hippocampus as a Cognitive Map*. Oxford: Clarendon University Press.
Pacherie, E. 2006. "Towards a Dynamic Theory of Intentions." In *Does Consciousness Cause Behavior? An Investigation of the Nature of Volition*, eds. S. Pockett, W. P. Banks, and S. Gallagher. Cambridge: MIT Press.
Palmer, S. 1999. *Vision Science*. Cambridge: MIT Press.
Peacocke, C. 1992. *A Study of Concepts*. Cambridge: MIT Press.
Penny, W., Zeidman, P., and Burgess, N. 2013. "Forward and Backward Inference in Spatial Cognition." *PLoS Computational Biology* 9: pp. e1003383.
Piccinini, G. 2008. "Computation Without Representation." *Philosophical Studies* 137: 205-241.
---. 2009. "Computationalism in the Philosophy of Mind." *Philosophy Compass* 4: 512-532.
---. 2015. *Physical Computation: A Mechanistic Account*. Oxford: Oxford University Press.
Potochnik, A. 2010. "Levels of Explanation Reconceived." *Philosophy of Science* 77: 59-72.
Pouget, A., Beck, J., Ma., W. J., and Latham, P. 2013. "Probabilistic Brains: Knowns and Unknowns." *Nature Neuroscience* 16: 1170–1178.
Putnam, H. 1975. *Mind, Language, and Reality: Philosophical Papers, vol. 2*. Cambridge: Cambridge University Press.
Pylyshyn, Z. 1984. *Computation and Cognition*. Cambridge: MIT Press.
Quine, W. V. 1960. *Word and Object*. Cambridge: MIT Press.
Ramsey, W. 2007. *Representation Reconsidered*. Cambridge: Cambridge University Press.

Rescorla, M. 2009. "Cognitive Maps and the Language of Thought." *The British Journal for the Philosophy of Science* 60: 377-407.

---. 2012. "How to Integrate Representation into Computational Modeling, and Why We Should." *Journal of Cognitive Science* 13: 1-38.

---. 2013a. "Against Structuralist Theories of Computational Implementation." *British Journal for the Philosophy of Science* 64: 681-707.

---. 2013b. "Millikan on Honeybee Navigation and Communication." In *Millikan and Her Critics*, eds. D. Ryder, J. Kingsbury, and K. Williford. Malden: Wiley-Blackwell.

---. 2014a. "The Causal Relevance of Content to Computation." *Philosophy and Phenomenological Research* 88: 173-208.

---. 2014b. "Computational Modeling of the Mind: What Role for Mental Representation?". *Wiley Interdisciplinary Reviews: Cognitive Science* 6: 65-73.

---. 2014c. "A Theory of Computational Implementation." *Synthese* 191: 1277-1307.

---. 2015a. "Bayesian Perceptual Psychology." In *The Oxford Handbook of the Philosophy of Perception*, ed. M. Matthen. Oxford: Oxford University Press.

---. 2015b. "The Computational Theory of Mind." *The Stanford Encyclopedia of Philosophy*, Fall 2015, ed. E. Zalta.

---. 2015c. "The Representational Foundations of Computation." *Philosophia Mathematica* 23: 338-366.

---. 2016a. "Bayesian Sensorimotor Psychology," *Mind and Language* 31: pp. 3-36.

---. 2016b. "Review of Gualtiero Piccinini's *Physical Computation*." *BJPS Review of Books*.

---. Forthcoming a. "From Ockham to Turing --- and Back Again." In *Turing 100: Philosophical Explorations of the Legacy of Alan Turing*, eds. A. Bokulich and J. Floyd. Springer.

---. Forthcoming b. "Maps in the Head?". *The Routledge Handbook of Philosophy of Animal Minds*, eds. K. Andrews and J. Beck.

Rosenbaum, D. 2002. "Motor Control." In *Stevens' Handbook of Experimental Psychology,* vol. 1, 3rd ed., eds. H. Pashler and S. Yantis. New York: Wiley.

Rumelhart, D., McClelland, J., and the PDP Research Group. 1986. *Parallel Distributed Processing*, vol. 1. Cambridge: MIT Press.

Searle, J. 1980. "Minds, Brains, and Programs." *Behavioral and Brain Sciences* 3: 417-424.

Semenov, N. 1935. *Chemical Kinematics and Chain Reactions*. Oxford: Clarendon Press.

Shadmehr, R., and Mussa-Ivaldi, S. 2012: *Biological Learning and Control*. Cambridge: MIT Press.

Skinner, B. F. 1938. *The Behavior of Organisms*. New York: Appleton-Century-Crofts.

Stich, S. 1983. *From Folk Psychology to Cognitive Science*. Cambridge: MIT Press.

Thrun, S., Burgard, W., and Fox, D. 2005. *Probabilistic Robotics*. Cambridge: MIT Press.

Strevens, M. 2008. *Depth*. Cambridge: Harvard University Press.

Tolman, E. 1948. "Cognitive Maps in Rats and Men." *Psychological Review* 55: 189-208.

Turing, A. 1936. "On Computable Numbers, with an Application to the Entscheidungsproblem." *Proceedings of the London Mathematical Society* 42: 230-265.

van Gelder, T. 1992. "What Might Cognition Be, If Not Computation?" *Journal of Philosophy* 92: 345–381.

Weihrauch, K. 2000. *Computable Analysis: An Introduction*. Berlin: Springer.

Williamson, T. 2000. *Knowledge and its Limits*. Oxford: Oxford University Press.