3

Machinery of Cognition

Charles R. Gallistel

Abstract

A Darwinian approach to decision-making mechanisms must focus on the representation of the options between which the animal decides. For example, in matching behavior, is the animal deciding between the different locations in which to forage or simply whether to leave its current location? A neurobiologically informed approach must be concerned with the mechanism of representation itself. In the computational theory of mind, options are represented by the symbols that carry information about them forward in time. In conventional computing machines, symbols reside in an addressable read-write memory. Current theorizing about the neurobiological mechanism of memory rejects this form of memory in favor of an associative memory. The problem is that the associative bond—and its neurobiological embodiment, the plastic synapse—is not suited to the function of carrying acquired information forward in time in a computationally accessible form. It is argued that this function is indispensable. Therefore, there must be such a mechanism in neural tissue, most probably realized at the molecular level.

Introduction

The computational theory of mind—that brains perceive the world and control behavior through computation—is the central doctrine of cognitive science. It is now widely though not universally embraced by neuroscientists; however, there is, no consensus about its neuroscientific implications. Does it imply that the brain has the architecture of a computer? Or does it imply that the brain has its own architecture, which is in some way superior to that of a computer? These questions are relevant to an evolutionary approach to decision making, because the brain is the organ that makes decisions, and those decisions depend on a representation of the options between which the decision is to be taken; what is not represented cannot be decided on. Moreover, if the animal's brain represents the decision to be made in a way other than the way in which the experimentalist and theorist conceive of it, the choices made may seem paradoxical and nonoptimal.

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

Before considering the machinery that makes decision-relevant representation possible—the machinery of memory—I review contrasting models of matching behavior. I do this to illustrate the importance in an evolutionary approach to mechanisms of decision making of two considerations that figured prominently in the discussions at the Forum: the correct identification of (a) the animal's representation of the options and (b) the function of the behavior.

When animals forage at more than one nearby location, moving back and forth between them, the durations of their visits are exponentially distributed, with expectations whose ratios approximately match the ratios of their current incomes from those locations (Heyman 1982; Gibbon 1995; Gallistel et al. 2001). The distinction between income and return is important: The income from a location is the amount of food obtained there per unit overall foraging time; its computation is independent of the amount of time spent there. The return is the amount of food obtained per unit of time spent at there; that is, per unit of time invested there. Matching equates the returns, not the incomes. It is often taken as more or less self-evident that (a) the function of the behavior is to maximize overall income, (b) the options between which the animal decides are the two locations, and (c) the decision-making strategy is based on the relative returns. When the return from one option is perceived to be greater than the return from the other, the animal adjusts its strategy so as to prolong its visits to the location with the higher return and shorten its visits to the location with the lower return (Herrnstein and Vaughan 1980; Vaughan 1981; Herrnstein and Prelec 1991), a strategy called melioration.

On the basis of data from my lab, we have suggested that all of these assumptions are wrong (Gallistel et al. 2001, 2007). First, the function of the behavior is to gather information about each location (what kinds of food are currently to be found there in what amounts and at what intervals). On this view, the harvesting of found food is ancillary; the animal simply eats what it finds (or not, depending on its motivational state). Second, the options between which the animal decides during a visit are whether to leave that location or not. Third, the decision to leave depends only on the incomes, not the returns.

Reviewing the evidence for our arguments and the model of matching that they lead to would digress too long from my main theme, which is the machinery that makes the representations possible. In this evolutionary context, however, I note that one of our arguments is that matching is not optimal from an income-maximizing perspective (Heyman and Luce 1979), but it is optimal from an information-gathering perspective (Gallistel et al. 2007).

The Machinery of Memory

In the computational theory of mind, options and decision variables are represented by symbols in the brain (Gallistel 1990; Gallistel and King 2009). For example, in most contemporary reinforcement learning models, decisions

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

depend at a minimum on the values of different options and, at a maximum, on a model of the relevant aspects of experienced environment (see Dayan, this volume). In a computing machine, symbols reside in memory. Their function is to carry acquired information forward in time (Gallistel and King 2009). The function of a symbol in memory is analogous to that of an action potential in an axon. However, the function of the action potential is to carry information from one location in the brain to another, whereas the function of the symbol is to carry information from an earlier location in time to a later location—in a computationally accessible form. In the world of the computer scientist, this function is implemented by means of an addressable read-write memory.

The problem is that most contemporary neuroscientists would not regard an addressable read-write memory as the mechanism by which options and evidence are represented in nervous systems. In most neuroscientific theorizing, altered synaptic conductances redirect the flow of signals between the input and the output. In the computer scientist's conception of memory, its most basic property is its ability to encode acquired information (write to memory) and redeliver it upon request (read). In at least the strongest version of the neuroscientist's conception of memory, the encoding of information is not on the list of the properties that the memory mechanism is thought to possess. In this latter view, the rewiring of a plastic brain by experience explains the effect of past experience on future behavior. The rewiring is mediated by changes in synaptic conductances. This theory is the extension to neuroscience of the associative theory of learning. The alterations in synaptic conductances are the material realization of the associative bond. This theory explicitly rejects the assumption that an addressable read-write memory is central to the machinery of cognition.

There are two schools of thought about whether the rewiring of the brain by experience implies that the brain has acquired the information about the experienced world that is implicit in the altered behavior. One school of thought holds that the brain does not store information in the explicit form in which a computer does; the brain is "sub-symbolic" (Smolensky 1986). This is consonant with the anti-representational stance that has always been a salient feature of associative and behaviorist theorizing in a materialist framework. In associative theorizing, the only elements of the brain that undergo enduring structural change are the associative bonds. The associative bond, however, was never intended to carry information forward in time. Associative theories of learning do not attempt to explain how associative bonds encode the information extracted from experience by sensory/perceptual processing, precisely because the associative bond is not conceived of as a carrier of information. Neither it nor its material realization, the plastic synapse, can readily be adapted to this function (Gallistel and King 2009).

Another school of thought holds that changes in synaptic conductances do, in some sense, encode information. This information, however, is distributed across the synapses of a complex multielement circuit in such a way that one

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

cannot readily specify how or where it is encoded. It is, so to speak, lost in a cloud. Martin and Morris (2000:650) explain this cloud view as follows:

> [Long-term potentiation] LTP may serve a universal function in the encoding and storage of memory traces, but what gets encoded and how is an emergent property of the network in which this plasticity is embedded, rather than of the mechanisms operating at the synapse in isolation.

Similarly, Neves, Cooke, and Bliss (2008) write:

> Two facts about the hippocampus have been common currency among neuroscientists for several decades. First, lesions of the hippocampus in humans prevent the acquisition of new episodic memories; second, activity-dependent synaptic plasticity is a prominent feature of hippocampal synapses. Given this background, the hypothesis that hippocampus-dependent memory is mediated, at least in part, by hippocampal synaptic plasticity has seemed as cogent in theory as it has been difficult to prove in practice. Here we argue that the recent development of transgenic molecular devices will encourage a shift from mechanistic investigations of synaptic plasticity in single neurons toward an analysis of how networks of neurons encode and represent memory, and we suggest ways in which this might be achieved. In the process, the hypothesis that synaptic plasticity is necessary and sufficient for information storage in the brain may finally be validated.

While Koch (1997) puts it this way:

> And what of memory? It is everywhere (but can't be randomly accessed). It resides in the concentration of free calcium in dendrites and the cell body; in the presynaptic terminal; in the density and exact voltage-dependency of the various ionic conductances; and in the density and configuration of specific proteins in the postsynaptic terminals.

In considering whether an addressable read-write memory is central to the functioning of a computational brain, one must understand the role it plays in computation. Computation is the composition of functions. A logical constraint of fundamental importance in shaping the architecture of a computing machine is that functions of arbitrarily many arguments may be realized through the composition of functions of two arguments, but not through the composition of functions of one argument. Examples of two-argument functions are the basic logical operations, AND and OR, the basic arithmetic operations ≥, +, –, *, ÷, and the CAT (concatenation). Examples of one-argument functions are NOT, log, sine, and abs. This logical constraint explains why an architecture consisting of one or more central processors coupled to a read-write (aka fetch/store) memory is a universal feature of engineered computing machines (Figure 3.1). The results of computations performed on earlier inputs to the machine are stored in memory (written). When they are needed in subsequent computations, they are retrieved from memory (read). The read-write memory frees composition from the constraints of space and time. It allows information acquired at different times to come together to inform current behavior.

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

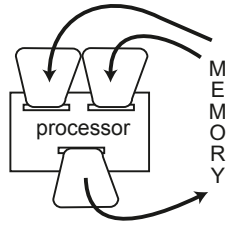

Figure 3.1 The essential architecture of a computing machine. The physical realization of the symbols that may become the arguments (inputs) to a two-argument function reside in memory. They are brought from memory to the machinery that effects the function (the processor). The result (processor output) is returned to memory, where it is carried forward in time so that it may become an argument of a further function. Fetching the inputs from memory is the read operation. Returning the result to memory is the write operation. The symbols to be fetched are located by means of their addresses. It is this architecture that makes possible the composition of functions in all known computing machines.

Although some computer scientists have suggested dispensing with this architecture (e.g., Backus 1978), it has not been discarded. There is currently little thought within computer science that it will or can be. Even highly parallel architectures, like the NUMA architecture in multiprocessor supercomputers, do not dispense with read-write memory. Nor do discussions of quantum computing envisage its absence if such machines are ever built. If any company that manufactures computing machines contemplates building one without a read-write memory, they are keeping it a closely guarded secret.

The constraint that leads to this architecture is physical and geometric: a large number of different symbols may serve as inputs to the combinatorial machinery (the processor). The number of the pairwise combinations grows as the square of the number of individuals that might be paired. This is why, in the design of computing machines going back to Babbage, the symbols to be processed have been stored in a readable memory, whence they are fetched by the processing machinery. It is possible to configure solids in three dimensions so that arbitrarily many are adjacent (Wilson 2002). Neurons, which can intertwine in complex ways, might be thought to be well suited to this manyfold spatial adjacency. There are, however, additional constraints beyond the basic constraint that the participants in a physical interaction must be at the same location at the same time (the spatiotemporal adjacency constraint). The constituent structures must encode information; that is, they must be physically realized symbols, like the coding sequences in DNA, and some of them must be capable of the basic combinatorial operations that underlie neural computation, whatever those basic operations may be. In short, to dispense with a read-write memory in a computing machine, it would be necessary to replace it with an architecture in which coding structures and processing structures were intertwined in such a way that there was universal adjacency of coding

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

structures (every coding structure was adjacent to every other) and every pair was adjacent to every elementary processing structure. To my knowledge, no one knows whether such architecture is possible.

The passages quoted earlier from the works of leading computational neuroscientists would seem to suggest that neuroscientists believe not only that such an architecture is possible, but also that it is realized in the brain. These passages make clear, however, that neuroscientists are frustratingly vague about what this alternative architecture is—too vague to offer guidance, for example, to the engineers at Intel. They are particularly vague about how the different bits of information gleaned from experience at different times are encoded in such a way that they may be retrieved on demand to serve as the inputs to the computations that inform the daily behavior of animals. Thus, there exists at this time a conceptual chasm between what seems to be required by the behavioral facts, some of which I review below, and what neuroscientists are inclined to believe about how memory works in the brain. Several leading cognitive scientists recently put the problem this way (Griffiths et al. 2010:363):

> In our view, the single biggest challenge for theoretical neuroscience is not to understand how the brain implements probabilistic inference, but how it represents the structured knowledge over which such inference is defined.

The problem that Griffiths et al. refer to arises because there is no plausible read-write addressable memory mechanism known to contemporary neuroscience. That is why theoretical neuroscientists generally try to find a way of doing computations without such a mechanism. They sometimes find this so nearly impossible that they end up positing, for example, "context units" (Elman 1990)—a form of read-write memory. Hidden units in a net write their current activity state to the context units, which then supply copies of that activity state to all the other hidden units in the next cycle of operation. In making this assumption, however, as in making the back-propagation assumption, neural net modelers surrender all claim to neural plausibility. The widespread use of context units in neural modeling work over the last two decades is a testimony to the seemingly indispensible role that a read-write memory plays in most computations—a role that was already understood by McCulloch and Pitts (1943), who were perhaps the first to conjecture that reverberating loops could serve this function. Whether or not reverberating loops are a plausible mechanism for carrying information forward over short intervals may be argued, but few would argue that they are a plausible mechanism for carrying information forward over hours, days, months, and years. Thus, contemporary neuroscience does not provide us with a mechanism capable of encoding information and carrying it forward over intervals of indefinite duration in a computationally accessible form.

The question arises: Whose problem is this? Is this a problem for cognitive science and computer science? Should they worry that neuroscience does not recognize the existence in neural tissue of an addressable read-write

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

memory mechanism capable, as Griffiths et al. (2010) posit, of "represent[ing] the structured knowledge over which...[behaviorally consequential] inference is defined." Or is this a problem for neuroscience? Should neuroscientists be looking for the addressable read-write memory mechanism that most cognitive scientists take for granted in their theorizing? Put another way, which science should we look to in pondering what may bridge this conceptual chasm in contemporary thinking about the machinery of cognition?

Building on the seminal work of Alan Turing (1936), computer scientists have developed a mathematically rigorous analysis of what is required in a powerful computing machine of any kind. In this analysis, a read-write memory is essential. In addition, as already noted, in all practical computing machines, the contents of memory must also be addressable, for theoretically well-understood reasons. Importantly, this is not pure theory; we are surrounded by working computing machines, all of which rely on addressable read-write memory.

According to most neuroscientists, however, there is no addressable read-write memory mechanism in the brain. Thus, we must ask: Which science provides a more secure foundation for reasoning about how the brain computes the symbols on which experientially informed decision making presumably depends? Relevant to this question is the following list of questions that I believe most theoretical neuroscientists would admit are both foundational and without consensus answers at this time:

- How is information encoded in spike trains (rate, interspike intervals)?
- What are the primitive combinatorial operations in the brain's computational architecture (its instruction set)?
- How does the brain implement the basic arithmetic operations?
- How does it implement variable binding?
- How does it implement data structures?
- How can changes in synaptic conductance encode facts gleaned from experience (e.g., distances, durations, directions)?

It would seem that a science that has yet to answer questions this basic is unable to provide a good foundation for theorizing about the machinery of cognition.

Behavioral Manifestations of the Composition of Functions

Many animals, perhaps most, have a home base from which they venture forth in search of food and reproductive opportunities and to which they must then return (Figure 3.2). This involves navigation, and navigation relies on the storage of information acquired from experience.

Experiments show that if the ant whose outward foraging path and homeward track is shown in Figure 3.2a were captured and displaced into unfamiliar territory, it would nonetheless run the same compass course it ran in returning to its nest (dashed line in Figure 3.2) for approximately the same distance and

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

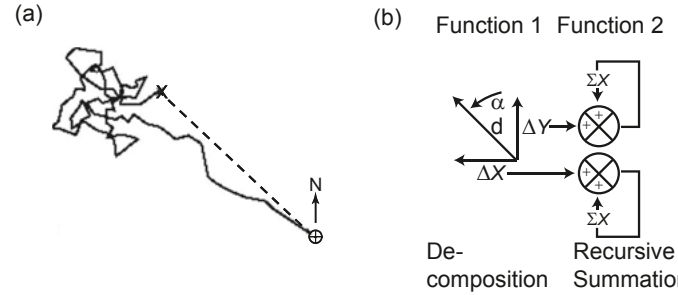

Figure 3.2 (a) Outbound (solid complexly twined line) and home bound (straight dashed line) track of a foraging ant (*Cataglyphis bicolor*) that found food at *X* (Harkness and Maroudas 1985). (b) The dead-reckoning computation that mediates the behavior seen in (a) requires Function 1—the decomposition of successive displacements into orthogonal components (ΔX and ΔY), which then serve as the inputs to Function 2—the recursive summation of successive orthogonal displacements. This second function requires carrying the sums forward in time via memory in a computationally accessible form, because the results of the current displacement are continually added to the sum of the previous displacements.

then begin a search for its nest (Wehner and Srinivasan 1981). This ability to run a prescribed course for a prescribed distance through unfamiliar territory implies dead reckoning. In the large literature on animal navigation, there is a broad consensus that dead reckoning plays a fundamental role. It also provides a clear example of the composition of functions, combining information gleaned from experiences spread out over time, as shown in Figure 3.2b. Dead reckoning is the integration of the velocity vector with respect to time to obtain the position vector as a function of time. In discrete terms, it requires the summation of successive displacement vectors, with the current displacement vector being continually added to the sum of the previous displacement vectors (Function 2 in Figure 3.2b). This recursive summation is itself an instance of the composition of functions; the output of a previous application of the function (the previous sum) serves as an input to the current application of the function. The other input is the current displacement vector. This recursive composition of a simple function (addition) requires a memory mechanism capable of carrying forward in time the information acquired from earlier experience (earlier displacements) in a form that permits that information to be integrated with subsequently acquired information (later displacements).

Animals, including ants and bees, use the Sun to maintain their compass orientation while dead reckoning. To do this, they learn the current solar ephemeris; that is, the compass direction of the Sun as a function of the time of day. The solar ephemeris varies with the season and the latitude of the navigator. Learning it involves the creation of a data structure, combining an encoding of the panorama around the hive or nest, the time of day as indicated by the brain's circadian clock, and the parameters of a universal ephemeris function, a function that specifies what is universally true about the solar ephemeris

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

(Lindauer 1957, 1959; Dyer and Dickinson 1994; Dickinson and Dyer 1996; Budzynski et al. 2000). The information encoded in the solar ephemeris may then be combined with information obtained either from its own foraging expeditions or from attending the dance of another forager to compute the current solar bearing of a food source; that is, the angle to be flown with respect to the Sun (Figure 3.3). Again, we see that information about different aspects of the experienced environment—in this case the solar ephemeris and the location of a food source—gleaned from very different sources by different computations at different times in the animal's past enter into the computation that determines the animal's present behavior. Insofar as we currently understand computational machines, and assuming that the brain is one, then we must conclude that the brain possesses the kind of memory that makes this possible, which is to say, an addressable read-write memory.

Setting courses between known locations is another simple example of the composition of functions with inputs derived from different experiences at different times in the past. Menzel et al. (2011), using radar tracking of individual bee foragers, have recently published the following ingenious experiment (Figure 3.4a): A foraging bee is shaped to visit one feeding station (F1 in Figure 3.4). Another forager is shaped to visit a different station (F2 in Figure 3.4). The first forager is seen to follow the dance of the second forager within the hive, thereby learning from the dance the location of F2. On a subsequent visit to its feeding station (F1), this first forager finds the beaker there empty. It then flies directly from F1 to F2. However, it does so only when the angle between the hive and the two stations is less than some critical value.

As a supplement to the dead reckoning of their location, ants and bees also use compass-oriented snapshots to wend their way through complex environments (Collett 2009). This implies data structures that are more elaborate than location vectors. A data structure is a structured vector or array of symbols

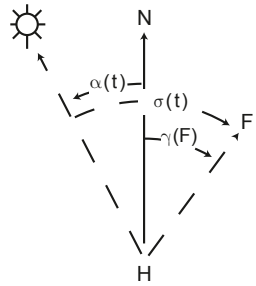

Figure 3.3 Setting a course from the hive to a food source whose location has been stored in memory, either during previous visits to it or from having attended the dance of another forager returning from it, involves the composition of the function σ(t), which gives the compass direction of the Sun as a function of the time indicated by the brain's circadian clock, with the function γ(F), which gives the compass bearing of F from the hive (H), to obtain α(t), the function that gives the angle which must be flown relative to the Sun to get to F from H.

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

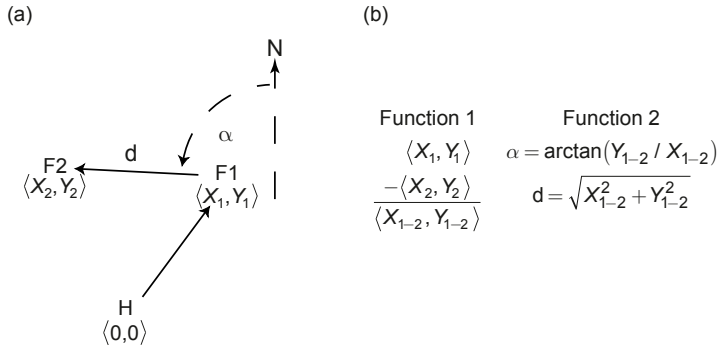

Figure 3.4 Schematic of the Menzel et al. (2011) experiment. (a) The forager learns the location of one feeding station (F1) by direct experience and travels repeatedly back and forth between it and the hive (H), bringing artificial nectar from the beaker at F1 to the hive. While in the hive, it observes the dance of a forager returning from F2, from which it learns the location of F2. On a subsequent visit to F1, it finds the beaker there empty. It then flies directly to F2, but only if the angle ∠F1HF2 is less than a critical value (k). (b) To set its course from F1 to F2, the bee must first compute the difference vector (Function 1), then transform that difference vector into its polar form (Function 2) to get the range (d) and bearing (α) of F1 from F2. In deciding whether to pursue this new course, it checks that the angle ∠F1HF2 is less than a critical value (Function 3). The inputs to Function 1 (the location vectors, $\langle X_1, Y_1 \rangle$ and $\langle X_2, Y_2 \rangle$ for the two feeding sources) come from different kinds of experience (direct vs. communicated) at different times. The location information, which is itself a simple data structure (because vectors are ordered number lists), must have been carried forward in time by a memory mechanism to make it accessible to these functions. This same memory mechanism enables the composition of these functions to inform the observed behavior of the bee.

such that the locations of the symbols within the array encode the relations between the symbols. For example, locations on a two-dimensional surface are encoded by ordered pairs of symbols for quantities: either two distances, if the vectors are Cartesian, or a distance and an angle, if the vector is polar. When we say that these pairs are ordered, we mean that the reference of a symbol depends on the order of the symbols. For example, by convention, latitude is given first, then longitude, or range, then bearing. In the case of a compass-oriented snapshot, the machine must store the snapshot—that is, a vector (string of symbols) that encodes the appearance of the landmark—together with a vector that encodes the compass direction in which the ant was looking when it took that snapshot. Yachtsmen will be familiar with the compass-oriented views and their function in navigation, because they are an important feature of pilot manuals. These manuals contain drawings or photographs of harbor entrances as seen from various approach directions. Recent research shows that for each compass-oriented snapshot, ants remember a sequence of bearing angles (Collett 2010). This implies a multifaceted data structure whose components are the snapshot, the compass orientation of the ant's visual system when

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

the snapshot was made, the panorama behind the snapshot, and the sequence of bearing angles.

As explained below, the addresses in an addressable read-write memory make variable binding possible, and with it, the encoding of arbitrarily complex data structures. Because computational neuroscientists assume that there is no addressable read-write memory mechanism in the brain, variable binding and the creation of complex data structures are unsolved problems in contemporary computational neuroscience (Smolensky 1990; Sun 1992; Lòpez-Moliner and Ma Sopena 1993; Browne and Pilkington 1994; Sougné 1998; Browne and Sun 2001; Frasconia et al. 2002; Gualtiero 2008). When memory is everywhere and nowhere and not addressable, it is hard to see how to use it to bind values to variables.

As a final example, consider the implications of the brilliant series of experiments on food-caching scrub jays conducted by Clayton, Dickinson, and their collaborators in recent years (Clayton and Dickinson 1998, 1999a, b; Clayton et al. 2001b, 2003, 2009; Emery and Clayton 2001a, b; Emory et al. 2004; Dally et al. 2005, 2006). In the wild, these jays make tens of thousands of caches every fall, spread over many square kilometers in their high mountain habitat. They live for months by retrieving the contents of their caches. The experiments from the Clayton and Dickinson laboratory show that jays remember not just *where* they made each cache, but also *what* sort of food they put into it, *when* they made it, and *who* was watching (Figure 3.5). In choosing a cache to visit, they compare how long it has been since they made that cache (subtracting from the current date and time the date and time at which they made the cache) to what they have subsequently learned about how rapidly that particular content (*the what*) takes to rot, the current state of their preference

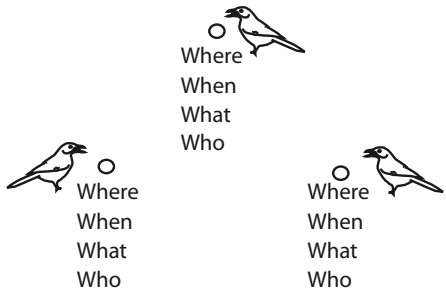

Figure 3.5 The food-caching and retrieving behavior of the scrub jay implies that each time it makes a cache—here only three are shown—it encodes the location of the cache (where), the date and time at which it was made (when), the kind of food cached (what), and which other jays, if any, were watching (who). These separate encodings of different aspects of a single episode must be stored in memory in such a way that the jay can recover the where, the when, the what, and the who for each different cache. This implies a sophisticated data structure. We know how to physically realize such structures—and their utilization—in computing machines, but this is possible *only* in machines with addressable read-write memory.

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

hierarchy for the different kinds of food they have buried, whether they have already emptied that cache, and which potential thieves were watching when they made it. Again, these behavior results imply that the record of each of many thousands of caches is a complex data structure, encoding in recoverable form disparate potentially useful information.

In reviewing their results, Clayton et al. (2006), contrast what they call "mechanistic" accounts of the causation of the jay's behavior (by which they mean associative accounts) with what they call "intentional" or "rational" accounts. They conclude, ruefully, that the intentional account better predicts their experimental findings. I say "ruefully" because Dickinson, at least, is a confirmed associationist. The implication of their contrastive language is that the intentional accounts are not mechanistic accounts. If one assumes that the brain does not have an addressable read-write memory—if one assumes that its memory is purely associative, which is the working assumption in most contemporary theoretical neuroscience—then, indeed, it is hard to see how one is ever going to be able to suggest a plausible reductionist account of these results (an account that specifies the underlying neurobiological mechanisms and the architecture that integrates their functioning). However, if one adopts the computational theory of mind and its materialist implications—namely, that the brain has an addressable read-write memory mechanism because such a mechanism is a sine qua non for an effective computing machine—then to contrast "mechanistic" with "intentional" (read "computational/representational") is bizarre. We know the kind of machine that can implement their intentional account. Their experimental results do not pose any profound computational mysteries. We can build autonomous robots that behave like these scrub jays, but only if we are allowed to put an addressable read-write memory in the robot's onboard computer. If there is a way to build an autonomous robot that mimics the scrub jay's behavior but does not possess an addressable read-write memory, no one knows what it is.

Addressability, Variable Binding, and the Creation of Data Structures

Implementing Variable Binding

The memory elements of an addressable read-write memory are bipartite: one part encodes the information to be carried forward; the other part is the address. This second part enables the machine to find the information conveyed by the first part when it needs it. The bipartite structure of the memory element makes it possible to bind a value to a variable, because the coding part and the address part use the same symbolic currency. The symbols in the coding part of a memory element take the form of bit patterns, strings of 1s and 0s, or, more physically speaking, sequences of binary voltage levels, or directions of magnetization, or some other physical realization of a binary number. Regardless of what the bit pattern encodes, it can be regarded as a number. This

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

is a profoundly important point in thinking about physical (neurobiological) structures that could encode information: if a structure can encode a number, it can encode any kind of information whatsoever. Thus, in thinking about encoding, it suffices to think only about how to encode numbers. The address is also a bit pattern (hence, a number). Thus, the address of one piece of information can be stored in the encoding part of another memory element. When so stored, the address becomes the symbol for the variable. This makes the variable accessible to computation on the same terms as its value. It also implements variable binding. The machine finds the value of a variable by going to the address of the variable, where it finds the address of the value. This is called *indirect addressing*.

In short, giving addresses to memories not only makes them findable, it creates and physically implements the distinction between a variable (e.g., cache location or cache content) and its value (e.g., the vector that specifies the location of a particular cache or its content). This distinction is fundamental to computation. Without it, functions have no generality. Physically implemented widely useful functions, such as addition and multiplication, generate the values of output variables when given the values of input variables.

Building Data Structures

Addressable memory makes possible the creation of the arbitrarily complex data structures that reside in the memories of contemporary computers. Data structures are physically realized by array variables. An array variable is a symbol specifying the first address in a sequence of addresses where the data in a data structure are stored. The machine gets to those addresses by way of a computation performed on the symbol for that address. The computation makes use of the fact that the other addresses in the array are specified by where they are in relation to the first. The linear order of the addresses is used to encode which variable is which. That they occur within the sequence following the array address encodes the fact that these variables all relate to the entity symbolized by the array address. Thus, for example, the symbols at the addresses of the latitude and longitude variables give the addresses of where the values of latitude and longitude can be found for the cache symbolized by the array address.

The Universality of This Manner of Encoding Complex Structure

One might suppose that this way of physically realizing an encoding of complex structure is peculiar to modern computing machines were it not for the fact that complex organic structure is encoded in the structure of DNA in the same way. The gene is the memory element that carries heritable information forward in time, and it, too, has a bipartite structure. The coding part specifies the sequence of amino acids in a protein. The system gains access to the

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.

information in the coding part by means of the other part, the promoter. As in computer memory, the information in the coding part of a gene is more often than not the address of another gene. It is the recipe for constructing a protein called a transcription factor. A transcription factor binds to the promoter of a gene to initiate the transcription of that gene; that is, the decoding of the information that it contains. It has the same function as the address probe in random access computer memory (RAM). As in computer memory, a promoter may give access to a sequence of genes. If a foreign gene is inserted into the sequence, it will be activated when the sequence is activated by a transcription factor that binds to the promoter of that sequence.

The indirect addressing of the information conveyed in the coding parts of genes enables the hierarchical structure in the genetic code. For example, it makes possible a gene for an eye, so that whenever this gene is activated, an eye develops (Halder et al. 1995). The coding part of this gene does not specify the amino acid sequence for any protein in the eye that develops. Instead, it specifies the amino acid sequence of a transcription factor. This transcription factor binds to the promoters of several other genes, which themselves encode transcription factors, and so on, until one gets to genes that encode the proteins that make up the structure of the eye. The analogy to the manner in which data structures are encoded in random access memory is a close one.

If there were a large number of different architectures for the physical realization of the encoding of complex structures—structures with many different components relating to one another in many different ways—then the close analogy between the architecture of information conveyance in DNA and architecture of information conveyance in RAM would be a remarkable coincidence. If, on the other hand, there were only one or a small number of architectures that are physically realizable and effective, then this coincidence is no more remarkable than the striking similarity between the eyes of gastropods, arachnids, and vertebrates. These architectures have evolved independently, but their structure has been strongly constrained by the laws of optics; that is, by the function that they serve.

In contrast to the structure of a synapse, the structure of a DNA molecule is transparently suited to the carrying of information forward in time. It is no mystery how to encode a number in the nucleotides sequence of DNA; hence, it is no mystery how to encode any information whatsoever into the structure of this molecule. Moreover, the structure of DNA is so thermodynamically stable that it endures for years after the animal has died—longer than many computer disks endure. From an engineering standpoint, DNA is a perfect marvel of a universal information-conveying medium. Because DNA and the complex molecular machinery for reading the information it contains already provide most of what is required in an addressable read-write memory—everything but the machinery for writing to it—it is tempting to conjecture that the neurobiological memory mechanism has co-opted either DNA or its close cousin RNA to perform this indispensable function in the machinery of cognition.

From "Evolution and the Mechanisms of Decision Making," edited by Peter Hammerstein and Jeffrey R. Stevens. 2012. Strüngmann Forum Report, vol. 11, J. Lupp, series ed. Cambridge, MA: MIT Press. 978-0-262-01808-1.