Stanford Encyclopedia of Philosophy

The Computational Theory of Mind

First published Fri Oct 16, 2015

Could a machine think? Could the mind itself be a thinking machine? The computer revolution transformed discussion of these questions, offering our best prospects yet for machines that emulate reasoning, decision-making, problem solving, perception, linguistic comprehension, and other characteristic mental processes. Advances in computing raise the prospect that the mind itself is a computational system—a position known as *the computational theory of mind* (CTM). *Computationalists* are researchers who endorse CTM, at least as applied to certain important mental processes. CTM played a central role within cognitive science during the 1960s and 1970s. For many years, it enjoyed orthodox status. More recently, it has come under pressure from various rival paradigms. A key task facing computationalists is to explain what one means when one says that the mind "computes". A second task is to argue that the mind "computes" in the relevant sense. A third task is to elucidate how computational description relates to other common types of description, especially *neurophysiological description* (which cites neurophysiological properties of the organism's brain or body) and *intentional description* (which cites representational properties of mental states).

---

1. Turing machines

The intuitive notions of *computation* and *algorithm* are central to mathematics. Roughly speaking, an algorithm is an explicit, step-by-step procedure for answering some question or solving some problem. An algorithm provides *routine mechanical instructions* dictating how to proceed at each step. Obeying the instructions requires no special ingenuity or creativity. For example, the familiar grade-school algorithms describe how to compute addition, multiplication, and division. Until the early twentieth century, mathematicians relied upon informal notions of computation and algorithm without attempting anything like a formal analysis. Developments in the foundations of mathematics eventually impelled logicians to pursue a more systematic treatment. Alan Turing's landmark paper "On Computable Numbers, With an Application to the Entscheidungsproblem" (Turing 1936) offered the analysis that has proved most influential.

A *Turing machine* is an abstract model of an idealized computing device with unlimited time and storage space at its disposal. The device manipulates *symbols*, much as a human computing agent manipulates pencil marks on paper during arithmetical computation. Turing says very little about the nature of symbols. He assumes that primitive symbols are drawn from a finite alphabet. He also assumes that symbols can be inscribed or erased at "memory locations". Turing's model works as follows:

- There are infinitely many memory locations, arrayed in a linear structure. Metaphorically, these memory locations are "cells" on an infinitely long "paper

tape". More literally, the memory locations might be physically realized in various media (e.g., silicon chips).
- There is a central processor, which can access one memory location at a time. Metaphorically, the central processor is a "scanner" that moves along the paper tape one "cell" at a time.
- The central processor can enter into finitely many *machine states*.
- The central processor can perform four elementary operations: write a symbol at a memory location; erase a symbol from a memory location; access the next memory location in the linear array ("move to the right on the tape"); access the previous memory location in the linear array ("move to the left on the tape").
- Which elementary operation the central processor performs depends entirely upon two facts: which symbol is currently inscribed at the present memory location; and the scanner's own current machine state.
- A *machine table* dictates which elementary operation the central processor performs, given its current machine state and the symbol it is currently accessing. The machine table also dictates how the central processor's machine state changes given those same factors. Thus, the machine table enshrines a finite set of routine mechanical instructions governing computation.

Turing translates this informal description into a rigorous mathematical model. For more details, see the entry on Turing machines.

Turing motivates his approach by reflecting on idealized human computing agents. Citing finitary limits on our perceptual and cognitive apparatus, he argues that any symbolic algorithm executed by a human can be replicated by a suitable Turing machine. He concludes that the Turing machine formalism, despite its extreme simplicity, is powerful enough to capture all humanly executable mechanical procedures over symbolic configurations. Subsequent discussants have almost universally agreed.

Turing computation is often described as *digital* rather than *analog*. What this means is not always so clear, but the basic idea is usually that computation operates over discrete configurations. By comparison, many historically important algorithms operate over continuously variable configurations. For example, Euclidean geometry assigns a large role to *ruler-and-compass constructions*, which manipulate geometric shapes. For any shape, one can find another that differs to an arbitrarily small extent. Symbolic configurations manipulated by a Turing machine do not differ to arbitrarily small extent. Turing machines operate over discrete strings of elements (digits) drawn from a finite alphabet. One recurring controversy concerns whether the digital paradigm is well-suited to model mental activity or whether an analog paradigm would instead be more fitting (MacLennan 2012; Piccinini and Bahar 2013).

Besides introducing Turing machines, Turing (1936) proved several seminal mathematical results involving them. In particular, he proved the existence of a *universal Turing*

machine (UTM). Roughly speaking, a UTM is a Turing machine that can mimic any other Turing machine. One provides the UTM with a symbolic input that codes the machine table for Turing machine *M*. The UTM replicates *M*'s behavior, executing instructions enshrined by *M*'s machine table. In that sense, the UTM is a *programmable general purpose computer*. To a first approximation, all personal computers are also general purpose: they can mimic any Turing machine, when suitably programmed. The main caveat is that physical computers have finite memory, whereas a Turing machine has unlimited memory. More accurately, then, a personal computer can mimic any Turing machine *until it exhausts its limited memory supply*.

Turing's discussion helped lay the foundations for *computer science*, which seeks to design, build, and understand computing systems. As we know, computer scientists can now build extremely sophisticated computing machines. All these machines implement something resembling Turing computation, although the details differ from Turing's simplified model.

2. Artificial intelligence

Rapid progress in computer science prompted many, including Turing, to contemplate whether we could build a computer capable of thought. *Artificial Intelligence* (AI) aims to construct "thinking machinery". More precisely, it aims to construct computing machines that execute core mental tasks such as reasoning, decision-making, problem solving, and so on. During the 1950s and 1960s, this goal came to seem increasingly realistic (Haugeland 1985).

Early AI research emphasized *logic*. Researchers sought to "mechanize" deductive reasoning. A famous example was the *Logic Theorist* computer program (Newell and Simon 1956), which proved 38 of the first 52 theorems from *Principia Mathematica* (Whitehead and Russell 1925). In one case, it discovered a simpler proof than *Principia*'s.

Early success of this kind stimulated enormous interest inside and outside the academy. Many researchers predicted that intelligent machines were only a few years away. Obviously, these predictions have not been fulfilled. Intelligent robots do not yet walk among us. Even relatively low-level mental processes such as perception vastly exceed the capacities of current computer programs. When confident predictions of thinking machines proved too optimistic, many observers lost interest or concluded that AI was a fool's errand. Nevertheless, the decades have witnessed gradual progress. One striking success was IBM's Deep Blue, which defeated chess champion Gary Kasparov in 1997. Another major success was the driverless car Stanley (Thrun, Montemerlo, Dahlkamp, et al. 2006), which completed a 132-mile course in the Mojave Desert, winning the 2005 Defense Advanced Research Projects Agency (DARPA) Grand Challenge. A less flashy success story is the vast improvement in speech recognition algorithms.

One problem that dogged early work in AI is *uncertainty*. Nearly all reasoning and decision-making operates under conditions of uncertainty. For example, you may need to decide whether to go on a picnic while being uncertain whether it will rain. *Bayesian decision theory* is the standard mathematical model of decision-making under uncertainty. Uncertainty is codified through *probability*. Precise rules dictate how to update probabilities in light of new evidence and how to select actions in light of probabilities and utilities. (See the entries Bayes's theorem and normative theories of rational choice: expected utility for details.) In the 1980s and 1990s, technological and conceptual developments enabled efficient computer programs that implement or approximate Bayesian inference in realistic scenarios. An explosion of Bayesian AI ensued (Thrun, Burgard, and Fox 2006), including the aforementioned advances in speech recognition and driverless vehicles. Tractable algorithms that handle uncertainty are a major achievement of contemporary AI, and possibly a harbinger of more impressive future progress.

Some philosophers insist that computers, no matter how sophisticated they become, will at best *mimic* rather than *replicate* thought. A computer simulation of the weather does not really rain. A computer simulation of flight does not really fly. Even if a computing system could simulate mental activity, why suspect that it would constitute the genuine article?

Turing (1950) anticipated these worries and tried to defuse them. He proposed a scenario, now called *the Turing Test*, where one evaluates whether an unseen interlocutor is a computer or a human. A computer *passes the Turing test* if one cannot determine that it is a computer. Turing proposed that we abandon the question "Could a computer think?" as hopelessly vague, replacing it with the question "Could a computer pass the Turing test?". Turing's discussion has received considerable attention, proving especially influential within AI. Ned Block (1981) offers an influential critique. He argues that certain possible machines pass the Turing test even though these machines do not come close to genuine thought or intelligence. See the entry the Turing test for discussion of Block's objection and other issues surrounding the Turing Test.

For more on AI, see the entry logic and artificial intelligence. For much more detail, see Russell and Norvig (2010).

3. The classical computational theory of mind

Warren McCulloch and Walter Pitts (1943) first suggested that something resembling the Turing machine might provide a good model for the mind. In the 1960s, Turing computation became central to the emerging interdisciplinary initiative *cognitive science*, which studies the mind by drawing upon psychology, computer science (especially AI), linguistics, philosophy, economics (especially game theory and behavioral economics),

anthropology, and neuroscience. The label *classical computational theory of mind* (which we will abbreviate as CCTM) is now fairly standard. According to CCTM, the mind is a computational system similar in important respects to a Turing machine, and core mental processes (e.g., reasoning, decision-making, and problem solving) are computations similar in important respects to computations executed by a Turing machine. These formulations are imprecise. CCTM is best seen as a family of views, rather than a single well-defined view.[1]

It is common to describe CCTM as embodying "the computer metaphor". This description is doubly misleading.

First, CCTM is better formulated by describing the mind as a "computing system" or a "computational system" rather than a "computer". As David Chalmers (2011) notes, describing a system as a "computer" strongly suggests that the system is *programmable*. As Chalmers also notes, one need not claim that the mind is programmable simply because one regards it as a Turing-style computational system. (Most Turing machines are not programmable.) Thus, the phrase "computer metaphor" strongly suggests theoretical commitments that are inessential to CCTM. The point here is not just terminological. Critics of CCTM often object that the mind is not a programmable general purpose computer (Churchland, Koch, and Sejnowski 1990). Since classical computationalists need not claim (and usually do not claim) that the mind is a programmable general purpose computer, the objection is misdirected.

Second, CCTM is not intended metaphorically. CCTM does not simply hold that the mind is *like* a computing system. CCTM holds that the mind *literally is* a computing system. Of course, the most familiar artificial computing systems are made from silicon chips or similar materials, whereas the human body is made from flesh and blood. But CCTM holds that this difference disguises a more fundamental similarity, which we can capture through a Turing-style computational model. In offering such a model, we prescind from physical details. We attain an abstract computational description that could be physically implemented in diverse ways (e.g., through silicon chips, or neurons, or pulleys and levers). CCTM holds that a suitable abstract computational model offers a literally true description of core mental processes.

It is common to summarize CCTM through the slogan "the mind is a Turing machine". This slogan is also somewhat misleading, because no one regards Turing's precise formalism as a plausible model of mental activity. The formalism seems too restrictive in several ways:

- Turing machines execute pure symbolic computation. The inputs and outputs are symbols inscribed in memory locations. In contrast, the mind receives *sensory input* (e.g., retinal stimulations) and produces *motor output* (e.g., muscle activations). A

complete theory must describe how mental computation interfaces with sensory inputs and motor outputs.

- A Turing machine has infinite discrete memory capacity. Ordinary biological systems have finite memory capacity. A plausible psychological model must replace the infinite memory store with a large but finite memory store
- Modern computers have *random access memory*: addressable memory locations that the central processor can directly access. Turing machine memory is not addressable. The central processor can access a location only by sequentially accessing intermediate locations. Computation without addressable memory is hopelessly inefficient. For that reason, C.R. Gallistel and Adam King (2009) argue that addressable memory gives a better model of the mind than non-addressable memory.
- A Turing machine has a central processor that operates *serially*, executing one instruction at a time. Other computational formalisms relax this assumption, allowing multiple processing units that operate in *parallel*. Classical computationalists can allow parallel computations (Fodor and Pylyshyn 1988; Gallistel and King 2009: 174). See Gandy (1980) and Sieg (2009) for general mathematical treatments that encompass both serial and parallel computation.
- Turing computation is *deterministic*: total computational state determines subsequent computational state. One might instead allow *stochastic* computations. In a stochastic model, current state does not dictate a unique next state. Rather, there is a certain probability that the machine will transition from one state to another.

CCTM claims that mental activity is "Turing-style computation", allowing these and other departures from Turing's own formalism.

3.1 Machine functionalism

Hilary Putnam (1967) introduced CCTM into philosophy. He contrasted his position with *logical behaviorism* and *type-identity theory*. Each position purports to reveal the nature of mental states, including propositional attitudes (e.g., beliefs), sensations (e.g., pains), and emotions (e.g., fear). According to logical behaviorism, mental states are behavioral dispositions. According to type-identity theory, mental states are brain states. Putnam advances an opposing *functionalist* view, on which mental states are functional states. According to functionalism, a system has a mind when the system has a suitable *functional organization*. Mental states are states that play appropriate roles in the system's functional organization. Each mental state is individuated by its interactions with sensory input, motor output, and other mental states.

Functionalism offers notable advantages over logical behaviorism and type-identity theory:

- Behaviorists want to associate each mental state with a characteristic pattern of behavior—a hopeless task, because individual mental states do not usually have characteristic behavioral effects. Behavior almost always results from distinct mental states operating together (e.g., a belief and a desire). Functionalism avoids this difficulty by individuating mental states through characteristic relations not only to sensory input and behavior but also to one another.
- Type-identity theorists want to associate each mental state with a characteristic physical or neurophysiological state. Putnam casts this project into doubt by arguing that mental states are *multiply realizable*: the same mental state can be realized by diverse physical systems, including not only terrestrial creatures but also hypothetical creatures (e.g., a silicon-based Martian). Functionalism is tailor-made to accommodate multiple realizability. According to functionalism, what matters for mentality is a pattern of organization, which could be physically realized in many different ways. See the entry multiple realizability for further discussion of this argument.

Putnam defends a brand of functionalism now called *machine functionalism*. He emphasizes *probabilistic automata*, which are similar to Turing machines except that transitions between computational states are stochastic. He proposes that mental activity implements a probabilistic automaton and that particular mental states are machine states of the automaton's central processor. The machine table specifies an appropriate functional organization, and it also specifies the role that individual mental states play within that functional organization. In this way, Putnam combines functionalism with CCTM.

Machine functionalism faces several problems. One problem, highlighted by Ned Block and Jerry Fodor (1972), concerns the *productivity of thought*. A normal human can entertain a potential infinity of propositions. Machine functionalism identifies mental states with machine states of a probabilistic automaton. Since there are only finitely many machine states, there are not enough machine states to pair one-one with possible mental states of a normal human. Of course, an actual human will only ever entertain finitely many propositions. However, Block and Fodor contend that this limitation reflects limits on lifespan and memory, rather than (say) some psychological law that restricts the class of humanly entertainable propositions. A probabilistic automaton is endowed with unlimited time and memory capacity yet even still has only finitely many machine states. Apparently, then, machine functionalism mislocates the finitary limits upon human cognition.

Another problem for machine functionalism, also highlighted by Block and Fodor (1972), concerns the *systematicity of thought.* An ability to entertain one proposition is correlated with an ability to think other propositions. For example, someone who can entertain the thought *that John loves Mary* can also entertain the thought *that Mary loves John*. Thus, there seem to be systematic relations between mental states. A good theory should reflect

those systematic relations. Yet machine functionalism identifies mental states with unstructured machines states, which lack the requisite systematic relations to another. For that reason, machine functionalism does not explain systematicity. In response to this objection, machine functionalists might deny that they are obligated to explain systematicity. Nevertheless, the objection suggests that machine functionalism neglects essential features of human mentality. A better theory would explain those features in a principled way.

While the productivity and systematicity objections to machine functionalism are perhaps not decisive, they provide strong impetus to pursue an improved version of CCTM. See Block (1978) for additional problems facing machine functionalism and functionalism more generally.

3.2 The representational theory of mind

Fodor (1975, 1981, 1987, 1990, 1994, 2008) advocates a version of CCTM that accommodates systematicity and productivity much more satisfactorily. He shifts attention to the *symbols* manipulated during Turing-style computation.

An old view, stretching back at least to William of Ockham's *Summa Logicae*, holds that thinking occurs in a *language of thought* (sometimes called *Mentalese*). Fodor revives this view. He postulates a system of mental representations, including both primitive representations and complex representations formed from primitive representations. For example, the primitive Mentalese words JOHN, MARY, and LOVES can combine to form the Mentalese sentence JOHN LOVES MARY. Mentalese is *compositional*: the meaning of a complex Mentalese expression is a function of the meanings of its parts and the way those parts are combined. Propositional attitudes are relations to Mentalese symbols. Fodor calls this view *the representational theory of mind* (*RTM*). Combining RTM with CCTM, he argues that mental activity involves Turing-style computation over the language of thought. Mental computation stores Mentalese symbols in memory locations, manipulating those symbols in accord with mechanical rules.

A prime virtue of RTM is how readily it accommodates productivity and systematicity:

Productivity: RTM postulates a finite set of primitive Mentalese expressions, combinable into a potential infinity of complex Mentalese expressions. A thinker with access to primitive Mentalese vocabulary and Mentalese compounding devices has the potential to entertain an infinity of Mentalese expressions. She therefore has the potential to instantiate infinitely many propositional attitudes (neglecting limits on time and memory).

Systematicity: According to RTM, there are systematic relations between which propositional attitudes a thinker can entertain. For example, suppose I can think that John loves Mary. According to RTM, my doing so involves my standing in some relation *R* to a

Mentalese sentence JOHN LOVES MARY, composed of Mentalese words JOHN, LOVES, and MARY combined in the right way. If I have this capacity, then I also have the capacity to stand in relation *R* to the distinct Mentalese sentence MARY LOVES JOHN, thereby thinking that Mary loves John. So the capacity to think that John loves Mary is systematically related to the capacity to think that Mary loves John.

By treating propositional attitudes as relations to complex mental symbols, RTM explains both productivity and systematicity.

CCTM+RTM differs from machine functionalism in several other respects. First, machine functionalism is a theory of mental states *in general*, while RTM is only a theory of propositional attitudes. Second, proponents of CCTM+RTM need not say that propositional attitudes are individuated functionally. As Fodor (2000: 105, fn. 4) notes, we must distinguish *computationalism* (mental processes are computational) from *functionalism* (mental states are functional states). Machine functionalism endorses both doctrines. CCTM+RTM endorses only the first. Unfortunately, many philosophers still mistakenly assume that computationalism entails a functionalist approach to propositional attitudes (see Piccinini 2004 for discussion).

Philosophical discussion of RTM tends to focus mainly on *high-level human thought*, especially belief and desire. However, CCTM+RTM is applicable to a much wider range of mental states and processes. Many cognitive scientists apply it to non-human animals. For example, Gallistel and King (2009) apply it to certain invertebrate phenomena (e.g., honeybee navigation). Even confining attention to humans, one can apply CCTM+RTM to *subpersonal processing*. Fodor (1983) argues that perception involves a subpersonal "module" that converts retinal input into Mentalese symbols and then performs computations over those symbols. Thus, talk about a language of *thought* is potentially misleading, since it suggests a non-existent restriction to higher-level mental activity.

Also potentially misleading is the description of Mentalese as a *language*, which suggests that all Mentalese symbols resemble expressions in a natural language. Many philosophers, including Fodor, sometimes seem to endorse that position. However, there are possible non-propositional formats for Mentalese symbols. Proponents of CCTM+RTM can adopt a pluralistic line, allowing mental computation to operate over items akin to images, maps, diagrams, or other non-propositional representations (Johnson-Laird 2004: 187; McDermott 2001: 69; Pinker 2005: 7; Sloman 1978: 144–176). The pluralistic line seems especially plausible as applied to subpersonal processes (such as perception) and non-human animals. Michael Rescorla (2009a,b) surveys research on *cognitive maps* (Tolman 1948; O'Keefe and Nadel 1978; Gallistel 1990), suggesting that some animals may navigate by computing over mental representations more similar to maps than sentences. Elisabeth Camp (2009), citing research on baboon social interaction (Cheney and Seyfarth 2007), argues that baboons may encode social dominance relations through non-sentential tree-structured representations.

CCTM+RTM is schematic. To fill in the schema, one must provide detailed computational models of specific mental processes. A complete model will:

- describe the Mentalese symbols manipulated by the process;
- isolate elementary operations that manipulate the symbols (e.g., *inscribing a symbol in a memory location*); and
- delineate mechanical rules governing application of elementary operations.

By providing a detailed computational model, we decompose a complex mental process into a series of elementary operations governed by precise, routine instructions.

CCTM+RTM remains neutral in the traditional debate between physicalism and substance dualism. A Turing-style model proceeds at a very abstract level, not saying whether mental computations are implemented by physical stuff or Cartesian soul-stuff (Block 1983: 522). In practice, all proponents of CCTM+RTM embrace a broadly physicalist outlook. They hold that mental computations are implemented not by soul-stuff but rather by the brain. On this view, Mentalese symbols are realized by neural states, and computational operations over Mentalese symbols are realized by neural processes. Ultimately, physicalist proponents of CCTM+RTM must produce empirically well-confirmed theories that explain how exactly neural activity implements Turing-style computation. As Gallistel and King (2009) emphasize, we do not currently have such theories—though see Zylberberg, Dehaene, Roelfsema, and Sigman (2011) for some speculations.

Fodor (1975) advances CCTM+RTM as a foundation for cognitive science. He discusses mental phenomena such as decision-making, perception, and linguistic processing. In each case, he maintains, our best scientific theories postulate Turing-style computation over mental representations. In fact, he argues that our *only* viable theories have this form. He concludes that CCTM+RTM is "the only game in town". Many cognitive scientists argue along similar lines. C.R. Gallistel and Adam King (2009), Philip Johnson-Laird (1988), Allen Newell and Herbert Simon (1976), and Zenon Pylyshyn (1984) all recommend Turing-style computation over mental symbols as the best foundation for scientific theorizing about the mind.

4. Neural networks

In the 1980s, connectionism emerged as a prominent rival to classical computationalism. Connectionists draw inspiration from neurophysiology rather than logic and computer science. They employ computational models, *neural networks*, that differ significantly from Turing-style models. A *neural network* is a collection of interconnected nodes. Nodes fall into three categories: *input* nodes, *output* nodes, and *hidden* nodes (which mediate between input and output nodes). Nodes have activation values, given by real

numbers. One node can bear a *weighted connection* to another node, also given by a real number. Activations of input nodes are determined exogenously: these are the inputs to computation. *Total input activation* of a hidden or output node is a weighted sum of the activations of nodes feeding into it. Activation of a hidden or output node is a function of its total input activation; the particular function varies with the network. During neural network computation, waves of activation propagate from input nodes to output nodes, as determined by weighted connections between nodes.

In a *feedforward network*, weighted connections flow only in one direction. *Recurrent networks* have feedback loops, in which connections emanating from hidden units circle back to hidden units. Recurrent networks are less mathematically tractable than feedforward networks. However, they figure crucially in psychological modeling of various phenomena, such as phenomena that involve some kind of memory (Elman 1990).

Weights in a neural network are typically mutable, evolving in accord with a *learning algorithm*. The literature offers various learning algorithms, but the basic idea is usually to adjust weights so that *actual outputs* gradually move closer to the *target outputs* one would expect for the relevant inputs. The *backpropagation algorithm* is a widely used algorithm of this kind (Rumelhart, Hinton, and Williams 1986).

Connectionism traces back to McCulloch and Pitts (1943), who studied networks of interconnected *logic gates* (e.g., AND-gates and OR-gates). One can view a network of logic gates as a neural network, with activations confined to two values (0 and 1) and activation functions given by the usual truth-functions. McCulloch and Pitts advanced logic gates as idealized models of individual neurons. Their discussion exerted a profound influence on computer science (von Neumann 1945). Modern digital computers are simply networks of logic gates. Within cognitive science, however, researchers usually focus upon networks whose elements are more “neuron-like” than logic gates. In particular, modern-day connectionists typically emphasize analog neural networks whose nodes take continuous rather than discrete activation values. Some authors even use the phrase “neural network” so that it exclusively denotes such networks.

Neural networks received relatively scant attention from cognitive scientists during the 1960s and 1970s, when Turing-style models dominated. The 1980s witnessed a huge resurgence of interest in neural networks, especially analog neural networks, with the two-volume *Parallel Distributed Processing* (Rumelhart, McClelland, and the PDP research group, 1986; McClelland, Rumelhart, and the PDP research group, 1987) serving as a manifesto. Researchers constructed connectionist models of diverse phenomena: object recognition, speech perception, sentence comprehension, cognitive development, and so on. Impressed by connectionism, many researchers concluded that CCTM+RTM was no longer “the only game in town”.

For a detailed overview of neural networks, see Haykin (2008). For a user-friendly introduction, with an emphasis on psychological applications, see Marcus (2003).

4.1 Relation between neural networks and classical computation

Neural networks have a very different "feel" than classical (i.e., Turing-style) models. Yet classical computation and neural network computation are not mutually exclusive:

- *One can implement a neural network in a classical model*. Indeed, every neural network ever physically constructed has been implemented on a digital computer.
- *One can implement a classical model in a neural network*. Modern digital computers implement Turing-style computation in networks of logic gates. Alternatively, one can implement Turing-style computation using an analog recurrent neural network whose nodes take continuous activation values (Siegelmann and Sontag 1995).

Although some researchers suggest a fundamental opposition between classical computation and neural network computation, it seems more accurate to identify two modeling traditions that overlap in certain cases but not others (cf. Boden 1991; Piccinini 2008b). In this connection, it is also worth noting that classical computationalism and connectionist computationalism have their common origin in the work of McCulloch and Pitts.

Philosophers often say that classical computation involves "rule-governed symbol manipulation" while neural network computation is non-symbolic. The intuitive picture is that "information" in neural networks is globally distributed across the weights and activations, rather than concentrated in localized symbols. However, the notion of "symbol" itself requires explication, so it is often unclear what theorists mean by describing computation as symbolic versus non-symbolic. As mentioned in §1, the Turing formalism places very few conditions on "symbols". Regarding primitive symbols, Turing assumes just that there are finitely many of them and that they can be inscribed in read/write memory locations. Neural networks can also manipulate symbols satisfying these two conditions: as just noted, one can implement a Turing-style model in a neural network.

Many discussions of the symbolic/non-symbolic dichotomy employ a more robust notion of "symbol". On the more robust approach, a symbol is the sort of thing that represents a subject matter. Thus, something is a symbol only if it has semantic or representational properties. If we employ this more robust notion of symbol, then the symbolic/non-symbolic distinction cross-cuts the distinction between Turing-style computation and neural network computation. A Turing machine need not employ symbols in the more robust sense. As far as the Turing formalism goes, symbols manipulated during Turing computation need not have representational properties (Chalmers 2011). Conversely, a

neural network can manipulate symbols with representational properties. Indeed, an analog neural network can manipulate symbols that have a combinatorial syntax and semantics (Horgan and Tienson 1996; Marcus 2003).

Following Steven Pinker and Alan Prince (1988), we may distinguish between *eliminative connectionism* and *implementationist connectionism*.

Eliminative connectionists advance connectionism as a rival to classical computationalism. They argue that the Turing formalism is irrelevant to psychological explanation. Often, though not always, they seek to revive the *associationist* tradition in psychology, a tradition that CCTM had forcefully challenged. Often, though not always, they attack the mentalist, nativist linguistics pioneered by Noam Chomsky (1965). Often, though not always, they manifest overt hostility to the very notion of mental representation. But the defining feature of eliminative connectionism is that it uses neural networks as *replacements* for Turing-style models. Eliminative connectionists view the mind as a computing system of a radically different kind than the Turing machine. A few authors explicitly espouse eliminative connectionism (Churchland 1989; Rumelhart and McClelland 1986; Horgan and Tienson 1996), and many others incline towards it.

Implementationist connectionism is a more ecumenical position. It allows a potentially valuable role for both Turing-style models *and* neural networks, operating harmoniously at different levels of description (Marcus 2003; Smolensky 1988). A Turing-style model is higher-level, whereas a neural network model is lower-level. The neural network illuminates how the brain implements the Turing-style model, just as a description in terms of logic gates illuminates how a personal computer executes a program in a high-level programming language.

4.2 Arguments for connectionism

Connectionism excites many researchers because of the analogy between neural networks and the brain. Nodes resemble neurons, while connections between nodes resemble synapses. Connectionist modeling therefore seems more “biologically plausible” than classical modeling. A connectionist model of a psychological phenomenon apparently captures (in an idealized way) how interconnected neurons might generate the phenomenon.

These appeals to biology are problematic, because most connectionist networks are actually not so biologically plausible (Bechtel and Abrahamsen 2002: 341–343; Bermúdez 2010: 237–239; Clark 2014: 87–89; Harnish 2002: 359–362). For example, real neurons are much more heterogeneous than the interchangeable nodes that figure in typical connectionist networks. It is far from clear how, if at all, properties of the interchangeable nodes map onto properties of real neurons. Especially problematic from a

biological perspective is the backpropagation algorithm. The algorithm requires that weights between nodes can vary between excitatory and inhibitory, yet actual synapses cannot so vary (Crick and Asanuma 1986). Moreover, the algorithm assumes target outputs supplied exogenously by modelers *who know the desired answer*. In that sense, learning is *supervised*. Very little learning in actual biological systems involves anything resembling supervised training.

Even if connectionist models are not biologically plausible, they might still be *more* biologically plausible than classical models. They certainly seem closer than Turing-style models, in both details and spirit, to neurophysiological description. Many cognitive scientists worry that CCTM reflects a misguided attempt at imposing the architecture of digital computers onto the brain. Some doubt that the brain implements anything resembling digital computation, i.e., computation over discrete configurations of digits (Piccinini and Bahar 2013). Others doubt that brains display clean Turing-style separation between central processor and read/write memory (Dayan 2009). Connectionist models fare better on both scores: they do not require computation over discrete configurations of digits, and they do not postulate a clean separation between central processor and read/write memory.

Classical computationalists typically reply that it is premature to draw firm conclusions based upon biological plausibility, given how little we understand about the relation between neural, computational, and cognitive levels of description (Gallistel and King 2009; Marcus 2003). At present, we have accumulated substantial knowledge about individual neurons and their interactions in the brain. Yet we still have a tremendous amount to learn about how neural tissue accomplishes the tasks that it surely accomplishes: perception, reasoning, decision-making, language acquisition, and so on. Given our present state of relative ignorance, it would be rash to insist that the brain does not implement anything resembling Turing computation.

Connectionists offer numerous further arguments that we should employ connectionist models instead of, or in addition to, classical models. See the entry connectionism for an overview. For purposes of this entry, we mention two additional arguments.

The first argument emphasizes *learning* (Bechtel and Abrahamsen 2002: 51). A vast range of cognitive phenomena involve learning from experience. Many connectionist models are explicitly designed to model learning, through backpropagation or some other algorithm that modifies the weights between nodes. By contrast, connectionists often complain that there are no good classical models of learning. Classical computationalists can answer this worry by citing perceived defects of connectionist learning algorithms (e.g., the heavy reliance of backpropagation upon supervised training). Classical computationalists can also cite the enormous success of Bayesian decision theory, which models learning as probabilistic updating. Admittedly, Bayesian updating in the general case is computationally intractable. Nevertheless, the advances mentioned in §2 show

how classical computing systems can *approximate* idealized Bayesian updating in various realistic scenarios. These advances provide hope that classical computation can model many important cases of learning.

The second argument emphasizes *speed of computation*. Neurons are much slower than silicon-based components of digital computers. For this reason, neurons could not execute serial computation quickly enough to match rapid human performance in perception, linguistic comprehension, decision-making, etc. Connectionists maintain that the only viable solution is to replace serial computation with a "massively parallel" computational architecture—precisely what neural networks provide (Feldman and Ballard 1982; Rumelhart 1989). However, this argument is only effective against classical computationalists who insist upon serial processing. As noted in §3, some Turing-style models involve parallel processing. Many classical computationalists are happy to allow "massively parallel" mental computation, and the argument gains no traction against these researchers. That being said, the argument highlights an important question that any computationalist—whether classical, connectionist, or otherwise—must address: How does a brain built from relatively slow neurons execute sophisticated computations so quickly? Neither classical nor connectionist computationalists have answered this question satisfactorily (Gallistel and King 2009: 174 and 265).

4.3 Systematicity and productivity

Fodor and Pylyshyn (1988) offer a widely discussed critique of eliminativist connectionism. They argue that systematicity and productivity fail in connectionist models, except when the connectionist model implements a classical model. Hence, connectionism does not furnish a viable alternative to CCTM. At best, it supplies a low-level description that helps bridge the gap between Turing-style computation and neuroscientific description.

This argument has elicited numerous replies and counter-replies. Some argue that neural networks can exhibit systematicity without implementing anything like classical computational architecture (Horgan and Tienson 1996; Chalmers 1990; Smolensky 1991; van Gelder 1990). Some argue that Fodor and Pylyshyn vastly exaggerate systematicity (Johnson 2004) or productivity (Rumelhart and McClelland 1986), especially for non-human animals (Dennett 1991). These issues, and many others raised by Fodor and Pylyshyn's argument, have been thoroughly investigated over the past few decades. For further discussion, see Bechtel and Abrahamsen (2002: 156–199), Bermúdez (2005: 244–278), Chalmers (1993), Clark (2014: 84–86), and the encyclopedia entries on the language of thought hypothesis and on connectionism.

Gallistel and King (2009) advance a related but distinct productivity argument. They emphasize *productivity of mental computation*, as opposed to *productivity of mental*

states. Through detailed empirical case studies, they argue that many non-human animals can extract, store, and retrieve detailed records of the surrounding environment. For example, the Western scrub jay records where it cached food, what kind of food it cached in each location, when it cached the food, and whether it has depleted a given cache (Clayton, Emery, and Dickinson 2006). The jay can access these records and exploit them in diverse computations: computing whether a food item stored in some cache is likely to have decayed; computing a route from one location to another; and so on. The number of possible computations a jay can execute is, for all practical purposes, infinite.

CCTM explains the productivity of mental computation by positing a central processor that stores and retrieves symbols in addressable read/write memory. When needed, the central processor can retrieve arbitrary, unpredicted combinations of symbols from memory. In contrast, Gallistel and King argue, connectionism has difficulty accommodating the productivity of mental computation. Although Gallistel and King do not carefully distinguish between eliminativist and implementationist connectionism, we may summarize their argument as follows:

- Eliminativist connectionism cannot explain how organisms combine stored memories (e.g., cache locations) for computational purposes (e.g., computing a route from one cache to another). There are a virtual infinity of possible combinations that might be useful, with no predicting in advance which pieces of information must be combined in future computations. The only computationally tractable solution is symbol storage in readily accessible read/write memory locations—a solution that eliminativist connectionists reject.
- Implementationist connectionists can postulate symbol storage in read/write memory, *as implemented by a neural network*. However, the mechanisms that connectionists usually propose for implementing memory are not plausible. Existing proposals are mainly variants upon a single idea: a recurrent neural network that allows reverberating activity to travel around a loop (Elman 1990). There are many reasons why the reverberatory loop model is hopeless as a theory of long-term memory. For example, noise in the nervous system ensures that signals would rapidly degrade in a few minutes. Implementationist connectionists have thus far offered no plausible model of read/write memory.[2]

Gallistel and King conclude that CCTM is much better suited than either eliminativist or implementationist connectionism to explain a vast range of cognitive phenomena.

Critics attack this new productivity argument from various angles, focusing mainly on the empirical case studies adduced by Gallistel and King. Peter Dayan (2009), John Donahoe (2010), and Christopher Mole (2014) argue that biologically plausible neural network models can accommodate at least some of the case studies. Dayan and Donahoe argue that empirically adequate neural network models can dispense with anything resembling

read/write memory. Mole argues that, in certain cases, empirically adequate neural network models can *implement* the read/write memory mechanisms posited by Gallistel and King. Debate on these fundamental issues seems poised to continue well into the future.

4.4 Computational neuroscience

Computational neuroscience describes the nervous system through computational models. Although this research program is grounded in mathematical modeling of individual neurons, the distinctive focus of computational neuroscience is *systems* of interconnected neurons. Computational neuroscience usually models these systems as neural networks. In that sense, it is a variant, off-shoot, or descendant of connectionism. However, most computational neuroscientists do not self-identify as connectionists. There are several differences between connectionism and computational neuroscience:

- Neural networks employed by computational neuroscientists are much more biologically realistic than those employed by connectionists. The computational neuroscience literature is filled with talk about firing rates, action potentials, tuning curves, etc. These notions play at best a limited role in connectionist research, such as most of the research canvassed in (Rogers and McClelland 2014).
- Computational neuroscience is driven in large measure by knowledge about the brain, and it assigns huge importance to neurophysiological data (e.g., cell recordings). Connectionists place much less emphasis upon such data. Their research is primarily driven by behavioral data (although more recent connectionist writings cite neurophysiological data with somewhat greater frequency).
- Computational neuroscientists usually regard individual nodes in neural networks as idealized descriptions of actual neurons. Connectionists usually instead regard nodes as *neuron-like processing units* (Rogers and McClelland 2014) while remaining neutral about how exactly these units map onto actual neurophysiological entities.

One might say that computational neuroscience is concerned mainly with *neural computation* (computation by systems of neurons), whereas connectionism is concerned mainly with abstract computational models *inspired* by neural computation. But the boundaries between connectionism and computational neuroscience are admittedly somewhat porous. For an overview of computational neuroscience, see Trappenberg (2010).

Serious philosophical engagement with neuroscience dates back at least to Patricia Churchland's *Neurophilosophy* (1986). As computational neuroscience matured, Churchland became one of its main philosophical champions (Churchland, Koch, and Sejnowski 1990; Churchland and Sejnowski 1992). She was joined by Paul Churchland (1995, 2007) and others (Eliasmith 2013; Eliasmith and Anderson 2003; Piccinini and

Bahar 2013; Piccinini and Shagrir 2014). All these authors hold that theorizing about mental computation should begin with the brain, not with Turing machines or other inappropriate tools drawn from logic and computer science. They also hold that neural network modeling should strive for greater biological realism than connectionist models typically attain. Chris Eliasmith (2013) develops this neurocomputational viewpoint through the *Neural Engineering Framework*, which supplements computational neuroscience with tools drawn from control theory (Brogan 1990). He aims to "reverse engineer" the brain, building large-scale, biologically plausible neural network models of cognitive phenomena.

Computational neuroscience differs in a crucial respect from CCTM and connectionism: it abandons multiply realizability. Computational neuroscientists cite specific neurophysiological properties and processes, so their models do not apply equally well to (say) a sufficiently different silicon-based creature. Thus, computational neuroscience sacrifices a key feature that originally attracted philosophers to CTM. Computational neuroscientists will respond that this sacrifice is worth the resultant insight into neurophysiological underpinnings. But many computationalists worry that, by focusing too much on neural underpinnings, we risk losing sight of the cognitive forest for the neuronal trees. Neurophysiological details are important, but don't we also need an additional abstract level of computational description that prescinds from such details? Gallistel and King (2009) argue that a myopic fixation upon what we currently know about the brain has led computational neuroscience to shortchange core cognitive phenomena such as navigation, spatial and temporal learning, and so on. Similarly, Edelman (2014) complains that the Neural Engineering Framework substitutes a blizzard of neurophysiological details for satisfying psychological explanations.

Despite the differences between connectionism and computational neuroscience, these two movements raise many similar issues. In particular, the dialectic from §4.4 regarding systematicity and productivity arises in similar form.

5. Computation and representation

Philosophers and cognitive scientists use the term "representation" in diverse ways. Within philosophy, the most dominant usage ties representation to intentionality, i.e., the "aboutness" of mental states. Contemporary philosophers usually elucidate intentionality by invoking *representational content*. A representational mental state has a content that represents the world as being a certain way, so we can ask whether the world is indeed that way. Thus, representationally contentful mental states are *semantically evaluable* with respect to properties such as truth, accuracy, fulfillment, and so on. To illustrate:

- Beliefs are the sorts of things that can be true or false. My belief *that Barack Obama is president* is true if Barack Obama is president, false if he is not.

- Perceptual states are the sorts of things that can be accurate or inaccurate. My perceptual experience *as of a red sphere* is accurate only if a red sphere is before me.
- Desires are the sorts of things that can fulfilled or thwarted. My desire *to eat chocolate* is fulfilled if I eat chocolate, thwarted if I do not eat chocolate.

Beliefs have truth-conditions (conditions under which they are true), perceptual states have accuracy-conditions (conditions under which they are accurate), and desires have fulfillment-conditions (conditions under which they are fulfilled).

In ordinary life, we frequently predict and explain behavior by invoking beliefs, desires, and other representationally contentful mental states. We identify these states through their representational properties. When we say "Frank believes that Barack Obama is president", we specify the condition under which Frank's belief is true (namely, that Barack Obama is president). When we say "Frank wants to eat chocolate", we specify the condition under which Frank's desire is fulfilled (namely, that Frank eats chocolate). So folk psychology assigns a central role to *intentional descriptions*, i.e., descriptions that identify mental states through their representational properties. Whether scientific psychology should likewise employ intentional descriptions is a contested issue within contemporary philosophy of mind.

Intentional realism is realism regarding representation. At a minimum, this position holds that representational properties are genuine aspects of mentality. Usually, it is also taken to hold that scientific psychology should freely employ intentional descriptions when appropriate. Intentional realism is a popular position, advocated by Tyler Burge (2010a), Jerry Fodor (1987), Christopher Peacocke (1992, 1994), and many others. One prominent argument for intentional realism cites *cognitive science practice*. The argument maintains that intentional description figures centrally in many core areas of cognitive science, such as perceptual psychology and linguistics. For example, perceptual psychology describes how perceptual activity transforms sensory inputs (e.g., retinal stimulations) into representations of the distal environment (e.g., perceptual representations of distal shapes, sizes, and colors). The science identifies perceptual states by citing representational properties (e.g., representational relations to specific distal shapes, sizes, colors). Assuming a broadly scientific realist perspective, the explanatory achievements of perceptual psychology support a realist posture towards intentionality.

Eliminativism is a strong form of anti-realism about intentionality. Eliminativists dismiss intentional description as vague, context-sensitive, interest-relative, explanatorily superficial, or otherwise problematic. They recommend that scientific psychology jettison representational content. An early example is W.V. Quine's *Word and Object* (1960), which seeks to replace intentional psychology with behaviorist stimulus-response psychology. Paul Churchland (1981), another prominent eliminativist, wants to replace intentional psychology with neuroscience.

Between intentional realism and eliminativism lie various intermediate positions. Daniel Dennett (1971, 1987) acknowledges that intentional discourse is predictively useful, but he questions whether mental states *really* have representational properties. According to Dennett, theorists who employ intentional descriptions are not *literally* asserting that mental states have representational properties. They are merely adopting the "intentional stance". Donald Davidson (1980) espouses a neighboring *interpretivist* position. He emphasizes the central role that intentional ascription plays within ordinary interpretive practice, i.e., our practice of interpreting one another's mental states and speech acts. At the same time, he questions whether intentional psychology will find a place within mature scientific theorizing. Davidson and Dennett both profess realism about intentional mental states. Nevertheless, both philosophers are customarily read as intentional anti-realists. (In particular, Dennett is frequently read as a kind of *instrumentalist* about intentionality.) One source of this customary reading involves *indeterminacy of interpretation*. Suppose that behavioral evidence allows two conflicting interpretations of a thinker's mental states. Following Quine, Davidson and Dennett both say there is then "no fact of the matter" regarding which interpretation is correct. This diagnosis indicates a less than fully realist attitude towards intentionality.

Debates over intentionality figure prominently in philosophical discussion of CTM. Let us survey some highlights.

5.1 Computation as formal

Classical computationalists typically assume what one might call *the formal-syntactic conception of computation* (FSC). The intuitive idea is that computation manipulates symbols in virtue of their formal syntactic properties rather than their semantic properties.

FSC stems from innovations in mathematical logic during the late 19th and early 20th centuries, especially seminal contributions by George Boole and Gottlob Frege. In his *Begriffsschrift* (1879/1967), Frege effected a thoroughgoing *formalization* of deductive reasoning. To formalize, we specify a *formal language* whose component linguistic expressions are individuated non-semantically (e.g., by their geometric shapes). We may have some intended interpretation in mind, but elements of the formal language are purely syntactic entities that we can discuss without invoking semantic properties such as reference or truth-conditions. In particular, we can specify *inference rules* in formal syntactic terms. If we choose our inference rules wisely, then they will cohere with our intended interpretation: they will carry true premises to true conclusions. Through formalization, Frege invested logic with unprecedented rigor. He thereby laid the groundwork for numerous subsequent mathematical and philosophical developments.

Formalization plays a significant foundational role within computer science. We can program a Turing-style computer that manipulates linguistic expressions drawn from a

formal language. If we program the computer wisely, then its syntactic machinations will cohere with our intended semantic interpretation. For example, we can program the computer so that it carries true premises only to true conclusions, or so that it updates probabilities as dictated by Bayesian decision theory.

FSC holds that *all* computation manipulates formal syntactic items, without regard to any semantic properties those items may have. Precise formulations of FSC vary. Computation is said to be "sensitive" to syntax but not semantics, or to have "access" only to syntactic properties, or to operate "in virtue" of syntactic rather than semantic properties, or to be impacted by semantic properties only as "mediated" by syntactic properties. It is not always so clear what these formulations mean or whether they are equivalent to one another. But the intuitive picture is that syntactic properties have causal/explanatory primacy over semantic properties in driving computation forward.

Fodor's article "Methodological Solipsism Considered as a Research Strategy in Cognitive Psychology" (1980) offers an early statement. Fodor combines FSC with CCTM+RTM. He analogizes Mentalese to formal languages studied by logicians: it contains simple and complex items individuated non-semantically, just as typical formal languages contain simple and complex expressions individuated by their shapes. Mentalese symbols have a semantic interpretation, but this interpretation does not (directly) impact mental computation. A symbol's formal properties, rather than its semantic properties, determine how computation manipulates the symbol. In that sense, the mind is a "syntactic engine". Virtually all classical computationalists follow Fodor in endorsing FSC.

Connectionists often deny that neural networks manipulate syntactically structured items. For that reason, many connectionists would hesitate to accept FSC. Nevertheless, most connectionists endorse a *generalized formality thesis*: computation is insensitive to semantic properties. The generalized formality thesis raises many of the same philosophical issues raised by FSC. We focus here on FSC, which has received the most philosophical discussion.

Fodor combines CCTM+RTM+FSC with intentional realism. He holds that CCTM+RTM+FSC vindicates folk psychology by helping us convert common sense intentional discourse into rigorous science. He motivates his position with a famous abductive argument for CCTM+RTM+FSC (1987: 18–20). Strikingly, mental activity tracks semantic properties in a coherent way. For example, deductive inference carries premises to conclusions that are true if the premises are true. How can we explain this crucial aspect of mental activity? Formalization shows that syntactic manipulations can track semantic properties, and computer science shows how to build physical machines that execute desired syntactic manipulations. If we treat the mind as a syntax-driven machine, then we can explain why mental activity tracks semantic properties in a coherent way. Moreover, our explanation does not posit causal mechanisms radically different from

those posited within the physical sciences. We thereby answer the pivotal question: *How is rationality mechanically possible*?

Stephen Stich (1983) and Hartry Field (2001) combine CCTM+FSC with eliminativism. They recommend that cognitive science model the mind in formal syntactic terms, eschewing intentionality altogether. They grant that mental states have representational properties, but they ask what explanatory value scientific psychology gains by invoking those properties. Why supplement formal syntactic description with intentional description? If the mind is a syntax-driven machine, then doesn't representational content drop out as explanatorily irrelevant?

At one point in his career, Putnam (1983: 139–154) combined CCTM+FSC with a Davidson-tinged *interpretivism*. Cognitive science should proceed along the lines suggested by Stich and Field, delineating purely formal syntactic computational models. Formal syntactic modeling co-exists with ordinary interpretive practice, in which we ascribe intentional contents to one another's mental states and speech acts. Interpretive practice is governed by holistic and heuristic constraints, which stymie attempts at converting intentional discourse into rigorous science. For Putnam, as for Field and Stich, the scientific action occurs at the formal syntactic level rather than the intentional level.

CTM+FSC comes under attack from various directions. One criticism targets *the causal relevance of representational content* (Block 1990; Figdor 2009; Kazez 1995). Intuitively speaking, the contents of mental states are causally relevant to mental activity and behavior. For example, my desire to drink water rather than orange juice causes me to walk to the sink rather than the refrigerator. The content of my desire (*that I drink water*) seems to play an important causal role in shaping my behavior. According to Fodor (1990: 137–159), CCTM+RTM+FSC accommodates such intuitions. Formal syntactic activity *implements* intentional mental activity, thereby ensuring that intentional mental states causally interact in accord with their contents. However, it is not so clear that this analysis secures the causal relevance of content. FSC says that computation is "sensitive" to syntax but not semantics. Depending on how one glosses the key term "sensitive", it can look like representational content is causally irrelevant, with formal syntax doing all the causal work. Here is an analogy to illustrate the worry. When a car drives along a road, there are stable patterns involving the car's shadow. Nevertheless, shadow position at one time does not influence shadow position at a later time. Similarly, CCTM+RTM+FSC may explain how mental activity instantiates stable patterns described in intentional terms, but this is not enough to ensure the causal relevance of content. If the mind is a syntax-driven machine, then causal efficacy seems to reside at the syntactic rather the semantic level. Semantics is just "along for the ride". Apparently, then, CTM+FSC encourages the conclusion that representational properties are causally inert. The conclusion may not trouble eliminativists, but intentional realists usually want to avoid it.

A second criticism dismisses the formal-syntactic picture as speculation ungrounded in scientific practice. Tyler Burge (2010a,b, 2013: 479–480) contends that formal syntactic description of mental activity plays no significant role within large areas of cognitive science, including the study of theoretical reasoning, practical reasoning, and perception. In each case, Burge argues, the science employs intentional description *rather than* formal syntactic description. For example, perceptual psychology individuates perceptual states not through formal syntactic properties but through representational relations to distal shapes, sizes, colors, and so on. To understand this criticism, we must distinguish *formal syntactic description* and *neurophysiological description*. Everyone agrees that a complete scientific psychology will assign prime importance to neurophysiological description. However, neurophysiological description is distinct from formal syntactic description, because formal syntactic description is supposed to be multiply realizable in the neurophysiological. The issue here is whether scientific psychology should supplement *intentional descriptions* and *neurophysiological descriptions* with *multiply realizable, non-intentional formal syntactic* descriptions.

5.2 Externalism about mental content

Putnam's landmark article "The Meaning of 'Meaning'" (1975: 215–271) introduced the *Twin Earth thought experiment*, which postulates a world just like our own except that H_2O is replaced by a qualitatively similar substance XYZ with different chemical composition. Putnam argues that XYZ is not water and that speakers on Twin Earth use the word "water" to refer to XYZ rather than to water. Burge (1982) extends this conclusion from *linguistic reference* to *mental content*. He argues that Twin Earthlings instantiate mental states with different contents. For example, if Oscar on Earth thinks *that water is thirst-quenching*, then his duplicate on Twin Earth thinks a thought with a different content, which we might gloss as *that twater is thirst-quenching*. Burge concludes that mental content does not supervene upon internal neurophysiology. Mental content is individuated partly by factors outside the thinker's skin, including causal relations to the environment. This position is *externalism about mental content*.

Formal syntactic properties of mental states are widely taken to supervene upon internal neurophysiology. For example, Oscar and Twin Oscar instantiate the same formal syntactic manipulations. Assuming content externalism, it follows that there is a huge gulf between ordinary intentional description and formal syntactic description.

Content externalism raises serious questions about the explanatory utility of representational content for scientific psychology:

Argument from Causation (Fodor 1987, 1991): How can mental content exert any causal influence except as manifested within internal neurophysiology? There is no "psychological action at a distance". Differences in the physical environment impact

behavior only by inducing differences in local brain states. So the only causally relevant factors are those that supervene upon internal neurophysiology. Externally individuated content is *causally irrelevant*.

Argument from Explanation (Stich 1983): Rigorous scientific explanation should not take into account factors outside the subject's skin. Folk psychology may taxonomize mental states through relations to the external environment, but scientific psychology should taxonomize mental states entirely through factors that supervene upon internal neurophysiology. It should treat Oscar and Twin Oscar as psychological duplicates.[3]

Some authors pursue the two arguments in conjunction with one another. Both arguments reach the same conclusion: externally individuated mental content finds no legitimate place within causal explanations provided by scientific psychology. Stich (1983) argues along these lines to motivate his formal-syntactic eliminativism.

Many philosophers respond to such worries by promoting *content internalism*. Whereas content externalists favor *wide content* (content that does not supervene upon internal neurophysiology), content internalists favor *narrow content* (content that does so supervene). Narrow content is what remains of mental content when one factors out all external elements. At one point in his career, Fodor (1981, 1987) pursued internalism as a strategy for integrating intentional psychology with CCTM+RTM+FSC. While conceding that wide content should not figure in scientific psychology, he maintained that narrow content should play a central explanatory role.

Radical internalists insist that *all* content is narrow. A typical analysis holds that Oscar is thinking not about water but about some more general category of substance that subsumes XYZ, so that Oscar and Twin Oscar entertain mental states with the same contents. Tim Crane (1991) and Gabriel Segal (2000) endorse such an analysis. They hold that folk psychology always individuates propositional attitudes narrowly. A less radical internalism recommends that we recognize narrow content *in addition to* wide content. Folk psychology may sometimes individuate propositional attitudes widely, but we can also delineate a viable notion of narrow content that advances important philosophical or scientific goals. Internalists have proposed various candidate notions of narrow content (Block 1986; Chalmers 2002; Cummins 1989; Fodor 1987; Lewis 1994; Loar 1988; Mendola 2008). See the entry narrow mental content for an overview of prominent candidates.

Externalists complain that existing theories of narrow content are sketchy, implausible, useless for psychological explanation, or otherwise objectionable (Burge 2007; Sawyer 2000; Stalnaker 1999). Externalists also question internalist arguments that scientific psychology requires narrow content:

Argument from Causation: Externalists insist that wide content can be causally relevant. The details vary among externalists, and discussion often becomes intertwined with complex issues surrounding causation, counterfactuals, and the metaphysics of mind. See the entry mental causation for an introductory overview, and see Burge (2007), Rescorla (2014a), and Yablo (1997, 2003) for representative externalist discussion.

Argument from Explanation: Externalists claim that psychological explanation can legitimately taxonomize mental states through factors that outstrip internal neurophysiology (Peacocke 1993). Burge observes that non-psychological sciences often individuate explanatory kinds *relationally*, i.e., through relations to external factors. For example, whether an entity counts as a heart depends (roughly) upon whether its biological function in its normal environment is to pump blood. So physiology individuates organ kinds relationally. Why can't psychology likewise individuate mental states relationally? For a notable exchange on these issues, see Burge (1986, 1989, 1995) and Fodor (1987, 1991).

Externalists doubt that we have any good reason to replace or supplement wide content with narrow content. They dismiss the search for narrow content as a wild goose chase.

Burge (2007, 2010a) defends externalism by analyzing current cognitive science. He argues that many branches of scientific psychology (especially perceptual psychology) individuate mental content through causal relations to the external environment. He concludes that scientific practice embodies an externalist perspective. By contrast, he maintains, narrow content is a philosophical fantasy ungrounded in current science.

Suppose we abandon the search for narrow content. What are the prospects for combining CTM+FSC with externalist intentional psychology? The most promising option emphasizes *levels of explanation*. We can say that intentional psychology occupies one level of explanation, while formal-syntactic computational psychology occupies a different level. Fodor advocates this approach in his later work (1994, 2008). He comes to reject narrow content as otiose. He suggests that formal syntactic mechanisms implement externalist psychological laws. Mental computation manipulates Mentalese expressions in accord with their formal syntactic properties, and these formal syntactic manipulations ensure that mental activity instantiates appropriate law-like patterns defined over wide contents.

In light of the internalism/externalism distinction, let us revisit the eliminativist challenge raised in §5.1: what explanatory value does intentional description add to formal-syntactic description? Internalists can respond that suitable formal syntactic manipulations determine and maybe even constitute narrow contents, so that internalist intentional description is already implicit in suitable formal syntactic description (cf. Field 2001: 75). Perhaps this response vindicates intentional realism, perhaps not. Crucially, though, no such response is available to content externalists. Externalist intentional description is not

implicit in formal syntactic description, because one can hold formal syntax fixed while varying wide content. Thus, content externalists who espouse CTM+FSC must say what we gain by supplementing formal-syntactic explanations with intentional explanations. Once we accept that mental computation is sensitive to syntax but not semantics, it is far from clear that any useful explanatory work remains for wide content. Fodor addresses this challenge at various points, offering his most systematic treatment in *The Elm and the Expert* (1994). See Arjo (1996), Aydede (1998), Aydede and Robbins (2001), Wakefield (2002); Perry (1998), and Wakefield (2002) for criticism. See Rupert (2008) and Schneider (2005) for positions close to Fodor's. See also Dretske (1993), which pursues an alternative strategy for vindicating the explanatory relevance of wide content.

5.3 Content-involving computation

The perceived gulf between computational description and intentional description animates many writings on CTM. A few philosophers try to bridge the gulf using computational descriptions that individuate computational states in representational terms. These descriptions are *content-involving*, to use Christopher Peacocke's (1994) terminology. On the content-involving approach, there is no rigid demarcation between computational and intentional description. In particular, certain scientifically valuable descriptions of mental activity are both computational and intentional. Call this position *content-involving computationalism*.

Content-involving computationalists need not say that all computational description is intentional. To illustrate, suppose we describe a simple Turing machine that manipulates symbols individuated by their geometric shapes. Then the resulting computational description is not plausibly content-involving. Accordingly, content-involving computationalists do not usually advance content-involving computation as a general theory of computation. They claim only that *some* important computational descriptions are content-involving.

One can develop content-involving computationalism in an internalist or externalist direction. *Internalist content-involving computationalists* hold that some computational descriptions identify mental states partly through their *narrow* contents. Murat Aydede (2005) recommends a position along these lines. *Externalist content-involving computationalism* holds that certain computational descriptions identify mental states partly through their *wide* contents. Tyler Burge (2010a: 95–101), Christopher Peacocke (1994, 1999), Michael Rescorla (2012), and Mark Sprevak (2010) espouse this position. Oron Shagrir (2001) advocates a content-involving computationalism that is neutral between internalism and externalism.

Externalist content-involving computationalists typically cite cognitive science practice as a motivating factor. For example, perceptual psychology describes the perceptual system

as computing an estimate of some object's size from retinal stimulations and from an estimate of the object's depth. Perceptual "estimates" are identified representationally, as representations of specific distal sizes and depths. Quite plausibly, representational relations to specific distal sizes and depths do not supervene on internal neurophysiology. Quite plausibly, then, perceptual psychology type-identifies perceptual computations through wide contents. So externalist content-involving computationalism seems to harmonize well with current cognitive science.

A major challenge facing content-involving computationalism concerns the interface with standard computationalism formalisms, such as the Turing machine. How exactly do content-involving descriptions relate to the computational models found in logic and computer science? Philosophers usually assume that these models offer non-intentional descriptions. If so, that would be a major and perhaps decisive blow to content-involving computationalism.

Arguably, though, many familiar computational formalisms allow a content-involving rather than formal syntactic construal. To illustrate, consider the Turing machine. One *can* individuate the "symbols" comprising the Turing machine alphabet non-semantically, through factors akin to geometric shape. But does Turing's formalism *require* a non-semantic individuative scheme? Arguably, the formalism allows us to individuate symbols partly through their contents. Of course, the machine table for a Turing machine does not explicitly cite semantic properties of symbols (e.g., denotations or truth-conditions). Nevertheless, the machine table can encode mechanical rules that describe how to manipulate symbols, where those symbols are type-identified in content-involving terms. In this way, the machine table dictates transitions among content-involving states without explicitly mentioning semantic properties. Aydede (2005) suggests an internalist version of this view, with symbols type-identified through their narrow contents.[4] Rescorla (forthcoming) develops the view in an externalist direction, with symbols type-identified through their wide contents. He argues that some Turing-style models describe computational operations over externalistically individuated Mentalese symbols.[5]

In principle, one might embrace both externalist content-involving computational description *and* formal syntactic description. One might say that these two kinds of description occupy distinct levels of explanation. Peacocke suggests such a view. Other content-involving computationalists regard formal syntactic descriptions of the mind more skeptically. For example, Burge questions what explanatory value formal syntactic description contributes to certain areas of scientific psychology (such as perceptual psychology). From this viewpoint, the eliminativist challenge posed in §5.1 has matters backwards. We should not assume that formal syntactic descriptions are explanatorily valuable and then ask what value intentional descriptions contribute. We should instead embrace the externalist intentional descriptions offered by current cognitive science and then ask what value formal syntactic description contributes.

Proponents of formal syntactic description respond by citing *implementation mechanisms*. Externalist description of mental activity presupposes that suitable causal-historical relations between the mind and the external physical environment are in place. But surely we want a "local" description that ignores external causal-historical relations, a description that reveals underlying causal mechanisms. Fodor (1987, 1994) argues in this way to motivate the formal syntactic picture. For possible externalist responses to the argument from implementation mechanisms, see Burge (2010b), Shea (2013), and Sprevak (2010). Debate over this argument, and more generally over the relation between computation and representation, seems likely to continue into the indefinite future.

6. Alternative conceptions of computation

The literature offers several alternative conceptions, usually advanced as foundations for CTM. In many cases, these conceptions overlap with one another or with the conceptions considered above.

6.1 Information-processing

It is common for cognitive scientists to describe computation as "information-processing". It is less common for proponents to clarify what they mean by "information" or "processing". Lacking clarification, the description is little more than an empty slogan.

Claude Shannon introduced a scientifically important notion of "information" in his 1948 article "A Mathematical Theory of Communication". The intuitive idea is that information measures *reduction in uncertainty*, where reduced uncertainty manifests as an altered probability distribution over possible states. Shannon codified this idea within a rigorous mathematical framework, laying the foundation for *information theory* (Cover and Thomas 2006). Shannon information is fundamental to modern engineering. It finds fruitful application within cognitive science, especially cognitive neuroscience. Does it support a convincing analysis of computation as "information-processing"? Consider an old-fashioned tape machine that records messages received over a wireless radio. Using Shannon's framework, one can measure how much information is carried by some recorded message. There is a sense in which the tape machine "processes" Shannon information whenever we replay a recorded message. Still, the machine does not seem to implement a non-trivial computational model.[6] Certainly, neither the Turing machine formalism nor the neural network formalism offers much insight into the machine's operations. Arguably, then, a system can process Shannon information without executing computations in any interesting sense.

Confronted with such examples, one might try to isolate a more demanding notion of "processing", so that the tape machine does not "process" Shannon information.

Alternatively, one might insist that the tape machine executes non-trivial computations. Piccinini and Scarantino (2010) advance a highly general notion of computation—which they dub *generic computation*—with that consequence.

A second prominent notion of information derives from Paul Grice's (1989) influential discussion of *natural meaning*. Natural meaning involves reliable, counterfactual-supporting correlations. For example, tree rings correlate with the age of the tree, and pox correlate with chickenpox. We colloquially describe tree rings as carrying information about tree age, pox as carrying information about chickenpox, and so on. Such descriptions suggest a conception that ties information to reliable, counterfactual-supporting correlations. Fred Dretske (1981) develops this conception into a systematic theory, as do various subsequent philosophers. Does Dretske-style information subserve a plausible analysis of computation as "information-processing"? Consider an old-fashioned *bimetallic strip thermostat*. Two metals are joined together into a strip. Differential expansion of the metals causes the strip to bend, thereby activating or deactivating a heating unit. Strip state reliably correlates with current ambient temperature, and the thermostat "processes" this information-bearing state when activating or deactivating the heater. Yet the thermostat does not seem to implement any non-trivial computational model. One would not ordinarily regard the thermostat as computing. Arguably, then, a system can process Dretske-style information without executing computations in any interesting sense. Of course, one might try to handle such examples through maneuvers parallel to those from the previous paragraph.

A third prominent notion of information is *semantic information*, i.e., representational content.[7] Some philosophers hold that a physical system computes only if the system's states have representational properties (Dietrich 1989; Fodor 1998: 10; Ladyman 2009; Shagrir 2006; Sprevak 2010). In that sense, information-processing is *necessary* for computation. As Fodor memorably puts it, "no computation without representation" (1975: 34). However, this position is debatable. Chalmers (2011) and Piccinini (2008a) contend that a Turing machine might execute computations even though symbols manipulated by the machine have no semantic interpretation. The machine's computations are purely syntactic in nature, lacking anything like semantic properties. On this view, representational content is not necessary for a physical system to count as computational.

It remains unclear whether the slogan "computation is information-processing" provides much insight. Nevertheless, the slogan seems unlikely to disappear from the literature anytime soon. For further discussion of possible connections between computation and information, see Gallistel and King (2009: 1–26), Lizier, Flecker, and Williams (2013), Milkowski (2013), and Piccinini and Scarantino (2010).

6.2 Function evaluation

In a widely cited passage, the perceptual psychologist David Marr (1982) distinguishes three levels at which one can describe an "information-processing device":

> *Computational theory*: "[t]he device is characterized as a mapping from one kind of information to another, the abstract properties of this mapping are defined precisely, and its appropriateness and adequacy for the task as hand are demonstrated" (p. 24).
>
> *Representation and algorithm*: "the choice of representation for the input and output and the algorithm to be used to transform one into the other" (pp. 24–25).
>
> *Hardware implementation*: "the details of how the algorithm and representation are realized physically" (p. 25).

Marr's three levels have attracted intense philosophical scrutiny. For our purposes, the key point is that Marr's "computational level" describes a mapping from inputs to outputs, without describing intermediate steps. Marr illustrates his approach by providing "computational level" theories of various perceptual processes, such as edge detection.

Marr's discussion suggests a *functional conception of computation*, on which computation is a matter of transforming inputs into appropriate outputs. Frances Egan elaborates the functional conception over a series of articles (1991, 1992, 1999, 2003, 2010, 2014). Like Marr, she treats computational description as description of input-output relations. She also claims that computational models characterize a purely *mathematical* function: that is, a mapping from mathematical inputs to mathematical outputs. She illustrates by considering a visual mechanism (called "Visua") that computes an object's depth from retinal disparity. She imagines a neurophysiological duplicate ("Twin Visua") embedded so differently in the physical environment that it does not represent depth. Visua and Twin Visua instantiate perceptual states with different representational properties. Nevertheless, Egan says, vision science treats Visua and Twin Visua as *computational duplicates*. Visua and Twin Visua compute the same mathematical function, even though the computations have different representational import in the two cases. Egan concludes that computational modeling of the mind yields an "abstract mathematical description" consistent with many alternative possible representational descriptions. Intentional attribution is just a heuristic gloss upon underlying computational description.

Chalmers (2012) argues that the functional conception neglects important features of computation. As he notes, computational models usually describe more than just input-output relations. They describe intermediate steps through which inputs are transformed into outputs. These intermediate steps, which Marr consigns to the "algorithmic" level, figure prominently in computational models offered by logicians and computer scientists.

Restricting the term “computation” to input-output description does not capture standard computational practice.

An additional worry faces functional theories, such as Egan’s, that exclusively emphasize *mathematical* inputs and outputs. Critics complain that Egan mistakenly elevates mathematical functions, at the expense of intentional explanations routinely offered by cognitive science (Burge 2005; Rescorla 2015; Silverberg 2006; Sprevak 2010). To illustrate, suppose perceptual psychology describes the perceptual system as estimating that some object’s depth is 5 meters. The perceptual depth-estimate has a representational content: it is accurate only if the object’s depth is 5 meters. We cite the number 5 to identify the depth-estimate. But our choice of this number depends upon our arbitrary choice of measurement units. Critics contend that the content of the depth-estimate, not the arbitrarily chosen number through which we theorists specify that content, is what matters for psychological explanation. Egan’s theory places the number rather than the content at explanatory center stage. According to Egan, computational explanation should describe the visual system as computing a *particular mathematical function* that carries *particular mathematical inputs* into *particular mathematical outputs*. Those particular mathematical inputs and outputs depend upon our arbitrary choice of measurement units, so they arguably lack the explanatory significance that Egan assigns to them.

We should distinguish the functional approach, as pursued by Marr and Egan, from the *functional programming paradigm* in computer science. The functional programming paradigm models evaluation of a complex function as successive evaluation of simpler functions. To take a simple example, one might evaluate $f(x, y) = (x^2 + y)$ by first evaluating the squaring function and then evaluating the addition function. Functional programming differs from the “computational level” descriptions emphasized by Marr, because it specifies intermediate computational stages. The functional programming paradigm stretches back to Alonzo Church’s (1936) *lambda calculus*, continuing with programming languages such as PCF and LISP. It plays an important role in AI and theoretical computer science. Some authors suggest that it offers special insight into mental computation (Klein 2012; Piantadosi, Tenenbaum, and Goodman 2012). However, many computational formalisms do not conform to the functional paradigm: Turing machines; imperative programming languages, such as C; logic programming languages, such as Prolog; and so on. Even though the functional paradigm describes numerous important computations (possibly including mental computations), it does not plausibly capture computation *in general*.

6.3 Structuralism

Many philosophical discussions embody a *structuralist conception of computation*: a computational model describes an abstract causal structure, without taking into account particular physical states that instantiate the structure. This conception traces back at least

to Putnam's original treatment (1967). Chalmers (1995, 1996a, 2011, 2012) develops it in detail. He introduces the *combinatorial-state automaton* (CSA) formalism, which subsumes most familiar models of computation (including Turing machines and neural networks). A CSA provides an abstract description of a physical system's *causal topology*: the pattern of causal interaction among the system's parts, independent of the nature of those parts or the causal mechanisms through which they interact. Computational description specifies a causal topology.

Chalmers deploys structuralism to delineate a very general version of CTM. He assumes the functionalist view that psychological states are individuated by their roles in a pattern of causal organization. Psychological description specifies causal roles, abstracted away from physical states that realize those roles. So psychological properties are *organizationally invariant*, in that they supervene upon causal topology. Since computational description characterizes a causal topology, satisfying a suitable computational description suffices for instantiating appropriate mental properties. It also follows that psychological description is a species of computational description, so that computational description should play a central role within psychological explanation. Thus, structuralist computation provides a solid foundation for cognitive science. Mentality is grounded in causal patterns, which are precisely what computational models articulate.

Structuralism comes packaged with an attractive account of the *implementation relation* between abstract computational models and physical systems. Under what conditions does a physical system implement a computational model? Structuralists say that a physical system implements a model just in case the model's causal structure is "isomorphic" to the model's formal structure. A computational model describes a physical system by articulating a formal structure that mirrors some relevant causal topology. Chalmers elaborates this intuitive idea, providing detailed necessary and sufficient conditions for physical realization of CSAs. Few if any alternative conceptions of computation can provide so substantive an account of the implementation relation.

We may instructively compare structuralist computationalism with some other theories discussed above:

Machine functionalism. Structuralist computationalism embraces the core idea behind machine functionalism: mental states are functional states describable through a suitable computational formalism. Putnam advances CTM as an empirical hypothesis, and he defends functionalism on that basis. In contrast, Chalmers follows David Lewis (1972) by grounding functionalism in the conceptual analysis of mentalistic discourse. Whereas Putnam defends functionalism by defending computationalism, Chalmers defends computationalism by assuming functionalism.

Classical computationalism, connectionism, and computational neuroscience. Structuralist computationalism emphasizes organizationally invariant descriptions, which are multiply realizable. In that respect, it diverges from computational neuroscience. Structuralism is compatible with both classical and connectionist computationalism, but it differs in spirit from those views. Classicists and connectionists present their rival positions as bold, substantive hypotheses. Chalmers advances structuralist computationalism as a relatively minimalist position unlikely to be disconfirmed.

Intentional realism and eliminativism. Structuralist computationalism is compatible with both positions. CSA description does not explicitly mention semantic properties such as reference, truth-conditions, representational content, and so on. Structuralist computationalists need not assign representational content any important role within scientific psychology. On the other hand, structuralist computationalism does not preclude an important role for representational content.

The formal-syntactic conception of computation. Wide content depends on causal-historical relations to the external environment, relations that outstrip causal topology. Thus, CSA description leaves wide content underdetermined. Narrow content presumably supervenes upon causal topology, but CSA description does not explicitly mention narrow contents. Overall, then, structuralist computationalism prioritizes a level of formal, non-semantic computational description. In that respect, it resembles FSC. On the other hand, structuralist computationalists need not say that computation is "insensitive" to semantic properties, so they need not endorse all aspects of FSC.

Although structuralist computationalism is distinct from CTM+FSC, it raises some similar issues. For example, Rescorla (2012) denies that causal topology plays the central explanatory role within cognitive science that structuralist computationalism dictates. He suggests that externalist intentional description rather than organizationally invariant description enjoys explanatory primacy. Coming from a different direction, computational neuroscientists will recommend that we forego organizationally invariant descriptions and instead employ more neurally specific computational models. In response to such objections, Chalmers (2012) argues that organizationally invariant computational description yields explanatory benefits that neither intentional description nor neurophysiological description replicate: it reveals the underlying mechanisms of cognition (unlike intentional description); and it abstracts away from neural implementation details that are irrelevant for many explanatory purposes.

6.4 Mechanistic theories

The mechanistic nature of computation is a recurring theme in logic, philosophy, and cognitive science. Gualtiero Piccinini (2007, 2012, 2015) and Marcin Milkowski (2013) develop this theme into a mechanistic theory of computing systems. A *functional*

mechanism is a system of interconnected components, where each component performs some function within the overall system. *Mechanistic explanation* proceeds by decomposing the system into parts, describing how the parts are organized into the larger system, and isolating the function performed by each part. A computing system is a functional mechanism of a particular kind. On Piccinini's account, a computing system is a mechanism whose components are functionally organized to process vehicles in accord with rules. Echoing Putnam's discussion of multiple realizability, Piccinini demands that the rules be *medium-independent*, in that they abstract away from the specific physical implementations of the vehicles. Computational explanation decomposes the system into parts and describes how each part helps the system process the relevant vehicles. If the system processes discretely structured vehicles, then the computation is digital. If the system processes continuous vehicles, then the computation is analog. Milkowski's version of the mechanistic approach is similar. He differs from Piccinini by pursuing an "information-processing" gloss, so that computational mechanisms operate over information-bearing states. Milkowski and Piccinini deploy their respective mechanistic theories to defend computationalism.

Mechanistic computationalists typically individuate computational states non-semantically. They therefore encounter worries about the explanatory role of representational content, similar to worries encountered by FSC and structuralism. In this spirit, Shagrir (2014) complains that mechanistic computationalism does not accommodate cognitive science explanations that are simultaneously computational and representational. The perceived force of this criticism will depend upon one's sympathy for content-involving computationalism.

6.5 Pluralism

We have surveyed various contrasting and sometimes overlapping conceptions of computation: classical computation, connectionist computation, neural computation, formal-syntactic computation, content-involving computation, information-processing computation, functional computation, structuralist computation, and mechanistic computation. Each conception yields a different form of computationalism. Each conception has its own strengths and weaknesses. One might adopt a *pluralistic* stance that recognizes distinct legitimate conceptions. Rather than elevate one conception above the others, pluralists happily employ whichever conception seems useful in a given explanatory context. Edelman (2008) takes a pluralistic line, as does Chalmers (2012) in his most recent discussion.

The pluralistic line raises some natural questions. Can we provide a general analysis that encompasses all or most types of computation? Do all computations share certain characteristic marks with one another? Are they perhaps instead united by something like

family resemblance? Deeper understanding of computation requires us to grapple with these questions.

7. Arguments against computationalism

CTM has attracted numerous objections. In many cases, the objections apply only to specific versions of CTM (such as classical computationalism or connectionist computationalism). Here are a few prominent objections. See also the entry the Chinese room argument for a widely discussed objection to classical computationalism advanced by John Searle (1980).

7.1 Triviality arguments

A recurring worry is that CTM is *trivial*, because we can describe almost any physical system as executing computations. Searle (1990) claims that a wall implements *any* computer program, since we can discern some pattern of molecular movements in the wall that is isomorphic to the formal structure of the program. Putnam (1988: 121–125) defends a less extreme but still very strong triviality thesis along the same lines. Triviality arguments play a large role in the philosophical literature. Anti-computationalists deploy triviality arguments against computationalism, while computationalists seek to avoid triviality.

Computationalists usually rebut triviality arguments by insisting that the arguments overlook constraints upon computational implementation, constraints that bar trivializing implementations. The constraints may be counterfactual, causal, semantic, or otherwise, depending on one's favored theory of computation. For example, David Chalmers (1995, 1996a) and B. Jack Copeland (1996) hold that Putnam's triviality argument ignores counterfactual conditionals that a physical system must satisfy in order to implement a computational model. Other philosophers say that a physical system must have representational properties to implement a computational model (Fodor 1998: 11–12; Ladyman 2009; Sprevak 2010) or at least to implement a content-involving computational model (Rescorla 2013, 2014b). The details here vary considerably, and computationalists debate amongst themselves exactly which types of computation can avoid which triviality arguments. But most computationalists agree that we can avoid any devastating triviality worries through a sufficiently robust theory of the implementation relation between computational models and physical systems.

Pancomputationalism holds that every physical system implements a computational model. This thesis is plausible, since any physical system arguably implements a sufficiently trivial computational model (e.g., a one-state finite state automaton). As Chalmers (2011) notes, pancomputationalism does not seem worrisome for

computationalism. What would be worrisome is the much stronger triviality thesis that almost every physical system implements almost every computational model.

For further discussion of triviality arguments and computational implementation, see the entry computation in physical systems.

7.2 Gödel's incompleteness theorem

According to some authors, Gödel's incompleteness theorems show that human mathematical capacities outstrip the capacities of any Turing machine (Nagel and Newman 1958). J.R. Lucas (1961) develops this position into a famous critique of CCTM. Roger Penrose pursues the critique in *The Emperor's New Mind* (1989) and subsequent writings. Various philosophers and logicians have answered the critique, arguing that existing formulations suffer from fallacies, question-begging assumptions, and even outright mathematical errors (Bowie 1982; Chalmers 1996b; Feferman 1996; Lewis 1969, 1979; Putnam 1975: 365–366, 1994; Shapiro 2003). There is a wide consensus that this criticism of CCTM lacks any force. It may turn out that certain human mental capacities outstrip Turing-computability, but Gödel's incompleteness theorems provide no reason to anticipate that outcome.

7.3 Limits of computational modeling

Could a computer compose the *Eroica* symphony? Or discover general relativity? Or even replicate a child's effortless ability to perceive the environment, tie her shoelaces, and discern the emotions of others? Intuitive, creative, or skillful human activity may seem to resist formalization by a computer program (Dreyfus 1972, 1992). More generally, one might worry that crucial aspects of human cognition elude computational modeling, especially classical computational modeling.

Ironically, Fodor promulgates a forceful version of this critique. Even in his earliest statements of CCTM, Fodor (1975: 197–205) expresses considerable skepticism that CCTM can handle all important cognitive phenomena. The pessimism becomes more pronounced in his later writings (1983, 2000), which focus especially on *abductive reasoning* as a mental phenomenon that potentially eludes computational modeling. His core argument may be summarized as follows:

(1) Turing-style computation is sensitive only to "local" properties of a mental representation, which are exhausted by the identity and arrangement of the representation's constituents.
(2) Many mental processes, paradigmatically abduction, are sensitive to "nonlocal" properties such as relevance, simplicity, and conservatism.
(3) Hence, we may have to abandon Turing-style modeling of the relevant processes.

(4) Unfortunately, we have currently have no idea what alternative theory might serve as a suitable replacement.

Some critics deny (1), arguing that suitable Turing-style computations can be sensitive to "nonlocal" properties (Schneider 2011; Wilson 2005). Some challenge (2), arguing that typical abductive inferences are sensitive only to "local" properties (Carruthers 2003; Ludwig and Schneider 2008; Sperber 2002). Some concede step (3) but dispute step (4), insisting that we have promising non-Turing-style models of the relevant mental processes (Pinker 2005). Partly spurred by such criticisms, Fodor elaborates his argument in considerable detail. To defend (2), he critiques theories that model abduction by deploying "local" heuristic algorithms (2005: 41–46; 2008: 115–126) or by positing a profusion of domain-specific cognitive modules (2005: 56–100). To defend (4), he critiques various theories that handle abduction through non-Turing-style models (2000: 46–53; 2008), such as connectionist networks.

The scope and limits of computational modeling remain controversial. We may expect this topic to remain an active focus of inquiry, pursued jointly with AI.

7.4 Temporal arguments

Mental activity unfolds in time. Moreover, the mind accomplishes sophisticated tasks (e.g., perceptual estimation) very quickly. Many critics worry that computationalism, especially classical computationalism, does not adequately accommodate temporal aspects of cognition. A Turing-style model makes no explicit mention of the time scale over which computation occurs. One could physically implement the same abstract Turing machine with a silicon-based device, or a slower vacuum-tube device, or an even slower pulley-and-lever device. Critics recommend that we reject CCTM in favor of some alternative framework that more directly incorporates temporal considerations. van Gelder and Port (1995) use this argument to promote a non-computational *dynamical systems framework* for modeling mental activity. Eliasmith (2003, 2013: 12–13) uses it to support his Neural Engineering Framework.

Computationalists respond that we can *supplement* an abstract computational model with temporal considerations (Piccinini 2010; Weiskopf 2004). For example, a Turing machine model presupposes discrete "stages of computation", without describing how the stages relate to physical time. But we can supplement our model by describing how long each stage lasts, thereby converting our non-temporal Turing machine model into a theory that yields detailed temporal predictions. Many advocates of CTM employ supplementation along these lines to study temporal properties of cognition (Newell 1990). Similar supplementation figures prominently in computer science, whose practitioners are quite concerned to build machines with appropriate temporal properties. Computationalists

conclude that a suitably supplemented version of CTM can adequately capture how cognition unfolds in time.

A second temporal objection highlights the contrast between *discrete* and *continuous* temporal evolution (van Gelder and Port 1995). Computation by a Turing machine unfolds in discrete stages, while mental activity unfolds in a continuous time. Thus, there is a fundamental mismatch between the temporal properties of Turing-style computation and those of actual mental activity. We need a psychological theory that describes continuous temporal evolution.

Computationalists respond that this objection assumes what is to be shown: that cognitive activity does not fall into explanatory significant discrete stages (Weiskopf 2004). Assuming that physical time is continuous, it follows that mental activity unfolds in continuous time. It does *not* follow that cognitive models must have continuous temporal structure. A personal computer operates in continuous time, and its physical state evolves continuously. A complete physical theory will reflect all those physical changes. But our *computational* model does not reflect every physical change to the computer. Our computational model has discrete temporal structure. Why assume that a good cognitive-level model of the mind must reflect every physical change to the brain? Even if there is a continuum of evolving *physical* states, why assume a continuum of evolving *cognitive* states? The mere fact of continuous temporal evolution does not militate against computational models with discrete temporal structure.

7.5 Embodied cognition

Embodied cognition is a research program that draws inspiration from the continental philosopher Maurice Merleau-Ponty, the perceptual psychologist J.J. Gibson, and other assorted influences. It is a fairly heterogeneous movement, but the basic strategy is to emphasize links between cognition, bodily action, and the surrounding environment. See Varela, Thompson, and Rosch (1991) for an influential early statement. In many cases, proponents deploy tools of dynamical systems theory. Proponents typically present their approach as a radical alternative to computationalism (Chemero 2009; Kelso 1995; Thelen and Smith 1994). CTM, they complain, treats mental activity as static symbol manipulation detached from the embedding environment. It neglects myriad complex ways that the environment causally or constitutively shapes mental activity. We should replace CTM with a new picture that emphasizes continuous links between mind, body, and environment. Agent-environment dynamics, not internal mental computation, holds the key to understanding cognition. Often, a broadly eliminativist attitude towards intentionality propels this critique.

Computationalists respond that CTM allows due recognition of cognition’s embodiment. Computational models can take into account how mind, body, and environment

continuously interact. After all, computational models can incorporate sensory inputs and motor outputs. There is no obvious reason why an emphasis upon agent-environment dynamics precludes a dual emphasis upon internal mental computation (Clark 2014: 140–165; Rupert 2009). Computationalists maintain that CTM can incorporate any legitimate insights offered by the embodied cognition movement. They also insist that CTM remains our best overall framework for explaining numerous core psychological phenomena.

Bibliography

Arjo, D., 1996, "Sticking Up for Oedipus: Fodor on Intentional Generalizations and Broad Content", *Mind and Language*, 11: 231–245.

Aydede, M., 1998, "Fodor on Concepts and Frege Puzzles", *Pacific Philosophical Quarterly*, 79: 289–294.

——, 2005, "Computationalism and Functionalism: Syntactic Theory of Mind Revisited", in *Turkish Studies in the History and Philosophy of Science*, G. Irzik and G. Güzeldere (eds), Dordrecht: Springer.

Aydede, M. and P. Robbins, 2001, "Are Frege Cases Exceptions to Intentional Generalizations?", *Canadian Journal of Philosophy*, 31: 1–22.

Bechtel, W. and A. Abrahamsen, 2002, *Connectionism and the Mind*, Malden: Blackwell.

Bermúdez, J.L., 2005, *Philosophy of Psychology: A Contemporary Introduction*, New York: Routledge.

——, 2010, *Cognitive Science: An Introduction to the Science of the Mind*, Cambridge: Cambridge University Press.

Block, N., 1978, "Troubles With Functionalism", *Minnesota Studies in the Philosophy of Science*, 9: 261–325.

——, 1981, "Psychologism and Behaviorism", *Philosophical Review*, 90: 5–43.

——, 1983, "Mental Pictures and Cognitive Science", *Philosophical Review* 92: 499–539.

——, 1986, "Advertisement for a Semantics for Psychology", *Midwest Studies in Philosophy*, 10: 615–678.

——, 1990, "Can the Mind Change the World?", in *Meaning and Method: Essays in Honor of Hilary Putnam*, G. Boolos (ed.), Cambridge: Cambridge University Press.

——, 1995, *The Mind as the Software of the Brain*, in *Invitation to Cognitive Science, vol. 3: Thinking*, E. Smith and B. Osherson (eds), Cambridge: MIT Press.

Block, N. and J. Fodor, 1972, "What Psychological States Are Not", *The Philosophical Review*, 81: 159–181.

Boden, M., 1991, "Horses of a Different Color?", in Ramsey et al. 1991: 3-19.

Bontly, T., 1998, "Individualism and the Nature of Syntactic States", *The British Journal for the Philosophy of Science*, 49: 557–574.

Bowie, G.L., 1982, "Lucas's Number is Finally Up", *Journal of Philosophical Logic*, 11: 79–285.

Brogan, W., 1990, *Modern Control Theory*, 3rd edition. Englewood Cliffs: Prentice Hall.

Burge, T., 1982, “Other Bodies”, in *Thought and Object*, A. Woodfield (ed.), Oxford: Oxford University Press. Reprinted in Burge 2007: 82-99.
——, 1986, “Individualism and Psychology”, *The Philosophical Review*, 95: 3–45. Reprinted in Burge 2007: 221-253.
——, 1989, “Individuation and Causation in Psychology”, *Pacific Philosophical Quarterly*, 70: 303–322. Reprinted in Burge 2007: 316-333.
——, 1995, “Intentional Properties and Causation”, in *Philosophy of Psychology*, C. MacDonald and G. MacDonald (eds), Oxford: Blackwell. Reprinted in Burge 2007: 334-343.
——, 2005, “Disjunctivism and Perceptual Psychology”, *Philosophical Topics*, 33: 1–78.
——, 2007, *Foundations of Mind*, Oxford: Oxford University Press.
——, 2010a, *Origins of Objectivity*, Oxford: Oxford University Press.
——, 2010b, “Origins of Perception”, *Disputatio*, 4: 1–38.
——, 2010c, “Steps Towards Origins of Propositional Thought”, *Disputatio*, 4: 39–67.
——, 2013, *Cognition through Understanding*, Oxford: Oxford University Press.
Camp, E., 2009, “A Language of Baboon Thought?”, in *The Philosophy of Animal Minds*, R. Lurz (ed), Cambridge: Cambridge University Press.
Carruthers, P., 2003, “On Fodor’s Problem”, *Mind and Language*, 18: 508–523.
Chalmers, D., 1990, “Syntactic Transformations on Distributed Representations”, *Connection Science*, 2: 53–62.
——, 1993, “Why Fodor and Pylyshyn Were Wrong: The Simplest Refutation”, *Philosophical Psychology*, 63: 305–319.
——, 1995, “On Implementing a Computation”, *Minds and Machines*, 4: 391–402.
——, 1996a, “Does a Rock Implement Every Finite State Automaton?”, *Synthese*, 108: 309–333.
——, 1996b, “Minds, Machines, and Mathematics”, *Psyche*, 2: 11–20.
——, 2002, “The Components of Content”, in *Philosophy of Mind: Classical and Contemporary Readings*, D. Chalmers (ed.), Oxford: Oxford University Press.
——, 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.
Cheney, D. and R. Seyfarth, 2007, *Baboon Metaphysics: The Evolution of a Social Mind*, Chicago: University of Chicago Press.
Chomsky, N., 1965, *Aspects of the Theory of Syntax*, Cambridge: MIT Press.
Church, A., 1936, “An Unsolvable Problem of Elementary Number Theory”, *American Journal of Mathematics*, 58: 345–363.
Churchland, P.M., 1981, “Eliminative Materialism and the Propositional Attitudes”, *Journal of Philosophy*, 78: 67–90.
——, 1989, *A Neurocomputational Perspective: The Nature of Mind and the Structure of Science*, Cambridge: MIT Press.

——, 1995, *The Engine of Reason, the Seat of the Soul*, Cambridge: MIT Press.
——, 2007, *Neurophilosophy At Work*, Cambridge: Cambridge University Press.
Churchland, P.S., 1986, *Neurophilosophy*, Cambridge: MIT Press.
Churchland, P.S., C. Koch, and T. Sejnowski, 1990, "What Is Computational Neuroscience?", in *Computational Neuroscience*, E. Schwartz (ed.), Cambridge: MIT Press.
Churchland, P.S. and T. Sejnowski, 1992, *The Computational Brain*, Cambridge: MIT Press.
Clark, A., 2014, *Mindware: An Introduction to the Philosophy of Cognitive Science*, Oxford: Oxford University Press.
Clayton, N., N. Emery, and A. Dickinson, 2006, "The Rationality of Animal Memory: Complex Caching Strategies of Western Scrub Jays", in *Rational Animals?*, M. Nudds and S. Hurley (eds), Oxford: Oxford University Press.
Copeland, J., 1996, "What is Computation?", *Synthese*, 108: 335–359.
Cover, T. and J. Thomas, 2006, *Elements of Information Theory*, Hoboken: Wiley.
Crane, T., 1991, "All the Difference in the World", *Philosophical Quarterly*, 41: 1–25.
Crick, F. and C. Asanuma, 1986, "Certain Aspects of the Anatomy and Physiology of the Cerebral Cortex", in McClelland et al. 1987: 333-371.
Cummins, R., 1989, *Meaning and Mental Representation*, Cambridge: MIT Press.
Davidson, D., 1980, *Essays on Actions and Events*, Oxford: Clarendon Press.
Dayan, P., 2009, "A Neurocomputational Jeremiad", *Nature Neuroscience*, 12: 1207.
Dennett, D., 1971, "Intentional Systems", *Journal of Philosophy*, 68: 87–106.
——, 1987, *The Intentional Stance*, Cambridge: MIT Press.
——, 1991, "Mother Nature versus the Walking Encyclopedia", in Ramsey, et al. 1991: 21-30.
Dietrich, E., 1989, "Semantics and the Computational Paradigm in Cognitive Psychology", *Synthese*, 79: 119–141.
Donahoe, J., 2010, "Man as Machine: A Review of *Memory and Computational Brain*, by C.R. Gallistel and A.P. King", *Behavior and Philosophy*, 38: 83–101.
Dreyfus, H., 1972, *What Computers Can't Do*, Cambridge: MIT Press.
——, 1992, *What Computers Still Can't Do*, Cambridge: MIT Press.
Dretske, F., 1981, *Knowledge and the Flow of Information*, Oxford: Blackwell.
——, 1993, "Mental Events as Structuring Causes of Behavior", in *Mental Causation*, J. Heil and A. Mele (eds), Oxford: Clarendon Press.
Edelman, S., 2008, *Computing the Mind*, Oxford: Oxford University Press.
——, 2014, "How to Write a 'How a Build a Brain' Book", *Trends in Cognitive Science*, 18: 118–119.
Egan, F., 1991, "Must Psychology be Individualistic?", *Philosophical Review*, 100: 179–203.
——, 1992, "Individualism, Computation, and Perceptual Content", *Mind*, 101: 443–459.
——, 1999, "In Defense of Narrow Mindedness", *Mind and Language*, 14: 177–194.

——, 2003, “Naturalistic Inquiry: Where Does Mental Representation Fit In?”, in *Chomsky and His Critics*, L. Antony and N. Hornstein (eds), Malden: Blackwell.
——, 2010, “A Modest Role for Content”, *Studies in History and Philosophy of Science*, 41: 253–259.
——, 2014, “How to Think about Mental Content”, *Philosophical Studies*, 170: 115–135.
Eliasmith, C., 2003, “Moving Beyond Metaphors: Understanding the Mind for What It Is”, *Journal of Philosophy*, 100: 493–520.
——, 2013, *How to Build a Brain*, Oxford: Oxford: University Press.
Eliasmith, C. and C.H. Anderson, 2003, *Neural Engineering: Computation, Representation and Dynamics in Neurobiological Systems*, Cambridge: MIT Press.
Elman, J., 1990, “Finding Structure in Time”, *Cognitive Science*, 14: 179–211.
Feferman, S., 1996, “Penrose’s Gödelian Argument”, *Psyche*, 2: 21–32.
Feldman, J. and D. Ballard, 1982, “Connectionist Models and their Properties”, *Cognitive Science*, 6: 205–254.
Field, H., 2001, *Truth and the Absence of Fact*, Oxford: Clarendon Press.
Figdor, C., 2009, “Semantic Externalism and the Mechanics of Thought”, *Minds and Machines*, 19: 1–24.
Fodor, J., 1975, *The Language of Thought*, New York: Thomas Y. Crowell.
——, 1980, “Methodological Solipsism Considered as a Research Strategy in Cognitive Psychology”, *Behavioral and Brain Science*, 3: 63–73. Reprinted in Fodor 1981: 225-253.
——, 1981, *Representations*, Cambridge: MIT Press.
——, 1983, *The Modularity of Mind*, Cambridge: MIT Press.
——, 1987, *Psychosemantics*, Cambridge: MIT Press.
——, 1990, *A Theory of Content and Other Essays*, Cambridge: MIT Press.
——, 1991, “A Modal Argument for Narrow Content”, *Journal of Philosophy*, 88: 5–26.
——, 1994, *The Elm and the Expert*, Cambridge: MIT Press.
——, 1998, *Concepts*, Oxford: Clarendon Press.
——, 2000, *The Mind Doesn’t Work That Way*, Cambridge: MIT Press.
——, 2005, “Reply to Steven Pinker ‘So How Does the Mind Work?’”, *Mind and Language*, 20: 25–32.
——, 2008, *LOT2*, Oxford: Clarendon Press.
Fodor, J. and Z. Pylyshyn, 1988, “Connectionism and Cognitive Architecture: A Critical Analysis”, *Cognition*, 28: 3–71.
Frege, G., 1879/1967, *Begriffsschrift, eine der Arithmetischen Nachgebildete Formelsprache des Reinen Denkens*. Reprinted as *Concept Script, a Formal Language of Pure Thought Modeled upon that of Arithmetic*, in *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*, J. van Heijenoort (ed.), S. Bauer-Mengelberg (trans.), Cambridge: Harvard University 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.

Gandy, R., 1980, “Church’s Thesis and Principles for Mechanism”, in *The Kleene Symposium*, J. Barwise, H. Keisler, and K. Kunen (eds). Amsterdam: North Holland.
Gödel, K., 1936/65. “On Formally Undecidable Propositions of Principia Mathematica and Related Systems”, Reprinted with a new Postscript in *The Undecidable*, M. Davis (ed.), New York: Raven Press Books.
Grice, P., 1989, *Studies in the Ways of Words*, Cambridge: Harvard University Press.
Hadley, R., 2000, “Cognition and the Computational Power of Connectionist Networks”, *Connection Science*, 12: 95–110.
Harnish, R., 2002, *Minds, Brains, Computers*, Malden: Blackwell.
Haykin, S., 2008, *Neural Networks: A Comprehensive Foundation*, New York: Prentice Hall.
Haugeland, J., 1985, *Artificial Intelligence: The Very Idea*, Cambridge: MIT Press.
Horgan, T. and J. Tienson, 1996, *Connectionism and the Philosophy of Psychology*, Cambridge: MIT Press.
Horowitz, A., 2007, “Computation, External Factors, and Cognitive Explanations”, *Philosophical Psychology*, 20: 65–80.
Johnson, K., 2004, “On the Systematicity of Language and Thought”, *Journal of Philosophy*, 101: 111–139.
Johnson-Laird, P., 1988, *The Computer and the Mind*, Cambridge: Harvard University Press.
——, 2004, “The History of Mental Models”, in *Psychology of Reasoning: Theoretical and Historical Perspectives*, K. Manktelow and M.C. Chung (eds), New York: Psychology Press.
Kazez, J., 1995, “Computationalism and the Causal Role of Content”, *Philosophical Studies*, 75: 231–260.
Kelso, J., 1995, *Dynamic Patterns*, Cambridge: MIT Press.
Klein, C., 2012, “Two Paradigms for Individuating Implementations”, *Journal of Cognitive Science*, 13: 167–179.
Ladyman, J., 2009, “What Does it Mean to Say that a Physical System Implements a Computation?”, *Theoretical Computer Science*, 410: 376–383.
Lewis, D., 1969, “Lucas against Mechanism”, *Philosophy*, 44: 231–3.
——, 1972, “Psychophysical and Theoretical Identifications”, *Australasian Journal of Philosophy*, 50: 249–58.
——, 1979, “Lucas Against Mechanism II”, *Canadian Journal of Philosophy*, 9: 373–376.
——, 1994, “Reduction of Mind”, in *A Companion to the Philosophy of Mind*, S. Guttenplan (ed.), Oxford: Blackwell.
Lizier, J., B. Flecker, and P. Williams, 2013, “Towards a Synergy-based Account of Measuring Information Modification”, *Proceedings of the 2013 IEEE Symposium on Artificial Life (ALIFE)*, Singapore: 43–51.
Ludwig, K. and S. Schneider, 2008, “Fodor’s Critique of the Classical Computational Theory of Mind”, *Mind and Language*, 23: 123–143.
Lucas, J.R., 1961, “Minds, Machines, and Gödel”, *Philosophy*, 36: 112–137.

MacLennan, B., 2012, “Analog Computation”, *Computational Complexity*, R. Meyers (ed.), New York: Springer.
McClelland, J., D. Rumelhart, and G. Hinton, 1986, “The Appeal of Parallel Distributed Processing”, in Rumelhart et al. 1986: 3-44.
McClelland, J., D. Rumelhart, and the PDP Research Group, 1987, *Parallel Distributed Processing*, vol. 2. Cambridge: MIT Press.
McCulloch, W. and W. Pitts, 1943, “A Logical Calculus of the Ideas Immanent in Nervous Activity”, *Bulletin of Mathematical Biophysics*, 7: 115–133.
McDermott, D., 2001, *Mind and Mechanism*, Cambridge: MIT Press.
Marcus, G., 2003, *The Algebraic Mind*, Cambridge: MIT Press.
Marr, D., 1982, *Vision*, San Francisco: W.H. Freeman.
Mendola, J., 2008, *Anti-Externalism*, Oxford: Oxford University Press.
Milkowski, M., 2013, *Explaining the Computational Mind*, Cambridge: MIT Press.
Mole, C., 2014, “Dead Reckoning in the Desert Ant: A Defense of Connectionist Models”, *Review of Philosophy and Psychology*, 5: 277–290.
Nagel, E. and J.R. Newman, 1958, *Gödel’s Proof*, New York: New York University Press.
Newell, A., 1990, *Unified Theories of Cognition*, Cambridge: Harvard University Press.
Newell, A. and H. Simon, 1956, “The Logic Theory Machine: A Complex Information Processing System”, *IRE Transactions on Information Theory, IT-2*, 3: 61–79.
——, 1976, “Computer Science as Empirical Inquiry: Symbols and Search”, *Communications of the ACM*, 19: 113–126.
O’Keefe, J. and L. Nadel, 1978, *The Hippocampus as a Cognitive Map*, Oxford: Clarendon University Press.
Ockham, W., 1957, *Summa Logicae*, in his *Philosophical Writings, A Selection*, P. Boehner (ed. and trans.), London: Nelson.
Peacocke, C., 1992, *A Study of Concepts*, Cambridge: MIT Press.
——, 1993, “Externalist Explanation”, *Proceedings of the Aristotelian Society*, 67: 203–230.
——, 1994, “Content, Computation, and Externalism”, *Mind and Language*, 9: 303–335.
——, 1999, “Computation as Involving Content: A Response to Egan”, *Mind and Language*, 14: 195–202.
Penrose, R., 1989, *The Emperor’s New Mind: Concerning Computers, Minds, and the Laws of Physics*, Oxford: Oxford University Press.
Perry, J., 1998, “Broadening the Mind”, *Philosophy and Phenomenological Research*, 58: 223–231.
Piantadosi, S., J. Tenenbaum, and N. Goodman, 2012, “Bootstrapping in a Language of Thought”, *Cognition*, 123: 199–217.
Piccinini, G., 2004, “Functionalism, Computationalism, and Mental States”, *Studies in History and Philosophy of Science*, 35: 811–833.
——, 2007, “Computing Mechanisms”, *Philosophy of Science*, 74: 501–526.
——, 2008a, “Computation Without Representation”, *Philosophical Studies*, 137: 205–241.

——, 2008b, “Some Neural Networks Compute, Others Don’t”, *Neural Networks*, 21: 311–321.
——, 2010, “The Resilience of Computationalism”, *Philosophy of Science*, 77: 852–861.
——, 2012, “Computationalism”, in *The Oxford Handbook of Philosophy and Cognitive Science*, E. Margolis, R. Samuels, and S. Stich (eds), Oxford: Oxford University Press.
——, 2015, *Physical Computation: A Mechanistic Account*, Oxford: Oxford University Press.
Piccinini, G. and A. Scarantino, 2010, “Computation vs. Information processing: Why their Difference Matters to Cognitive Science”, *Studies in History and Philosophy of Science*, 41: 237–246.
Piccinini, G. and S. Bahar, 2013, “Neural Computation and the Computational Theory of Cognition”, *Cognitive Science*, 37: 453–488.
Piccinini, G. and O. Shagrir, 2014, “Foundations of Computational Neuroscience”, *Current Opinion in Neurobiology*, 25: 25–30.
Pinker, S., 2005, “So How Does the Mind Work?”, *Mind and Language*, 20: 1–24.
Pinker, S. and A. Prince, 1988, “On Language and Connectionism”, *Cognition*, 28: 73–193.
Putnam, H., 1967, “Psychophysical Predicates”, in *Art, Mind, and Religion*, W. Capitan and D. Merrill (eds), Pittsburgh: University of Pittsburgh Press. Reprinted in Putnam 1975 as “The Nature of Mental States”: 429-440.
——, 1975, *Mind, Language, and Reality: Philosophical Papers, vol. 2*, Cambridge: Cambridge University Press.
——, 1983, *Realism and Reason: Philosophical Papers*, vol. 3. Cambridge: Cambridge University Press.
——, 1988, *Representation and Reality*, Cambridge, MA: MIT Press.
——, 1994, “The Best of All Possible Brains?”, *The New York Times*, November 20, 1994: 7.
Pylyshyn, Z., 1984, *Computation and Cognition*, Cambridge, MA: MIT Press.
Quine, W.V.O., 1960, *Word and Object*, Cambridge, MA: MIT Press.
Ramsey, W., S. Stich, and D. Rumelhart (eds), 1991, *Philosophy and Connectionist Theory*, Hillsdale: Lawrence Erlbaum Associates.
Rescorla, M., 2009a, “Chrysippus’s Dog as a Case Study in Non-Linguistic Cognition”, In *The Philosophy of Animal Minds*, R. Lurz (ed.), Cambridge: Cambridge University Press.
——, 2009b, “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.
——, 2013, “Against Structuralist Theories of Computational Implementation”, *British Journal for the Philosophy of Science*, 64: 681–707.

——, 2014a, “The Causal Relevance of Content to Computation”, *Philosophy and Phenomenological Research*, 88: 173–208.
——, 2014b, “A Theory of Computational Implementation”, *Synthese*, 191: 1277–1307.
——, 2015, “Bayesian Perceptual Psychology”, in *The Oxford Handbook of the Philosophy of Perception*, M. Matthen (ed.), Oxford: Oxford University Press.
——, forthcoming, “From Ockham to Turing—and Back Again”, in *Turing 100: Philosophical Explorations of the Legacy of Alan Turing*, (*Boston Studies in the Philosophy and History*), A. Bokulich and J. Floyd (eds), Springer.
Rogers, T. and J. McClelland, 2014, “Parallel Distributed Processing at 25: Further Explorations of the Microstructure of Cognition”, *Cognitive Science*, 38: 1024–1077.
Rumelhart, D., 1989, “The Architecture of Mind: A Connectionist Approach”, in *Foundations of Cognitive Science*, M. Posner (ed.), Cambridge: MIT Press.
Rumelhart, D., G. Hinton, and R. Williams, 1986, “Learning Representations by Back-propagating Errors”, *Nature*, 323: 533–536.
Rumelhart, D. and J. McClelland, 1986, “PDP Models and General Issues in Cognitive Science”, in Rumelhart et al. 1986: 110-146.
Rumelhart, D., J. McClelland, and the PDP Research Group, 1986, *Parallel Distributed Processing*, vol. 1. Cambridge: MIT Press.
Rupert, R., 2008, “Frege’s Puzzle and Frege Cases: Defending a Quasi-Syntactic Solution”, *Cognitive Systems Research*, 9: 76–91.
——, 2009, *Cognitive Systems and the Extended Mind*, Oxford: Oxford University Press.
Russell, S. and P. Norvig, 2010, *Artificial Intelligence: A Modern Approach*, 3rd ed., New York: Prentice Hall.
Sawyer, S., 2000, “There Is No Viable Notion of Narrow Content”, in *Contemporary Debates in Philosophy of Mind*, B. McLaughlin and J. Cohen (eds), Malden: Blackwell.
Schneider, S., 2005, “Direct Reference, Psychological Explanation, and Frege Cases”, *Mind and Language*, 20: 423–447.
——, 2011, *The Language of Thought: A New Philosophical Direction*, Cambridge: MIT Press.
Searle, J., 1980, “Minds, Brains, and Programs”, *Behavioral and Brain Sciences*, 3: 417–457.
——, 1990, “Is the Brain a Digital Computer?”, *Proceedings and Addresses of the American Philosophical Association*, 64: 21–37.
Segal, G., 2000, *A Slim Book About Narrow Content*, Cambridge: MIT Press.
Shagrir, O., 2001, “Content, Computation, and Externalism”, *Mind*, 110: 369–400.
——, 2006, “Why We View the Brain as a Computer”, *Synthese*, 153: 393–416.
——, 2014, “Review of *Explaining the Computational Theory of Mind*, by Marcin Milkowski”, *Notre Dame Review of Philosophy*, January 2014.
Shannon, C., 1948, “A Mathematical Theory of Communication”, *Bell System Technical Journal* 27: 379–423, 623–656.

Shapiro, S., 2003, “Truth, Mechanism, and Penrose’s New Argument”, *Journal of Philosophical Logic*, 32: 19–42.
Shea, N., 2013, “Naturalizing Representational Content”, *Philosophy Compass*, 8: 496–509.
Sieg, W., 2009, “On Computability”, in *Philosophy of Mathematics*, A. Irvine (ed.), Burlington: Elsevier.
Siegelmann, H. and E. Sontag, 1995, “On the Computational Power of Neural Nets”, *Journal of Computer and Science Systems*, 50: 132–150.
Silverberg, A., 2006, “Chomsky and Egan on Computational Theories of Vision”, *Minds and Machines*, 16: 495–524.
Sloman, A., 1978, *The Computer Revolution in Philosophy*, Hassocks: The Harvester Press.
Smolensky, P., 1988, “On the Proper Treatment of Connectionism”, *Behavioral and Brain Sciences*, 11: 1–74.
——, 1991, “Connectionism, Constituency, and the Language of Thought”, in *Meaning in Mind: Fodor and His Critics*, B. Loewer and G. Rey (eds), Cambridge: Blackwell.
Sperber, D., 2002, “In Defense of Massive Modularity”, in *Language, Brain, and Cognitive Development: Essays in Honor of Jacques Mehler*, E. Dupoux (ed.), Cambridge: MIT Press.
Sprevak, M., 2010, “Computation, Individuation, and the Received View on Representation”, *Studies in History and Philosophy of Science*, 41: 260–270.
Stalnaker, R., 1999, *Context and Content*, Oxford: Oxford University Press.
Stich, S., 1983, *From Folk Psychology to Cognitive Science*, Cambridge: MIT Press.
Thelen, E. and L. Smith, 1994, *A Dynamical Systems Approach to the Development of Cognition and Action*, Cambridge: MIT Press.
Thrun, S., W. Burgard, and D. Fox, 2006, *Probabilistic Robotics*, Cambridge: MIT Press.
Thrun, S., M. Montemerlo, and H. Dahlkamp, et al., 2006, “Stanley: The Robot That Won the DARPA Grand Challenge”, *Journal of Field Robotics*, 23: 661–692.
Tolman, E., 1948, “Cognitive Maps in Rats and Men”, *Psychological Review*, 55: 189–208.
Trappenberg, T., 2010, *Fundamentals of Computational Neuroscience*, Oxford: Oxford University Press.
Turing, A., 1936, “On Computable Numbers, with an Application to the Entscheidungsproblem”, *Proceedings of the London Mathematical Society*, 42: 230–265.
——, 1950, “Computing Machinery and Intelligence”, *Mind*, 49: 433–460.
van Gelder, T., 1990, “Compositionality: A Connectionist Variation on a Classical Theme”,*Cognitive Science*, 14: 355–384.
van Gelder, T. and R. Port, 1995, “It’s About Time: An Overview of the Dynamical Approach to Cognition”, in *Mind as Motion: Explorations in the Dynamics of Cognition*, R. Port and T. van Gelder (eds), Cambridge: MIT Press.

Varela, F., Thompson, E. and Rosch, E., 1991, *The Embodied Mind: Cognitive Science and Human Experience*, Cambridge: MIT Press.
von Neumann, J., 1945, "First Draft of a Report on the EDVAC", Moore School of Electrical Engineering, University of Pennsylvania. Philadelphia, PA.
Wakefield, J., 2002, "Broad versus Narrow Content in the Explanation of Action: Fodor on Frege Cases", *Philosophical Psychology*, 15: 119–133.
Weiskopf, D., 2004, "The Place of Time in Cognition", *British Journal for the Philosophy of Science*, 55: 87–105.
Whitehead, A.N. and B. Russell, 1925, *Principia Mathematica*, vol. 1, 2nd ed., Cambridge: Cambridge University Press.
Wilson, R., 2005, "What Computers (Still, Still) Can't Do", in *New Essays in Philosophy of Language and Mind*, R. Stainton, M. Ezcurdia, and C.D. Viger (eds). *Canadian Journal of Philosophy*, supplementary issue 30: 407–425.
Yablo, S., 1997, "Wide Causation", *Philosophical Perspectives*, 11: 251–281.
——, 2003, "Causal Relevance", *Philosophical Issues*, 13: 316–327.
Zylberberg, A., S. Dehaene, P. Roelfsema, and M. Sigman, 2011, "The Human Turing Machine", *Trends in Cognitive Science*, 15: 293–300.

Academic Tools

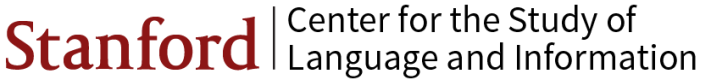
Other Internet Resources

- Horst, Steven, "The Computational Theory of Mind", *Stanford Encyclopedia of Philosophy* (Summer 2015 Edition), Edward N. Zalta (ed.), URL = <https://plato.stanford.edu/archives/sum2015/entries/computational-mind/>. [This is the previous entry on the Computational Theory of Mind in the *Stanford Encyclopedia of Philosophy* — see the version history.]
- Marcin Milkowski, "The Computational Theory of Mind," in the *Internet Encyclopedia of Philosophy*.
- Bibliography on philosophy of artificial intelligence, in Philpapers.org.

Related Entries

analogy and analogical reasoning | anomalous monism | causation: the metaphysics of | Chinese room argument | Church-Turing Thesis | cognitive science | computability and complexity | computation: in physical systems | computer science, philosophy of | computing: modern history of | connectionism | culture: and cognitive science | folk psychology: as mental simulation | frame problem | functionalism | Gödel, Kurt | Gödel, Kurt: incompleteness theorems | Hilbert, David: program in the foundations of mathematics | language of thought hypothesis | mental causation | mental content: causal theories of | mental content: externalism about | mental content: narrow | mental content: teleological theories of | mental imagery | mental representation | mental representation: in medieval philosophy | mind/brain identity theory | models in science | multiple realizability | other minds | reasoning: automated | reasoning: defeasible | reduction, scientific | simulations in science | Turing, Alan | Turing machines | Turing test | zombies

Open access to the Encyclopedia has been made possible, in part, with a financial contribution from the University of Wisconsin System Libraries. We gratefully acknowledge this support.



Library of Congress Catalog Data: ISSN 1095-5054