

# Material Representations: From the Genetic Code to the Evolution of Cellular Automata

---

Luis Mateus Rocha

School of Informatics

and Cognitive Science Program

Indiana University

1900 East Tenth Street

Bloomington, IN 47406

rocha@indiana.edu

Wim Hordijk

Santa Fe Institute

1399 Hyde Park Road

Santa Fe, NM 86501

wim@santafe.edu

**Abstract** We present a new definition of the concept of representation for cognitive science that is based on a study of the origin of structures that are used to store memory in evolving systems. This study consists of novel computer experiments in the evolution of cellular automata to perform nontrivial tasks as well as evidence from biology concerning genetic memory. Our key observation is that representations require inert structures to encode information used to construct appropriate dynamic configurations for the evolving system. We propose criteria to decide if a given structure is a representation by unpacking the idea of inert structures that can be used as memory for arbitrary dynamic configurations. Using a genetic algorithm, we evolved cellular automata rules that can perform nontrivial tasks related to the density task (or majority classification problem) commonly used in the literature. We present the particle catalogs of the new rules following the computational mechanics framework. We discuss if the evolved cellular automata particles may be seen as representations according to our criteria. We show that while they capture some of the essential characteristics of representations, they lack an essential one. Our goal is to show that artificial life can be used to shed new light on the computation-versus-dynamics debate in cognitive science, and indeed function as a constructive bridge between the two camps. Our definitions of representation and cellular automata experiments are proposed as a complementary approach, with both dynamics and informational modes of explanation.

---

## Keywords

Representation, cellular automata, evolutionary computation, artificial life, cognitive science, symbols, genetic code, memory

---

## 1 Background

The notion of internal representation in cognitive systems and its role in intelligent behavior has been undergoing serious reformulation in cognitive science, ranging from rejection [20, 61, 62] to externalization or leakage into the environment [8]. Ultimately, however, such reformulation must address the familiar computation-versus-dynamics debate.<sup>1</sup> Typically, researchers in cognitive

---

<sup>1</sup> Other terms may be used to designate the computation and dynamics camps. The first is often also identified with terms such as symbolic, representationalist, and information processing. The dynamics camp is also known as emergent, connectionist, self-organization, or embodied.

science deal with this debate by choosing one of these paradigms and pursuing its own research questions without getting any nearer to the central problem of this field: the matter-mind problem:

Adequate explanation in cognitive science must at some stage address the matter-mind problem, that is, the problem of how symbol reference or how the world and our images of the world are coordinated. However, this problem has proved so difficult over the centuries, that not only is there no apparent convergence toward some solution, but there is no consensus on what is required of any solution. So cognitive scientists have a secondary problem of choosing what types of investigation they can do if they can't attack the matter-mind problem directly. [43: pp. 325–326]

Indeed, the success of the paradigms on either side of the feud is that they manage to avoid the problem entirely. In the extreme computationalist camp, there is the assumption that the essential nature of mind is implementation independent—the computationalist approach to artificial life regards the concept of life in precisely the same manner. Mind (or life) is a computational process that can be instantiated in different material embodiments; therefore the research program is directed at describing the higher-level elements and constraints responsible for intelligent (or living) behavior.

In cognitive science, these elements are internal *representations* standing for events and characteristics of the external environment, which can be manipulated independently of, and in lieu of, the actual aspects of the environment they represent. Such representations are also thought of as *symbols*. The research program of this camp is devoted to the study of the rules that can be used to manipulate symbols in order to obtain intelligent behavior, but not usually to the study of how representations come to represent the environment to begin with, or how symbols come to have meaning (symbol grounding [25]), or in the end, how matter becomes mind.

At the root of the computationalist approach to artificial life is the concept of a generalized genotype-phenotype mapping [35], which regards life as an essentially representational process in which genes (symbols) represent phenotypical attributes. Again, the research program of this approach to artificial life is devoted to implementing and studying rules to manipulate generalized genotype-phenotype mappings, but not at the *origin* of such mappings to begin with [44, 58].

Similarly, in the extreme dynamics camp, mind (and life) is an embodied, dynamical process that cannot be separated from the environment and emerges from the nonlinear interaction of many material components. This embodied cognition camp rejects information processing constructs such as representations and symbols, opting instead for a dynamical systems explanatory framework (e.g., [5]). In other words, this camp opts for an account of mind (and life) exclusively as a material process following, and fully describable by, the laws of the dynamics of matter. This way, there is never a need to explain the problem of getting symbols, and ultimately mind (and life), out of matter, since all that there is to describe is the global dynamics of many material components following physical law. In the artificial life context, this camp prefers to regard genes and genotypes as just another set of material components of the dynamics of life. Therefore, the genotype-phenotype mapping is regarded as an artificial construct of the modelers of living systems prone to information-centric modes of explanation.

Clearly, both of these camps are very attractive, since each one chooses only one side of the matter-symbol problem, while avoiding the essential question of the *origin* of symbols from matter in cognitive science and/or artificial life. One can argue that it is plausible that this essential question is ill-posed and inherently dualist (e.g., [57]). But we ultimately communicate (among other modalities) with discrete symbols, which trigger dynamic processes in our brains. Moreover, our current understanding of biology shows that processes of a symbolic nature are essential to evolution (as we discuss below). Given what we know about biology, there is certainly a need to explain the interface between symbols and dynamics in artificial life, which is neglected by theories entertained solely in either one of these camps. We argue here that the study of this interface in artificial life, namely with systems such as the cellular automata simulations in Section 7, can help us gain a better understanding of the concept of representation in cognitive science.

Both the computationalist and dynamics camps reduce the matter-symbol problem to a single dimension: cognition and life are nothing but representations or genotype-phenotype mappings, respectively, instantiated in mostly irrelevant material embodiments (computationalism); or alternatively, cognition, life, representations, and symbols are nothing but the complex attractor behavior of dynamical systems. Those who exclusively pursue either of the two extreme camps often fear that the alternative would necessarily be some veiled dualism, but Pattee has proposed an alternative view based on complementary modes of description:

The problem and the attraction of physics and computation as bases for models is that they are *both* universal, but complementary, modes of description. . . . The issue is whether the description of matter-symbol systems by either mode alone is a satisfactory explanation. The cases at point: Do descriptions of symbol vehicles taken only as matter following laws of motion, have explanatory value; or do descriptions of coordinated dynamics taken only as programs, have explanatory value?" [43: pp. 328–329]

The crux of the complementary view is acknowledging that while symbols must have a material embodiment following laws of dynamics, this is not the same as saying that symbols are “nothing but” matter. As we will discuss in detail, symbolic behavior is an evolutionary property. Thus, symbol function is not a property of the dynamics of material symbol vehicles themselves, but rather arises from arbitrary associations between inert material components (such as DNA) and dynamic regularities (such as those arising from protein interactions). Because the function of material symbols is to carry information and not to act in dynamic processes directly, we require a complementary description of the material components of life. The dynamics of DNA is largely irrelevant to understanding the living organization. It is rather the information DNA stores to produce RNA and proteins that is relevant. On the other hand, the dynamics of proteins (and even RNA) as catalysts and active biochemical players is most relevant. We thus need a complementary view of the living organization that allows for complementary dynamic and information processing modes of description, thus rejecting reductionism of either kind, as well as any non-materialistic dualism.

Using the living organization as a guideline, and artificial life as a laboratory, it is our goal to bring the complementary approach to the symbol-matter problem to embodied cognitive science. Given the nature of information in biology, our first observation is that a complementary approach to cognitive science must deal with the problem of the *origin* of symbols from dynamics in an evolutionary setting. Pattee [43] suggests that cognitive science should tackle such an origin problem by dealing with much simpler matter-symbol systems than the matter-mind problem:

We need simpler embodiments of *natural* matter-symbol systems with both empirical power and conceptual generality. Why should we work only with the ultimate functional complexity of brains, or the ultimate artificiality of computers, or the ultimate meanings of philosophical discourse? As a first trial I suggest trying to adapt our fundamental concepts of cognitive science to the basic symbol-matter problems of biology, and even physics, where a few rungs have already been secured. [43: p. 327]

Others in cognitive science have tried to approach the explanatory shortcomings of the two extreme paradigms above by reworking the concepts of symbol and representation into an essentially dynamicist view of embodied cognition (e.g., [8]). In particular, Wheeler and Clark [64] have proposed the biologically motivated notion of *genic representation* to discuss the need to incorporate information processing elements in complementary models of cognitive behavior. Such models do justice to the evidence supporting the embodied, dynamic view of cognition while also explaining the symbolic dimension that clearly exists in biological organisms.

From the complex systems perspective, Mitchell [39] and Rocha [52, 53] have espoused a similar approach, while further proposing a set of experiments with cellular automata as a simpler, yet

artificial, matter-symbol system to study the origin of representations and computation. In these experiments we may observe the emergence of representations from a dynamical substrate.

Both of these approaches point to the neglected middle ground in cognitive science: the origin and nature of representations. They call for a more explicit study of the different kinds of representations that can exist in biological and complex dynamical systems. Indeed, we know that in biological systems a coded, symbolic form of replication (genetic reproduction) has emerged and evolved out of the dynamics of self-organization [66, 4, 46]. It is at least plausible that other dynamical systems such as embodied cognitive agents have undergone a similar evolutionary process of origin of symbols, and make use of those symbols for their functioning—besides reproduction.

To build empirical knowledge from simpler symbol-matter systems, as Pattee suggests, we need to agree on a set of concepts to contrast several such systems with one another. Artificial life, poised somewhere between theoretical biology and complex systems theory, offers an ideal setting to approach the problem of the origin of symbols from dynamical systems. It allows us to define simpler, artificial matter-symbol systems that we can study computationally in evolutionary experiments, or even possibly with evolutionary robotics. In this sense, artificial life may be used to approach the toughest cognitive science problems with more easily decidable and comparable experiments. In the following, we explore what we should expect natural representations (and their simulations) to be in embodied agents, thus offering criteria to reformulate the concept of representation in cognitive science. We also demonstrate how simulations can be used to study the process of the origin of representations. To this end, we supplement the cellular automata experiments, proposed by Mitchell to study representations, with more difficult tasks, resulting in the emergence of representations with some of the characteristics described by our reformulation—though lacking an important one.

It needs to be stressed that this exercise is not an attempt to generalize the concepts of representation or cognition. We are not interested in claiming that cells or cellular automata are systems with minds. We also do not attempt to study the concept of representation per se, but rather the process of *origin* of representations, symbols, computation, and the like. We are aware that much important work has been done in cognitive science and evolutionary systems theory to define the concept of representation in a biological and evolutionary setting (e.g., [21, 18, 16, 64, 6, 31]), some of which is quite convergent with our view of representation; the work here presented is directed specifically at the study of the origin of representations from a dynamical milieu using the methodology of artificial life and current knowledge of biology. Thus, a comprehensive discussion of the concept of representation in cognitive science is beyond the scope of the present work. Rather, we work with Haugeland's [26] definition of representation, which captures the essence of the concept in cognitive science, using it in contrast to the types of representations that arise in evolving dynamical systems—the origin process we are interested in. The inspiration for this artificial life, bottom-up approach to studying representations is drawn once again from Howard Pattee:

In traditional philosophy epistemic cuts are viewed as problems only at the cognitive level. They are called problems of reference or how symbols come to “stand for” or to “be about” material structures and events [7, 25, 65]. I have always found the complementary problem much more fundamental: How do material structures ever come to be symbolic? I think that if we fully understood how molecules become messages in cells we would have some understanding of how messages have meaning. [44: pp. 25–26]

## 2 Physical Symbol Processing

Since we are interested in simple matter-symbol systems, let us start our discussion with what we know about the physics of representations or symbols. The *physical symbol system hypothesis* put forward by Newell and Simon [41] states that any physical symbol system (of sufficient size) will have the

ability to exhibit intelligent behavior, and, furthermore, that any system that exhibits intelligent behavior necessarily has to be a physical symbol system:

A physical symbol system consists of a set of entities, called symbols, which are physical patterns that occur as components of another type of entity called an expression (or symbol structure). Thus, a symbol structure is composed of a number of instances (or tokens) of symbols related in some physical way (such as one token being next to another). At any instant of time the system will contain a collection of these symbol structures. Besides these structures, the system also contains a collection of processes that operate on expressions to produce other expressions: processes of creation, modification, reproduction and destruction. A physical symbol system is a machine that produces through time an evolving collection of symbol structures. [41]

Given this definition, the physical symbol system hypothesis is simply stated as: “A physical symbol system has the necessary and sufficient means for general intelligent action” [41]. Many attempts at modeling intelligence or cognition are based on this hypothesis, that is, that the underlying system that exhibits intelligent behavior is some form of a physical symbol system. The symbol processing schemes derived from the research program that stemmed from this hypothesis have proven very useful for building artificial intelligence applications. But what this definition of physical symbol systems describes are the mechanisms that can be used to manipulate and process physical structures taken as representations or symbols. It adds nothing to the problem of the origin of symbols or to the empirical constraints of actual matter-symbol systems.

In fact, regarding this origin problem, the hypothesis merely states the obvious: Every symbol system, like any other thing, must follow physical law. It does not specify the conditions in which matter may become symbolic, the constraints that matter and symbol may impose and require from one another, or the conditions in which an information processing model is necessary or preferable over a physical one. Indeed, it does not even clearly distinguish matter from symbol.

Here we study simpler matter-symbol systems in the hopes of answering some of these questions, namely:

1. Which material structures can instantiate representations and symbols?
2. In which conditions can we say that matter becomes symbolic?
3. When is it preferable to use complementary (dynamic or symbolic) modes of description?

### 3 Is Dynamical Systems Theory Enough?

#### 3.1 From Self-Organization to Embodied Cognition

The dynamics approach to cognitive science is rooted in the notion of self-organizing systems. *Self-organization* is seen as the process by which energetically open systems of many components tend to reach a particular state, a set of cycling states, or a small volume of their state space (attractor basins), with no external interference. This attractor behavior is often recognized at a different level of observation as the spontaneous formation of well-organized structures, patterns, or behaviors, from random initial conditions (emergent behavior).

Self-organization is often studied computationally with discrete dynamical systems (state-determined systems) such as Boolean networks or cellular automata. The state-determined transition rules are interpreted as the laws of some physical system [34, 55]: The state of each component depends on the states of its neighbor (or related) components at the previous time instant. It follows from the observed attractor behavior [67] that there is a propensity for matter whose physical law is modeled by the transition rules of dynamical systems theory (DST) to self-organize (e.g., [33]).

In this sense, matter is described by the (lower-level) dynamics of state transitions and the observed (emergent) attractor behavior of self-organization.

The attractor basins of dynamical systems can be used to refer to observables accessible to the self-organizing system in its environment, and thus perform environmental *classifications* (e.g., classifying neural networks). These self-organizing systems do this by producing dynamic stabilities in interaction with their environments, routing situations from this interaction into a small set of attractor basins, which produce a corresponding small set of behaviors. In this sense we say that the self-organizing system *classifies* its environment into a small subset of behaviors. This process of obtaining classifications of an environment by a self-organizing system has been referred to generally as *emergent classification*—“emergent” because it is the global result of the local, state-determined interaction of the basic components of the self-organizing system with its environment. A more detailed discussion of this emergent classification, also known as eigenbehavior, was pursued in [50].

The dynamics or self-organizing approach to cognitive science is rooted in the use of connectionist systems capable of emergent classification, such as neural networks. These systems originate from designs by McCulloch and Pitts [37] meant to model the actual material components of the brain (rather than the higher-level information processing constructs of the physical symbol systems hypothesis). Their original cybernetic designs led to the development of neural network models, and more generally to *parallel distributed processing* (PDP) or connectionist models (e.g., [36]).

Connectionist systems are defined by networks of highly simplified neurons. In an artificial neuron there are a number of input signals carrying the states of other neurons. These inputs are weighted and integrated (typically summed) by the receiving neuron. Finally, the result of this integration is fed into a threshold function that determines the neuron’s own output state, which again serves as input to other neurons. A network of artificial neurons is created by connecting a number of them (feeding the output of one neuron to the inputs of other neurons). A neural network of this type is a state-determined system as defined above. The local weights and connectivity of a network dictate how it self-organizes into a small set of global states, or attractors. We can control the attractor states by tweaking the weights and connectivity with specialized algorithms.

By associating the set of stable attractors that these networks can produce from situations in their environment, we can use the weight-adjusting algorithms to route related situations to the same attractor states. In this sense, we train connectionist systems to classify situations in their environments appropriately.

The main characteristic of these systems is that they do not rely on explicit representations of their environments; in fact, they do not possess symbolic structures holding a memory of environmental situations. Each neuron is a simple state-determined machine producing its next state in strict dynamic fashion, and all neurons produce their next state in parallel. It is only the network as a whole that can be said to classify the environment. If the network possesses a representation of its environment, it is distributed in the collection of weights and connections of the entire network. There are no explicit symbols representing objects or facts in a certain domain, as the classification function is implemented by the *quantitative* state of the entire system [48].

Connectionist architectures turn out to be very successful in performing pattern recognition tasks and often behave in biologically plausible ways, for example, by mirroring the errors of young children in language acquisition. Furthermore, they provide a counterexample to the physical symbol system hypothesis, in that they are capable of performing tasks that we generally view as requiring intelligence, but there does not seem to exist any explicit symbol manipulation governing the classification function in these systems.

Another important characteristic of dynamical systems is rate dependence [60]. One way to think of this is to realize that the components of a connectionist network are dynamically coupled to their environment. In the language of physics, we can say that the network is *dynamically coherent* with its environment. Dynamically, the network and the environment are in fact one system; a change of state in one produces a coordinated, state-determined chain of events in the other—there is a

definite temporal relation between events and components of the network-environment coupling.<sup>2</sup> Thus, it is very difficult to distinguish a connectionist system from its environment. Indeed, the emergent behavior of agents that are dynamically coherent with their environments (embodied agents) is as much a result of the production rules of the agent as of the laws of the environment [5, 8, 55].

Wheeler and Clark [64] refer to this lack of separation between dynamically coherent agents and their environments as *causal spread*. When studying dynamically coherent agent-environment couplings, or embodied cognition, it indeed becomes hard to argue against the radical dynamicist's claim that cognition is best studied with nonrepresentational explanatory devices, except on the grounds that using representational explanations is intuitively more appropriate for us human beings. Are we indeed better off abandoning the notions of representations and symbols altogether and adopting a purely dynamical view of the world, in which cognitive agents are networks of components indistinguishable from their environment?

### 3.2 The Problem of Biological Function

The answer to the previous question depends on whether there are limitations to the current dynamical systems framework [39]—more specifically, whether the reduction of matter-symbol systems to pure dynamical systems, requiring *nothing but* a material or physical mode of description, turns out to be insufficient to explain natural phenomena. Clark [8] and Mitchell [39] point first to the scaling issue. It is not clear how the current dynamical systems models of evolutionary robots, neural networks, and the like, which possess relatively small numbers of components, will scale to systems with many more components. Clark also gives powerful arguments against the ability of current dynamic models of simple cognitive abilities to explain more “representation-hungry” problems.

DST is used to explain natural phenomena by describing the properties of matter that are, by the principles of invariance and symmetry, as independent of observers and individual measurements as possible. These are by definition the universal physical laws of matter [45]. Therefore, dynamical systems equations describe only those aspects of matter that have no context-specific significance for individuals. DST is not equipped to deal with concepts such as biological *function*, *adaptation*, and *selection*. A dynamical systems model needs other tools to explain how a bird's wing functions both as an airfoil and an engine at the same time [56], though it can clearly explain the possible dynamic configurations of the components of the wing. This kind of model also does not allow us to “understand how two adaptive systems with very different dynamical portraits give rise to similar functional behavior” [39].

The problem is that biological function is a selective property of populations of individuals evolving under natural selection. Natural selection can be described as a statistical bias on the rates of reproduction of populations of individuals. But this is as far as (statistical) dynamics theory can take us to describe this process. It can describe which components see their reproduction rates increase in population dynamics trajectories, but it cannot describe the individual adaptive strategies responsible for the increase. It cannot describe how a certain phenotypical characteristic of an organism gives it an advantage in a particular environment. To construct a model of this type of context-specific constraint on the biological organization, the modeled agents require control of initial and boundary conditions, in addition to dynamical equations or state transitions.

What distinguishes biology from physics is that biological organisms, subjected to natural selection, gained some control of constraints to their own dynamics. Obviously, living organisms obey the dynamics of physical law, but unlike nonliving systems, which can be described solely by physical law equations, biological organisms store in genetic memory initial and boundary conditions to produce and reproduce their own dynamics. Below we elaborate further on the fact that genetic

<sup>2</sup> In continuous dynamical systems this temporal relation is governed by the rates of change of variables as expressed in differential equations. In discrete dynamical systems it is governed by the transition functions of all the components. In the latter case, time is replaced by synchronous computation of these functions.

memory is dynamically decoupled from the biochemical dynamics of life, and that the information it stores is used to set up, and thus control, context-specific dynamics. A deeper discussion of this issue has been given by Pattee [43, 45, 46] and further elaborated by Rocha [54].

## 4 Genetic Memory

How genetic information can arise from a purely dynamic biochemical system is still a very hard question—the origin-of-life problem. Interesting models of the origin of the genetic code have been proposed (e.g., [3, 4] or more recently [42, 66]). In any case, the genotype-phenotype distinction in living systems is an observed symbol-matter system, the characteristics of which we discuss in this section.

### 4.1 Evolving Systems Require Selectable Initial Conditions

To evolve in a changing environment, dynamical systems must be able to change their own dynamics in order to produce new basins of attraction responsible for new classifications and behaviors. For natural selection to operate, an evolving dynamical system must have access to distinct configurations—there must be different alternatives to be inherited and subjected to variation [50, 51, 54]. How can a dynamical system store and access different configurations?

Notice that dynamical systems do not truly allow for alternatives. “The only meaning we can attach to a choice of alternatives in a system described by deterministic laws is through measurement and control on initial conditions” [45: p. 15]. The attractor landscape of a given dynamical system, while providing several possible outcomes for the system’s dynamics, does not truly yield alternatives until there is control of the initial conditions that send the dynamical system into one attractor or another. Therefore, we need some kind of *memory* that can store the alternative initial conditions that allow evolving dynamic systems to produce different, inheritable configurations and behaviors. These memories reenter the dynamics only when selected as initial condition constraints. Since we require alternatives to describe the selective process of evolving systems, our explanatory language must include both the dynamics of matter and the memory components that can store selectable initial conditions.

We could of course conceive of another dynamical system, with a different time scale, whose states could set the alternative initial conditions for the first dynamical system. But then, if the entire system were to evolve, we would need yet another dynamical system to store the initial conditions of the previous one, and so forth, in an infinite regression, as von Neumann [63] discussed in his self-reproducing scheme [51]. Therefore, we are better off treating the components that store initial conditions for a given evolving dynamical system as memory. Clearly, significantly distinct time scales play a role in what is considered a dynamic component or static memory for a given evolving dynamical system. Below we elaborate on what kind of inert components can serve as memory for an evolving dynamical system.

What needs to be stressed here is that evolutionary systems require both dynamic and memory components for selection. It is not sufficient to consider exclusively state-determined, rule-following dynamics—which would amount to agents with no selectable alternatives. We need components that can store initial conditions. This complementary view of evolutionary systems leads us to think of what kinds of selection processes are possible given different types of memory used to store initial conditions, a question that was explored in [54]. Here we deal with the characteristics of known biological memory components and simpler artificial constructs. Let us first investigate the characteristics of genetic memory components in biological organisms.

### 4.2 Dynamical Incoherence in the Genetic System

In biology, organisms employ *inert* structures to store initial conditions used to construct appropriate dynamical (biochemically active) components as well as produce other organisms like themselves. But what kind of material structures can be conceived of as inert, and used as memory of initial



conditions for producing and reproducing dynamic agents? These must be material components with many dynamically equivalent states, which can be used to set up an arbitrary (representational) relation with another set of material components that instantiate the dynamic system. The genetic system gives us this insight.

The genotype (set of nucleotide sequences in DNA) of biological organisms is the memory that encodes initial conditions (amino acid chains) used to produce components (proteins), which ultimately self-organize into some dynamical configuration (phenotype). Fundamentally important is that almost any sequence of nucleotides is equally possible, which means that DNA molecules provide many dynamically equivalent states. Furthermore, the informational value of a DNA sequence (genetic information) is independent of the particular dynamic (biochemical) behavior of the DNA molecule that contains the sequence. Genetic information is not expressed by the biochemistry of nucleotide sequences in DNA molecules, but is instead mediated through a code that translates nucleotide sequences into amino acid sequences. Unlike DNA, amino acid sequences do possess a very dynamic biochemistry, which defines phenotypes—the ultimate expression of genetic information in an environment.

It is precisely the biochemical (dynamic) irrelevance of nucleotide sequences (inertness) that makes DNA molecules the ideal memory structure for genetic information given an arbitrary genetic code [44, 59]. DNA qua carrier of genetic information in biological organisms is virtually *dynamically incoherent* with the biochemistry of the organism-environment coupling, since the information needed to construct a given protein (the memory) can be retrieved at any time, much as a book can be retrieved from a library [47].

What does dynamical incoherence mean? Ultimately, all biological components have a biochemical, dynamic substrate. The point is that in a system where certain components are used as memory, their dynamics is irrelevant. It is not their dynamic (biochemical) characteristics that elicit responses from the dynamic components of the system. Rather, it is their informational value, extracted via a code (instantiated by the ribosomes, tRNA, and other machinery in the cell), which is relevant for producing behavior. Indeed, recent organisms carry many of the same genes used by primordial organisms to produce the same proteins. More and more transgenic technology pervades our life; we can use bacteria to produce human proteins such as insulin, or produce plague resistant tomatoes with proteins encoded in genes from pigs. Viruses too are little more than genetic memory that uses the dynamic machinery of host cells to reproduce. All these facts show that in their role of information carriers, the dynamical substrate of genes (biochemistry of DNA molecules) is mostly irrelevant.

Genes can be *decoupled* from a particular organism and introduced into another, where they will be used for their *memory function*, which is the specification of initial conditions that produce particular cell dynamics, thus reentering the dynamics of their host cells. One can argue that we can also introduce a particular protein into an organism to produce particular cell dynamics. But such an action would not be decoupled from the host dynamics, since it would produce a dynamically coherent reaction. In contrast, the introduction of a gene into an organism's genotype, via reverse transcription as in transgenic technology, allows the organism to produce the encoded protein on demand. Essentially, once a gene is encoded in a genome, it can be accessed very much like random-access memory [54], because (as von Neumann [63] observed early on), it is decoupled from the biochemical machinery of the cell. Without genetic information, for a protein to exist in an organism it needs to be constantly introduced—as medication, for instance. With genetic information, the organism can produce the protein from memory as needed.

Now, of course, being instantiated as molecules, genetic memory cannot be completely dynamically incoherent. Dynamical incoherence needs to be understood in a frame of stable existence of the whole dynamic machinery that can decode the memory. Genes can be read by the living cell's machinery, as long as the cell exists as such, very much in the same way as a writing system is useful as long as it is understood by an appropriate reading framework. Therefore, dynamical incoherence is not an absolute concept, but rather a relative one. For a given dynamics, with particular time scales, we can treat certain components as incoherent if their dynamics are

irrelevant for those time scales. We can further treat them as *memory* if such dynamically incoherent components are used as information to specify sets of initial conditions for the original dynamics.

Our understanding of genetic memory naturally depends on the particular evolutionary process of life on Earth. Other forms of evolutionary processes are possible that may not make use of dynamically incoherent memory. A study of different possible evolutionary processes is detailed in Rocha [54], where the advantage of dynamically incoherent memory is discussed. In particular, based on the arguments of von Neumann and Pattee, it is argued that open-ended evolution requires dynamically incoherent memory of initial conditions. For the purpose of this article, it is sufficient to observe that dynamically incoherent memory exists in biology, and its explanation requires complementary dynamics and informational models.

### 4.3 Genetic Memory Instantiates a Matter-Symbol System

The notion of a genotype memory needs to be dissociated from that of a computer program. When we say that the genome encodes the memory to produce the dynamics of a particular phenotype, we are not saying that the genotype completely specifies (or programs) the phenotype. Clearly, genes do not encode the complete specification of proteins and the means to produce them. For example, genes do not encode how to fold a protein. All of this comes for free with the laws of dynamic matter [40]. In other words, genes are not programs that completely specify some dynamics. The initial conditions that genetic representations encode merely start off the dynamics of a set of components; they do not encode the dynamics itself. Moreover, the specification of initial conditions (e.g., amino acid sequences) is only the first step in the development of a given dynamics (e.g., a phenotype). The final dynamical outcome can also be modulated by the environment in many different ways [30, 49]. For instance, the gender of some crocodiles depends on the temperature in the nest [23], rather than being genetically controlled.

Genetic memory is part of a matter-symbol system, which is a different concept from the abstract symbols of the physical symbol system hypothesis. A matter-symbol system entails a *construction code* such as the genetic code. In turn, such a code presupposes a mapping between two sets of material components: a small, finite number of symbols (e.g., codons in DNA), and a finite number of building blocks (e.g., amino acids). The symbol set needs to be instantiated by inert material structures with many dynamically equivalent states, as described above (e.g., DNA molecules). The building blocks, on the other hand, need to produce very dynamical (reactive) products (e.g., proteins). The code itself needs to be instantiated by machinery capable of producing the dynamic products from the memory stored in the material symbols (e.g., the ribosome and tRNA molecules).

It is important to stress that a particular matter-symbol system is tied to specific building blocks. The richer these are, the smaller the required descriptions, but also the smaller the number of constructable products. Conrad [9] referred to this as a tradeoff between programmability and high evolutionary plasticity or efficient use of computational resources. Genetic memory does not need to describe all the dynamical characteristics of a given protein (full programmability); it merely needs to specify an amino acid chain, which will itself self-organize (fold) into the protein with a given biochemistry.

The cost of this efficient ability to specify proteins is that the genetic code is not a universal symbol system, in that it cannot specify anything whatsoever, but only those things that can be constructed from amino acid chains. Genetic memory can store any conceivable amino acid chain built from the twenty amino acids used by the genetic code [17], but nothing else. Therefore, we should think of genetic memory not as a program capable of universal specification, but rather as a material symbol system used to construct building-block-specific dynamic configurations, that is, encoded initial conditions for specific dynamic systems.

## 5 Material Representations

In cognitive science, representations are typically thought of as structures capable of participating in processes in lieu of actual components that they stand for: As Haugeland explains, when features

that are necessary are not present, they may “be represented; that is, something else can stand in for them, with the power to guide behavior in their stead. That which stands in for something else in this way is a *representation*; that which it stands in for is its *content*; and its standing in for that content is *representing it*” [26: p. 62]. Haugeland further requires that only those “stand-ins” that participate in a *representational scheme* are representations. Such a scheme requires that there exist several possible representations and contents; that there be a systematic, consistent code to determine the content of representations; and that there be a systematic syntax for producing, maintaining, and modifying representations.

## 5.1 The Active and Passive Roles of Material Representations

Our view of material representations as observed in the genetic system shares some of the characteristics of representations as defined by Haugeland, but shows a very important difference, which becomes apparent when we discuss the active and passive roles of representations. Genetic memory is *never* used in a dynamical process in lieu of something else. It is simply used, when decoded, for construction of amino acid sequences. Simply put, genes never stand in for proteins in the biochemistry of life. The relation between genes and proteins is rather a reading and construction process—referred to in biology as *transcription* and *translation* processes, respectively. Indeed, the necessity of conceiving the genotype as symbolic memory rather than dynamic matter lies in its particular role as the repository of initial conditions for, and not as a dynamic player in, the biochemistry of life (Section 4). If genetic memory were a dynamic player, we would not need the concept of genetic memory and the complementary approach to biological organization to begin with!

In his self-reproducing scheme, von Neumann [63] referred to the process of constructing an automaton by interpreting a description as the *active role* of descriptions. As discussed in the previous section, we do not regard genetic memory as a computer program that completely specifies a dynamic automaton, as in von Neumann’s self-reproducing automata. However, it is useful to regard the process of reading genetic memory to construct a protein as the *active role of material representations*. What really takes place is a construction and not a standing-in process; therefore, the concept of an active role of representation is elucidative.

Extrapolating from genetic memory to material representations at large, if they occur in other dynamical systems, we argue that representations do not participate in dynamical processes, not even in a standing-in role. Rather, via a code (or Haugeland’s representational scheme), they are (actively) *used for* constructing or specifying dynamic products. More specifically, dynamically incoherent (inert) material structures are used as representations to be decoded into dynamic products. In this active role, representations are indeed used to produce and guide dynamics and behavior, as Haugeland requires, but not by standing in for something else, rather by being used as information to construct something else. We can think of this construction process as the *semantics* of material representation [54]. Instead of conceiving semantics as a substitution relation, using genetic memory as a foundation, we conceive it as a construction process: a mapping from dynamically incoherent material symbols onto dynamic products.

But the material representations of the genetic system function in yet another role, what von Neumann referred to as the *passive role* of descriptions, which turns out to be useful, in fact essential, for open-ended evolution through natural selection [54]. In this role, memory is produced, manipulated, changed, and copied, but without any reference or recourse to its content, that is, without decoding its content. Biological examples of these operations are the transcription of mRNA from DNA, RNA editing processes [2, 49], and—most important for natural selection—mutation and sexual recombination (variation). We can think of the manipulation of material representations without recourse to content as the *syntax* of matter-symbol systems [54]. Notice that in this passive role, representations also do not stand in for content and furthermore are not even used to produce or guide dynamics or behavior, but rather disseminate existing or produce new representations. Because of this role, representations can be communicated and novelty created.

Now, eventually, the ultimate value of material representations is found when their content is expressed in an environment. The environmental ramifications of the encoded constructs, the

phenotypes, are a result of their success in that environment. We can refer to this process as the *pragmatics* of representations: the survival of the encoded dynamics.

## 5.2 Requirements for Material Representations

Genetic memory contains material representations that do not stand in, but are rather used to construct content (active role), and are useful for communication and variation, essential characteristics for open-ended evolution and natural selection (passive role). How does this apply to a more general notion of representation, particularly at the cognitive level? Let us summarize what material representations entail.

*Requirement 1: Dynamically incoherent memory.* Material representations demand inert physical structures which can effectively be seen as dynamically incoherent regarding the overall machinery in which they are utilized as representations. The role of these physical structures is not defined by their dynamic characteristics but rather by their informational value. Let us refer to these structures as symbols. Symbols must have the ability to be manipulated and combined into new symbols. The set of possible symbol manipulations defines the syntax of the representations. Syntax is required for communication and for variation, both essential for natural selection. Biological examples of inert structures are DNA molecules, of material symbols are codons, and of representations are genes, or the portions of DNA that encode specific proteins.

*Requirement 2: Construction code.* There needs to exist machinery to construct arrangements of building blocks (which subsequently self-organize to produce some dynamic behavior) from the representations of Requirement 1—an arbitrary construction code.<sup>3</sup> This code leads us to think of the semantics of representations as a material affair, rather than an abstract, surrogate relationship between internal and external elements of a system. Here, representations are conceptualized as something used to start off, not stand in for, dynamic processes. They are used to (literally) materialize dynamical systems. In our view of material representations, semantics is about construction of dynamics and behavior. This construction is not about correspondence between internal and external elements, but about a material process of controlling the behavior of embodied agents in an environment. Symbols and representations are dynamically incoherent elements that make this control process possible by providing selectable alternatives for dynamic outcomes.

*Requirement 3: Self-organization and selection.* The encoded building blocks, after construction via the code of Requirement 2, self-organize at different levels: The first products of the self-organization of dynamic building blocks (e.g., amino acid chains) become themselves building blocks (e.g., proteins) for other self-organizing processes, and so on. After expression of representations into building blocks, for evolution to occur, a feedback selection mechanism must exist to favor the representations of successful dynamic outcomes: the pragmatics of representations. In natural selection, this feedback is instantiated by increasing rates of reproduction of the more successful agents.

## 5.3 Material Representations for Cognitive Science

We do not offer the notion of material representation as an explanation of the mind, but rather as a set of guidelines to think about representations in the dynamical systems framework of embodied cognition. If material representations exist in cognition, they must be implemented in arrangements of the dynamic components of the brain. First and foremost we need a set of such arrangements that may be considered inert (in the sense described in Section 4) vis-à-vis the dynamics of the brain, to be reliably used as memory. Note that we can restrict the components of such arrangements to neural states, or we can pursue a more situated, extended view of cognition that includes not only bodily but also environmental components [8].

---

<sup>3</sup> To say “arbitrary code” is actually redundant, since arbitrariness is a condition for having a code to begin with [59]. The point is that the content of the (symbolic) representations does not depend on their physical characteristics. There is no known reason to disclaim the idea that the current genetic code could have been set up differently, that is, with different RNA-codon–amino-acid relations defined by tRNA molecules [10, 27].

The three requirements of Section 5.2 give us some important insight into what cognitive representations based on our gene-inspired material representations would be. Material representations are not necessarily something internal to a cognitive agent standing in for something else external. Rather, they are low-level components used to control and produce dynamic behavior more effectively, and may even be meaningless to the overall system that uses them. Just as a single gene may not have a single identifiable phenotypic outcome, material representations do not necessarily represent any high-level observable in a cognitive agent's environment. They should simply be seen as mappings to construct dynamic outcomes from dynamically incoherent symbols (active role), and not as stand-ins for something external.

Our view, however, raises at least two questions: How do cognitive organisms store or organize their high-level organism-environment content? And how can neural states organize to produce the low-level dynamically incoherent representations? The speculative answer to the first is that by analogy with biology, higher-level behavior is produced by a process analogous to biological development, that is, a completely dynamic self-organization of building blocks whose initial conditions are encoded in lower-level representations. To approach the second question a bit less speculatively, in the next section we study the emergence of symbols in dynamical systems simulated by cellular automata.

The advantage of representations for controlling dynamics in living organisms is biological fact. It seems plausible that the same benefits may be discovered by other dynamical systems, including embodied cognitive systems. Since we make use of symbols in human language in much of the same way biology does in the genetic system, it is reasonable to expect that cognition itself makes use of material representations in the brain and its environment. Much research will need to be pursued in order to identify neural-environmental elements that pass the requirements of the previous section. It is beyond the scope of our expertise to propose such research. Nonetheless, using artificial life, we can certainly explore the process of emergence of representations from dynamical systems. In Sections 6 and 7 we study such an emergence process in cellular automata.

## 6 Emergent Computation in Cellular Automata

### 6.1 Evolution of Cellular Automata for Nontrivial Computational Tasks

One-dimensional cellular automata (CAs) are arguably the simplest instances of parallel distributed architectures used to model dynamical systems and self-organization computationally. They consist of a one-dimensional *lattice* of  $N$  identical cells, each a state-determined automaton with  $k$  possible states. Here, only two-state CAs are considered, that is,  $k=2$ . Let  $s_i(t)$  denote the state of cell  $i$  at time  $t$ , with  $s_i \in \{0,1\}$ . Each cell is *connected* to  $2r$  other cells, which we think of as its neighborhood of *radius*  $r$ . Usually, periodic boundary conditions are employed, that is, cells 1 and  $N$  are each other's neighbor. In homogeneous CAs, each cell's automaton is defined by the same local update rule  $\phi$ , which takes as input the cell's neighborhood state,  $\mu_i = (s_{i-r}(t), \dots, s_i(t), \dots, s_{i+r}(t))$ , and maps this to the new state of the cell at time  $t+1$ :  $s_i(t+1) = \phi(\mu_i)$ .

The initial conditions for a CA are defined by a particular *initial configuration* (IC) of (typically random) cell states. In discrete time steps, all the cells subsequently update their state synchronously according to the update rule  $\phi$ . This update rule can be represented by a *lookup table* with one entry for each of the  $2^{2r+1}$  possible neighborhood configurations  $\mu$  and their corresponding output values for  $s(t+1)$ . Given that each output value  $s(t+1)$  can either be a 0 or a 1, there are  $2^{2r+1}$  possible one-dimensional, two-state, radius- $r$  CAs. Here we deal with CA rules of radius  $r=3$ ; thus the lookup table contains 128 entries, and there are  $\approx 3.4 \times 10^{38}$  such rules.

Das et al. [14, 15] used a genetic algorithm to evolve one-dimensional, two-state,  $r=3$  CAs to perform certain computational tasks. A *genetic algorithm* (GA) is a search procedure modeled after natural selection [28]. The GA maintains a population of candidate solutions (phenotypes) to a given problem, usually encoded as bit strings (genotypes). In this case, each CA rule is encoded as a 128 bit string, where each bit encodes the outcome of each entry in the rule's lookup table. Each candidate

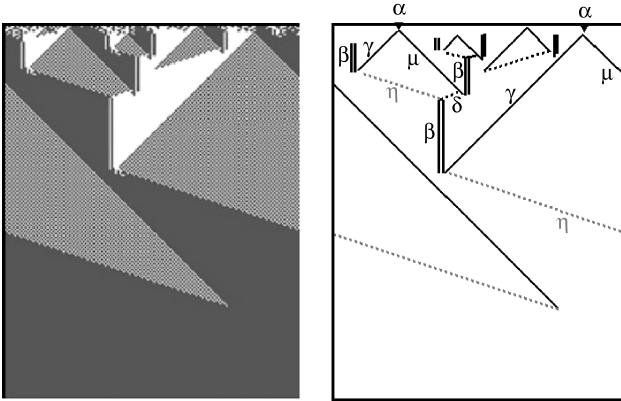


Figure 1. (Left) Space-time diagram for  $\phi_{\text{DMC}}$  given a random IC with a majority of dark cells. The rule correctly classifies this IC in 141 iterations when applied to the density classification task. (Right) Space-time diagram with regular domains filtered out, depicting particles and their interactions after the initial transient is removed.

solution is assigned a fitness value according to how well it solves the given problem. New generations of candidate solutions are created from existing solutions according to fitness scores; the higher the fitness, the greater the chances of a solution being selected for the population of solutions in the next generation. Variation is also applied to the encoded solutions via bit string crossover and random mutation [22, 38]. This way, the GA is biased to evolve good solutions through an idealized model of genetic variation and selection.

CAs have been evolved via GA to perform several computational tasks, such as the *density classification* task (majority classification problem). For this task, the goal is to find a CA that decides whether or not the IC contains a majority of 1's (i.e., has high density). Let  $\rho_0$  denote the density of 1's in the IC. If  $\rho_0 > 1/2$ , then within  $M$  time steps the CA should reach the fixed-point attractor configuration of all 1's (i.e., all cells in state 1 for all subsequent iterations); otherwise, within  $M$  time steps it should reach the fixed-point configuration of all 0's.  $M$  is a parameter of the task that depends on the lattice size  $N$ . Since the CA cells have access only to local interactions (with other cells within radius  $r$ ), this task requires the CA to propagate information across the lattice in order to achieve global coordination. In this sense, the task requires nontrivial computation.

The unbiased performance  $\mathcal{P}_{N,I}(\phi)$  of a CA rule  $\phi$  on a given task is defined as the fraction of  $I$  randomly generated ICs for which  $\phi$  reaches the desired behavior within  $M$  time steps on a lattice of length  $N$ . Here, unless otherwise noted, we employ  $N = 149$ ,  $M = 2N$ , and  $I = 10^5$ .

In the experiments of Das, Mitchell, and Crutchfield [15] one particular run of the GA on the density classification task was discussed. Figure 1a shows the space-time diagram of one of the CA rules evolved:  $\phi_{\text{DMC}}$ .<sup>4</sup> The lattice is started with a random IC (0 is denoted by white, 1 by dark). Each row in the space-time diagram shows the CA lattice at a particular time step  $t$ , and time increases down the page.

## 6.2 Particle Computation in Evolved Cellular Automata

One thing that is immediately obvious is that there are large, regular, relatively stable regions in the space-time diagram. These regions are called *regular domains*. Examples in Figure 1a are the all-white, all-dark, and checkerboard (alternating white and dark) regions. Crutchfield, Mitchell, and Das [13] refer to the boundaries between these domains as *particles*. They are localized patterns that move through the lattice with a certain constant velocity. Finally, there are collisions between particles, which result in a specific *particle interaction* scheme.

<sup>4</sup> This rule is defined by a 128 bit string as discussed above. The hexadecimal representation of this string, where each hexadecimal digit should be converted to the corresponding 4 bit binary string, is 0504058705000F77037755837BFFB77F.

These dynamical structures—domains, particles, and particle interactions—are dynamic attractors or emergent properties of the CA (see Section 3). It is typically very hard, or even impossible, to derive these emergent structures directly from the 0's and 1's in the CA lookup table. They are only apparent after iteration of the CA rule  $\phi$  and examination of the corresponding dynamical space-time behavior. Domains and particles were defined formally in the *computational mechanics* framework [11, 24]. Computational mechanics further provides a way of suppressing (by way of filtering out) the domains in a space-time diagram, making the particles more explicit. An example of the result of this filtering process for  $\phi_{\text{DMC}}$  is shown in Figure 1b.

Particles behave according to certain rules. For example, they have a certain constant velocity at which they move through the lattice. Velocity is defined as the number of cells the particle moves at each iteration of the CA; it is positive if the movement is to the right of the lattice, and negative if to the left. Particles also interact with one another according to deterministic rules. These rules and the velocities of particles are referred to as a *particle catalog* for a given CA. Typically, such a catalog is based on a small number of particles,  $\alpha$ ,  $\beta$ ,  $\delta$ ,  $\gamma$ ,  $\eta$ , and  $\mu$ , and a small number of rules, such as  $\beta + \gamma \rightarrow \eta$ , meaning that when particles  $\beta$  and  $\gamma$  collide, the particle  $\eta$  results. We can think of these deterministic rules as a production grammar in which the presence (collision) of two particles produces other particles. The catalog of two-particle interactions for rule  $\phi_{\text{DMC}}$  is shown in Table 1.

Particles transfer information about properties of local regions across the lattice to distant sites. Crutchfield, Mitchell, and Das [13] argue that particle collisions are the loci of information processing and result either in the creation of new information (in the form of other particles) or in annihilation.

Hordijk et al. [29] used the computational mechanics framework to collect and catalog particle properties, using them to construct a formal model of the evolved CA's computational capabilities. This “embedded-particle model” abstracts away from the individual cells in the CA space-time configurations and models the behavior of a CA rule at the level of the emergent particles. Indeed, close agreement between the predicted (model) and actual (CA) performance was found, which shows that the particle-level description of CA behavior captures the main mechanisms by which the CA transfers and processes local information to accomplish the computation required by the given task.

We can thus say with confidence that the particle interaction scheme instantiates a rule system used to integrate local information in the lattice and to produce a final global outcome. The rules in the particle interaction catalog are similar to a production grammar for CA particles. What is exciting about these production rules is that they are not specified in the lookup table of the rule, but are

Table 1. Catalog of regular domains, particles, and particle interactions for rule  $\phi_{\text{DMC}}$ .

Regular domains:	
$\Lambda^0 = \{0+\}$ , $\Lambda^1 = \{1+\}$ , $\Lambda^2 = \{(01)+\}$	
Particles (velocities):	
$\alpha \sim \Lambda^0 \Lambda^1 (-)$ , $\beta \sim \Lambda^1 \Lambda^0 (0)$ , $\gamma \sim \Lambda^0 \Lambda^2 (-1)$ ,	
$\mu \sim \Lambda^2 \Lambda^1 (1)$ , $\delta \sim \Lambda^2 \Lambda^0 (-3)$ , $\eta \sim \Lambda^1 \Lambda^2 (3)$	
Observed interactions:	
Decay	$\alpha \rightarrow \gamma + \mu$
Reactions	$\beta + \gamma \rightarrow \eta$ , $\mu + \beta \rightarrow \delta$ , $\eta + \delta \rightarrow \beta$
Annihilations	$\eta + \mu \rightarrow \Lambda^1$ , $\gamma + \delta \rightarrow \Lambda^0$

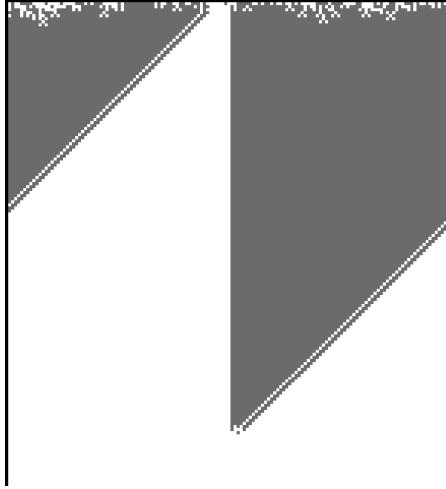


Figure 2. Space-time diagram of block expansion rule  $\phi_{\text{block}}$  when applied to the density task.

rather a product of the space-time dynamics of the CA. Thus, by using an artificial evolutionary algorithm, a CA was evolved to produce a rule-based system that can integrate local information globally to solve the nontrivial density problem, even though its update rule has access only to local information (its neighborhood). Next we discuss the role of particles as information processing entities, and, as suggested by Mitchell [39], as representations.

## 7 Emergent Memory in Evolving Automata

Most CA rules evolved with the evolutionary process described in Section 6 show very simple space-time patterns: They try to solve the problem by block expansion, that is, when large neighborhoods of either 1 or 0 states exist in the initial configuration, they are expanded. Figure 2 shows a space-time diagram of one such rule:  $\phi_{\text{block}}$ .<sup>5</sup> These block expansion rules tackle the density task by a kind of a domino expansion: by taking into account only local information—a 0 or 1 local majority—and propagating that majority directly to neighbors. But, unlike the rules with intricate particle rule systems, such as the rule  $\phi_{\text{DMC}}$ , they lack the ability to globally integrate local information to produce an accurate result. Indeed, the performance of block expansion rules is quite inferior to that of  $\phi_{\text{DMC}}$ , which gives an obvious evolutionary advantage to the latter.

Specifically, the performance (see Section 6) of the two rules for lattices of 149 cells is  $\mathcal{P}_{149,I}(\phi_{\text{DMC}}) = 0.773$  and  $\mathcal{P}_{149,I}(\phi_{\text{block}}) = 0.641$ . Furthermore, when we increase the size of the lattice, the difference is even larger:  $\mathcal{P}_{599,I}(\phi_{\text{DMC}}) = 0.726$ ,  $\mathcal{P}_{999,I}(\phi_{\text{DMC}}) = 0.707$ ; and  $\mathcal{P}_{599,I}(\phi_{\text{block}}) = 0.523$ ,  $\mathcal{P}_{999,I}(\phi_{\text{block}}) = 0.501$ . Whereas the  $\phi_{\text{DMC}}$  rule maintains a similar level of performance for larger lattices, the  $\phi_{\text{block}}$  rule performs very close to random guessing. This fact means that CAs endowed with particle interaction rule systems are indeed capable of effectively integrating information from local areas of large lattices, whereas block-expansion rules are not [12].

### 7.1 Memory and Communication

To understand the system of particle computation in CAs as a simple simulation of material representations, we first need to understand how memory is used by CAs with intricate particle systems.

Ultimately CAs are evaluated for their final dynamic behavior: for the density task, a stable attractor of all cells at state 0 or 1. The objective of the evolved particles is to lead the CA dynamics

<sup>5</sup> This rule in hexadecimal format is 027D3AF97AEF1E6F507AE57F35FEE767.



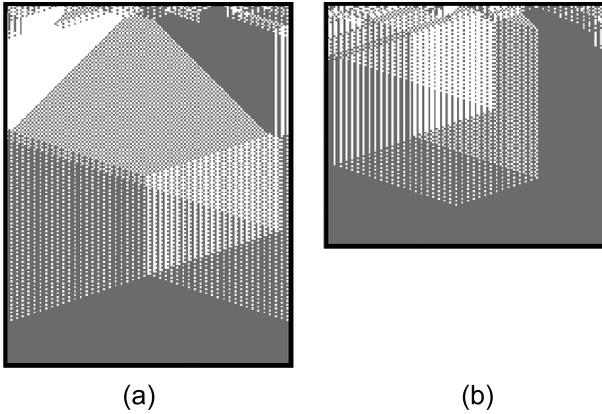


Figure 3.  $\phi_{ABK}$  rule. (a) Four intermediate domains:  $\{(01)^+\}$ ,  $\{(010100)^+\}$ ,  $\{(110010)^+\}$ ,  $\{(11110)^+\}$ . (b) Seven intermediate domains:  $\{(01)^+\}$ ,  $\{(010100)^+\}$ ,  $\{(110010)^+\}$ ,  $\{(11110)^+\}$ ,  $\{(000001)^+\}$ ,  $\{(111010)^+\}$ ,  $\{(001)^+\}$ ,  $\{(110)^+\}$ .

into this very small set of attractor states desirable for the task and selected by the GA. This way, particles are used to produce the (initial) conditions for some target dynamic behavior.

The target final (attractor) behavior is a regular domain in the space-time description. To reach it, the evolved CA employs several intermediate domains in subsets of their lattices (see Section 6). These regular domains, being regions that are “space- and time-translation invariant” [13: p. 17] can be seen as *memory* structures. Each domain is defined by a cyclic repetition of strings (words) from its regular language (the 0’s and 1’s of the CA) in space and time. Unless otherwise perturbed, these domains retain their cycles in space and time. For instance, for the CA rule  $\phi_{DMC}$  (see Figure 1) we observe the three domains specified in Table 1.  $\Lambda^0$  and  $\Lambda^1$  refer to the two desired outcomes for the density task, while  $\Lambda^2$  refers to an intermediate domain used in the process of integrating lattice information and producing the final outcome.<sup>6</sup> Indeed, the introduction of intermediate domains in CAs with intricate particle systems is their key difference from block-expansion rules, which simply propagate the final outcome domains  $\Lambda^0$  and  $\Lambda^1$ . Here we define CAs with intricate particle systems as those CAs that employ at least one intermediate domain.

Domains interact in turn with one another by one taking over the other or by establishing an inalterable border. In either case, their interaction defines the particles described in Section 6. In the first case, we obtain particles (e.g.,  $\mu$  and  $\gamma$  in Figure 1b) that propagate in the direction of the receding domain, at greater or lesser velocity, while in the second case we obtain a particle (e.g.,  $\beta$  in Figure 1b) with zero velocity, which maintains the same lattice position in time, creating a vertical line in the space-time diagram.

The CAs with intricate particle systems use the intermediate domains as *memory* stores for intermediate results, and the particles to *communicate* these results across the lattice. Furthermore, the particle interaction rules are used to *integrate* the information stored in the various intervening domains, ultimately to produce a final homogeneous lattice state. Notice that the content of the memories stored in the domains is at best task-specific.  $\Lambda^2$  in  $\phi_{DMC}$  denotes the presence of a majority of cells with state 1 to one side and 0 to the other in the initial lattice. Its presence in the CA is also used to give equal weight to the  $\Lambda^0$  and  $\Lambda^1$  domains; without it, CAs are forced to favor and expand either  $\Lambda^0$  or  $\Lambda^1$  as block expansion rules do. With the third intermediate memory state,  $\Lambda^0$  and  $\Lambda^1$  do not expand into one another, but only into  $\Lambda^2$ . The inclusion of an additional memory state establishes a more effective means to perform the density task in a distributed manner.

<sup>6</sup> This domain repeats the word 01 in space and the pattern  $01 \rightarrow 10 \rightarrow 01 \rightarrow \dots$  in time.

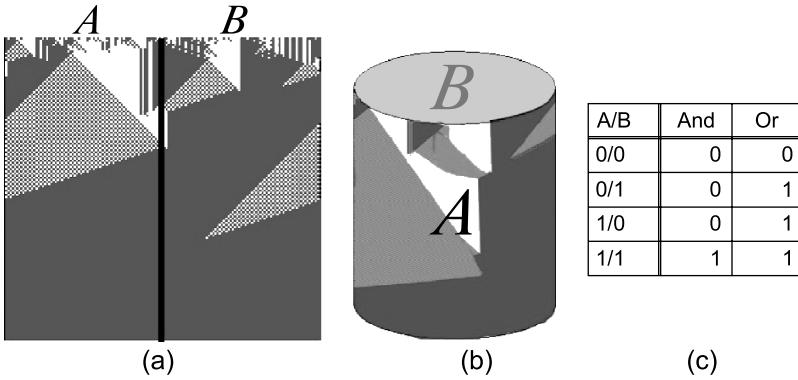


Figure 4. Implementation of logical tasks in a one-dimensional CA. (a) The lattice is divided into halves  $A$  and  $B$ , each interpreted as a separate logical variable whose value is 1 if it contains a majority of cells in state 1, and 0 otherwise. (b) The space-time lattice is periodic; the first cell neighbors the last. (c) Truth table for AND and OR.

But intermediate domains do not necessarily have an obvious meaning regarding the ultimate task. For instance, Andre, Bennet, and Koza [1] evolved a CA rule,  $\phi_{\text{ABK}}$ , for the density task<sup>7</sup> with at least eight intermediate domains besides  $\Lambda^0$  and  $\Lambda^1$  (see Figure 3). These domains seem to be used very effectively to account for different concentrations of 1's and 0's in the initial lattice, creating what seems to be a gradation of intermediate memory states. Indeed, the performance of this rule is higher than for  $\phi_{\text{DMC}}$ :  $\mathcal{P}_{149,I}(\phi_{\text{ABK}}) = 0.826$ ,  $\mathcal{P}_{599,I}(\phi_{\text{ABK}}) = 0.766$ , and  $\mathcal{P}_{999,I}(\phi_{\text{ABK}}) = 0.73$ . The CA rule  $\phi_{\text{ABK}}$  is an example of a very intricate particle interaction system used to produce the expected CA behavior to solve the density task, but where the content of each memory store (domain) is not obvious—unlike  $\phi_{\text{DMC}}$ . This relates to our earlier observation that the content of representations does not need to be meaningful externally.

What we need to investigate now is if domains, particles, and particle interaction rules in these CAs can be seen as a simulation of material representations where memory is used in both active and passive roles. The next subsection details some additional experiments we have conducted to better understand the characteristics of emergent particles and memory in CA.

## 7.2 Building Up Memory: Logical Tasks

The role of domains as emergent memory structures used for distributed information processing via the particle interaction scheme can be further appreciated when we notice that memory can be built upon in order to perform more complicated tasks. Rocha [52, 53] conducted some additional experiments to evolve CAs that perform more than one task. The goal was the evolution of CA rules with radius 3 that can perform both the density task and some related, but more complicated, logical tasks [52, 53]. Here we present a novel analysis of the particle systems evolved for these tasks.

To implement logical tasks the CA lattice is functionally divided in two halves (the center cell is not used):  $A$  and  $B$  (Figure 4a). Each half is interpreted as a separate logical variable in traditional logical operations. A variable is 1 if there is a majority of 1 cells in its lattice half, and 0 otherwise. Notice that since the boundary conditions of the lattice are periodic (see Section 6), this lattice has two boundaries between the two variables (halves)  $A$  and  $B$  (Figure 4b). The cells in the neighborhood of these boundaries compute their values from cells in both halves. However, since we are looking for global integration across the lattice, the local errors at the boundaries are not usually important, especially as lattices grow in size.

The logical tasks AND and OR depend on the density value of the lattice halves  $A$  and  $B$ . Figure 4c depicts the AND and OR truth tables for the four possible density states of lattice halves

<sup>7</sup> With genetic programming rather than genetic algorithms. The rule is defined by the following hexadecimal string: 050055050500550555FF55FF55FF55FF.

$A$  and  $B$ . Both logical tasks are related to the density task, because when the density of both halves is below (above) 0.5, both logical variables are 0 (1), leading to a desired final lattice with all cells 0 (1). They differ from each other and from the density task for the cases when the two halves of the lattice have opposing densities. The gist of these logical tasks is that they should ideally perform the density task in each half, and then integrate the results appropriately. Several rules were evolved with a GA whose initial population of 100 individuals was composed of 20 individuals encoding some of the best rules evolved so far for the density task, including  $\phi_{\text{DMC}}$  and  $\phi_{\text{ABK}}$  as well as rules evolved by Juillé and Pollack [32] and others [53], and 80 randomly generated individuals. The fitness function used in this GA was calculated by presenting each rule with 100 different ICs: 50 to be analyzed by the density task, and the other 50 by either the AND or the OR task, depending on the run. The 50 ICs presented to the density task had their density of 1's uniformly distributed over the unit interval (just as in the experiments described in Section 6). The 50 ICs presented to the AND (OR) task were biased to a uniform distribution of lattices where for 50% of lattices the density of at least one of the halves  $A$  or  $B$  was 0 (1), and for the other 50% the density of both halves was 1 (0). If we were to use an unbiased distribution of lattices, only 25% of the time would the case of both halves having density 1 (0) be generated, thus making rules that always tend to 0 (1) too favorable in the evolutionary process.

From these experiments, several CA rules were evolved that can simultaneously perform the density task and one of the logical tasks very well (details in [52, 53]). The significance of having rules that can perform more than one task was discussed in [53]. What we want to highlight here is the manner in which evolved CA particle systems dealt with the different requirements for information integration across the lattice demanded by the logical tasks. Because the logical tasks divide the lattice into two halves, we expected evolved CA rules to create additional domains and particles, which would behave more like static, local memory stores, whose information could be accessed at a later time as needed.

Indeed, this is what we observed in the CA rule that best performed the AND task,  $\phi_{\text{AND}}$ .<sup>8</sup>  $\mathcal{P}_{149,I}(\phi_{\text{AND}}) = 0.839$ . The strategy of this rule builds on the rule  $\phi_{\text{DMC}}$  by creating an additional intermediate domain, which keeps local lattice information without expanding. The domain and particle catalog of  $\phi_{\text{AND}}$  is detailed in Table 2. Figures 5, 6, and 7 show space-time diagrams for this rule, with particle interaction schematics.

The most striking feature of the particle catalog of the rule  $\phi_{\text{AND}}$  is the existence of several particles with zero velocity. These are particles that remain in the same position in the lattice until other particles collide with them. Whereas the rule  $\phi_{\text{DMC}}$  had only one particle with zero velocity ( $\beta$ ), the rule  $\phi_{\text{AND}}$  produces three such particles ( $\beta$ ,  $\beta'$ , and  $\beta''$ ). We named all these particles  $\beta$  to highlight the similarity of their behavior to that of the particle  $\beta$  of  $\phi_{\text{DMC}}$ .

The particles  $\beta'$  and  $\beta''$  are both due to the fourth domain  $\Lambda^3$  introduced by the rule  $\phi_{\text{AND}}$ . This domain does not expand into the final domains  $\Lambda^0$  and  $\Lambda^1$ , so its borders with these domains, the particles, have velocity zero. It only expands into the intermediate domain  $\Lambda^2$  with particle  $\delta^9$ . The domain  $\Lambda^3$  functions as a static intermediate memory store. In the rule  $\phi_{\text{DMC}}$ , without  $\Lambda^3$ , when the particles involving domain  $\Lambda^2$  collide with others, the result is always one of the final domains  $\Lambda^0$  or  $\Lambda^1$ , while in  $\phi_{\text{AND}}$  some collisions result in the additional intermediate domain  $\Lambda^3$ . This way, the domain  $\Lambda^3$ , confined by the static particle  $\beta'$  or  $\beta''$ , preserves an intermediate result without spreading it into neighbor domains. The intermediate result can later be integrated with particles from other lattice regions: A collision with particle  $\nu$  results in the all-0 domain  $\Lambda^0$ , and a collision with particle  $\epsilon$  results in the all-1 domain  $\Lambda^1$ .

The existence of the fourth domain and its static particles is particularly useful for the logical tasks (AND in this case). Because two arbitrary halves are defined, the task encourages the evolution of rules that can hold intermediate results in one part of the lattice to be integrated with those from another part. Indeed, the logical task can be better executed when a *static type of memory* is produced to

<sup>8</sup> This rule is defined by the hexadecimal string 005F1053405F045F005FFD5F005DFF5F.

<sup>9</sup> We note that  $\Lambda^3$  typically exists as  $\{(110)^+\}$  but can also exist as  $\{(001)^+\}$ . We consider these patterns to be the same domain because they behave in exactly the same manner in particle interactions, and are in effect interchangeable.

Table 2. Catalog of regular domains, particles, and particle interactions for rule  $\phi_{\text{AND}}$ .

---

Regular domains:

$$\Lambda^0 = \{0+\}, \Lambda^1 = \{1+\}, \Lambda^2 = \{(01)+\}, \Lambda^3 = \{(110)+\} \vee \{(001)+\}$$

Particles (velocities):\*

$$\alpha \sim \Lambda^1 \Lambda^0 (-),$$

$$\beta \sim \Lambda^0 \Lambda^1 (0), \beta' \sim \Lambda^0 \Lambda^3 (0), \beta'' \sim \Lambda^3 \Lambda^1 (0),$$

$$\gamma \sim \Lambda^1 \Lambda^2 (-1), \delta \sim \Lambda^2 \Lambda^3 (-3), \epsilon \sim \Lambda^1 \Lambda^3 (3), \eta \sim \Lambda^0 \Lambda^2 (3),$$

$$\mu \sim \Lambda^2 \Lambda^0 (1), \nu \sim \Lambda^3 \Lambda^0 (-3)$$

Observed interactions:

Decay  $\alpha \rightarrow \gamma + \mu$

Reactions  $\beta + \gamma \rightarrow \eta, \beta'' + \gamma \rightarrow \nu + \eta,$

$$\mu + \beta \rightarrow \delta + \beta'', \mu + \beta' \rightarrow \delta, \eta + \delta \rightarrow \beta', \gamma + \delta \rightarrow \epsilon,$$

$$\epsilon + \nu \rightarrow \gamma + \mu$$

Annihilations  $\beta' + \nu \rightarrow \Lambda^0, \eta + \mu \rightarrow \Lambda^0, \epsilon + \beta'' \rightarrow \Lambda^1$

---

\* The domain combinations  $\Lambda^2 \Lambda^1$  and  $\Lambda^3 \Lambda^2$  were not observed as stable boundaries or particles.

hold intermediate results, which in this case is implemented by the domain  $\Lambda^3$  and its static particles  $\beta'$  and  $\beta''$ .

## 8 Are There Representations in Evolving Automata?

In Section 5 we argued that material representations possess two main roles: the active role referring to the construction of dynamic configurations from initial conditions encoded in representations, and the passive role referring to the ability to manipulate and communicate representations without recourse to content. We also stressed that the passive role is possible only because representations are implemented as dynamically incoherent memory. This leads to a view of representations that are not dynamic stand-ins, but are rather constraints (initial conditions) on dynamics, which can evolve with the dynamics, but are kept as a separate type of material structure: memory.

To decide if the process of emergent computation in CA described in Sections 6 and 7 can be proposed as a simulation of material representations, we now discuss if it meets the three requirements for material representations proposed in Section 5. We proceed in reverse order.

### 8.1 Requirement 3: Self-Organization and Selection

This is the easiest requirement to meet in the evolving CA experiments, because the GA models the process of natural selection, and the CA models a process of self-organization. Clearly, both tools are very incomplete models of the natural phenomena, but they do capture the essential characteristics we

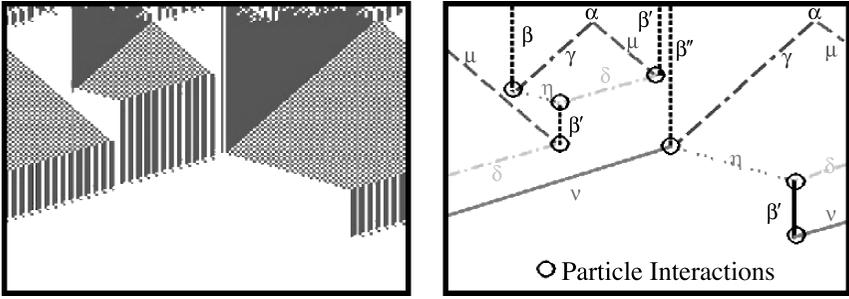


Figure 5. Space-time diagram and particle interactions for  $\phi_{AND}$  given a random IC, leading to an all-0 lattice.

need to model here. Self-organization is modeled by the parallel, state-determined transitions of cellular automata. The GA models a feedback mechanism that increases the reproduction rates of those CA rules that best solve a task. If the evolved particle catalog is a representation system, then those systems that best solve the task are selected. Indeed, if one tracks the artificial evolutionary process (e.g., as Das, Mitchell, and Crutchfield [15] do), one sees the step-by-step evolution of a better and better particle interaction scheme, as the behavior of particles changes to accommodate the task.

### 8.2 Requirement 2: Construction Code

We can think of the set of particle interaction rules that emerge in the evolving CA experiments as a process that maps from the random initial state of the CA lattice (IC) into a final desired state for the task. Crutchfield, Mitchell, and Das [13] regard this process as a computation that produces a final outcome from the IC input. As we detail below, we do not see this process as a computation, but it can be seen as the construction of a desired dynamics. Whether or not we can conceive of the individual particles as representations, they are certainly the elements in the space-time behavior of the CAs that communicate information across the lattice: the loci of information processing [13]. Therefore, the collection of particle interactions in space-time is a process of integration of the information carried by each of the individual particles into a final desired dynamical outcome. We can thus say that, from the individual particles generated initially given the IC, the particle interaction rules construct with their collisions the (initial) conditions for a target attractor behavior. Therefore, the set of particle interaction rules is a *construction code* for obtaining a desired dynamical behavior from a set of particles—a code that mediates between information-carrying entities (the particles) and dynamical behavior (a final attractor state).

Notice also that the particular particle interaction scheme is malleable to evolution. There is in this sense an *arbitrary* relationship between particles and the results of a specific particle interaction scheme. We can appreciate this by contrasting the particle interaction schemes of the rules  $\phi_{DMC}$  and  $\phi_{AND}$ . The evolutionary process tweaked the first to endow the second with a few more parti-

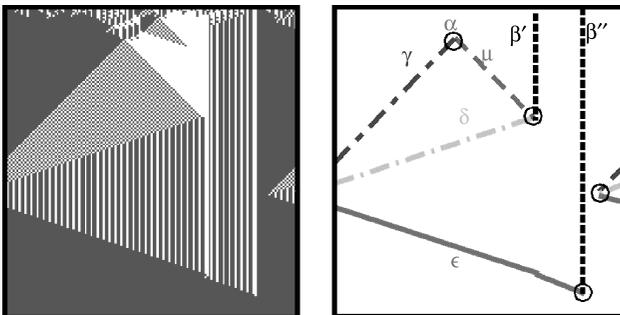


Figure 6. Space-time diagram and particle interactions for  $\phi_{AND}$  given a random IC, leading to an all-1 lattice.

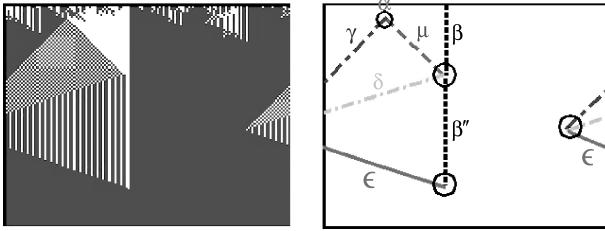


Figure 7. Space-time diagram and respective particle interactions for  $\phi_{\text{AND}}$  given random IC, leading to an all-1 lattice.

cles and interaction behavior. There is a very large set of possible CA rules leading to different particles and particle interaction schemes, which justifies our assertion that, evolutionarily, the mapping between particles and final dynamic outcomes instantiated by a particle interaction scheme is an arbitrary code.

### 8.3 Requirement 1: Dynamically Incoherent Memory

Above we considered the evolved CA particles as the candidates for representations. For them to be seen as representations as defined in Section 5, they would need to be dynamically incoherent with the rest of the CA dynamics. Clearly, particles are used for communication across the CA lattice, but the manner in which they are used is not truly dynamically incoherent—they are not inert in the sense described in Section 5. Also, the particles cannot be used without direct access to their content—there is not a syntax for symbol manipulation. Let us elaborate on this.

Dynamically incoherent symbols require random access memory, that is, memory that can be accessed at any time, and whose value is independent of dynamics. This is the same as saying that the value in the memory is the same independent of the rate of access to it. When a computer stores the value of a variable in a memory store (e.g., the tape in a universal Turing machine), that value remains unchanged when accessed, and the speed of the computer in accessing the memory and performing computations also does not change it. Similarly, a computation is a process of integrating memory in store with algebraic and logical operations. But the speed of the computer does not change the result of the computation:  $2+2=4$  in any computer.

Clearly this does not happen with the particles of the evolved CA. Particles have a velocity, and the result of all particle interactions in space-time depends on when the particles meet each other, which depends on the velocity of the particles. If the particles start from different locations in the lattice, even preserving lattice density, they may collide differently and produce a different outcome for the tasks we studied here. It is as if it were the case that  $2+2=4$  only when 2 and 2 meet at the right time. This is why we do not see the process of particle interactions as a computation, but rather as a construction code.

It was because of this issue that we created the logical tasks. In this case, the evolved CA came as close as possible to creating dynamically incoherent memory stores. Indeed, the fourth domain  $\Lambda^3$  created by the rule  $\phi_{\text{AND}}$ , as discussed in Section 7, preserves its memory without spreading it into the final outcome domains  $\Lambda^0$  and  $\Lambda^1$ . In a sense, it keeps its memory until it is accessed. The several particles  $\beta$  created by this rule have zero velocity; therefore they preserve the same information until a particle of nonzero velocity collides with them. In that respect, they function more like traditional memory stores.

However, the information they store is still not separated from the dynamics. They are not inert in the sense that DNA is. It is by virtue of their dynamics, the way they collide, that information is expressed. In contrast, information is read out of DNA by “third-party” machinery, without destroying or reacting with it. So while the  $\beta$  particles of the evolved CA were able to create static memory stores, these are still reactive with the embedding dynamics. Therefore, the information they carry is expressed in dynamic reactions, which destroy them, and not via a separate encoding.

This point is obvious when we notice that while processes such as the transcription of mRNA from DNA and RNA editing work on genetic memory without access to its content (the encoded

proteins), our evolved CA cannot manipulate their particles without access to their content. Particle reactions are simply domain interactions. In this sense, information carriers (would-be representations) and content are inseparable. This way, we can say that while the evolved CA particles observe the active role of representations as they construct specific dynamics from the interaction rules (a construction code), they do not observe the passive role of representations as defined in Section 5. They do not function as inert memory stores, which can be manipulated without access to content. As a corollary, if particles are not symbolic representations, their interaction to perform a task is not a computational process in any traditional sense [49].

## 9 Where To?

In this article we have presented a definition of representation based on what we know about genetic memory. Our key observations are that representations are mechanisms used to control dynamics more effectively by encoding information to construct alternative dynamic configurations. They are not stand-ins in dynamic processes, and do not need to refer to situations external to a given dynamics (Sections 4 and 5). We also propose criteria to decide if a given structure is a material representation:

1. *Dynamically incoherent memory*: Information in representations can be used without access to content and is implemented in non-reactive structures.
2. *Construction code*: Information in representations is used to construct dynamic configurations; representations encode alternative initial conditions for a dynamical system-environment coupling.
3. *Self-organization and selection*: Representations are material structures existing under the constraints of self-organization and are selected in an evolutionary process.

Additionally, we discussed experiments with CA leading to the evolution of rules that can perform nontrivial tasks, and presented new results for this problem. We discussed if the evolved CA particles may be seen as representations according to our criteria. While these particles capture the active role of representations (construction code), they do not capture the passive role (dynamically incoherent memory). Does this mean that we cannot study the emergence of material representations computationally?

Our stumbling block was in obtaining the necessary separation between representations and content, that is, of obtaining a means to manipulate representations without recourse to content (syntax). This has been a recurrent difficulty in artificial life. For instance, Langton [34] proposed a self-reproduction scheme in CAs in which the separation between genotype (information) and phenotype (content) was blurred. This lack of separation was actually seen as a worthwhile model for studying artificial life, with a generalized concept of genotype-phenotype mappings [35]. But, as was clear for some theoretical biologists looking at artificial life, a strict separation between genotype and phenotype is the key feature of life as we know it [44] and a necessary condition for open-ended evolution [63, 45, 54]. Thus, the study of the emergence of a strict separation between genotype and phenotype, between representations and content, from a purely dynamic milieu should still be the primary goal of artificial life.

The evolving CA experiments here described are an ideal testbed for studying the emergence of information processing from self-organizing systems. The experimental model contains all the necessary ingredients: self-organization, selection, and the requirement of global information integration from local sources. As demonstrated in this article, with the evolution of the rule  $\Phi_{\text{AND}}$  for performing the AND task we come very close to producing proper static memory stores. But the dynamics produced by one-dimensional CAs may be too simple to achieve what we desire to model.

Indeed, the homogeneous CA as a model of material dynamics—our artificial chemistry—is rather poor. In biology, the genotype-phenotype mapping is based on the existence of two basic, distinct types of material (chemical) structures: DNA/RNA and amino acid chains. The two are quite different: DNA is remarkably unreactive (biochemically inert), RNA can be both reactive and

unreactive whereas amino acid chains are incredibly rich biochemical machines. In contrast, our one-dimensional homogeneous CAs compute the same exact update rule in each cell.

It seems reasonable that in order to evolve a system in which more reactive structures use non-reactive structures as information stores, we need to work with more heterogeneous dynamical systems where different populations of artificial chemistry structures interact. Perhaps heterogeneous CAs will suffice, or we may need more intricate models of artificial chemistry such as that of Fontana [19]. But we should still retain the complete hands-off approach of the evolving CA experiments. We should not preassign which types of structures are information carrying and reactive ones; such functional assignment should emerge from the experiments. The preassignment is very reasonable in computational studies of the origin of the genetic code (e.g., [66]), which we know exists, but not when we wish to study the evolution of representations from a dynamical milieu at large—a very worthwhile goal for artificial life.

## References

1. Andre, D., Bennett, F. H., III, & Koza, J. R. (1996). Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem. In J. R. Koza, D. E. Goldberg, D. B. Fogel, & R. L. Riolo (Eds.), *Genetic programming 1996* (pp. 3–11). Cambridge, MA: MIT Press.
2. Bass, B. (2001). *RNA editing*. Oxford, UK: Oxford University Press.
3. Bedian, V. (1982). The possible role of assignment catalysts in the origin of the genetic code. *Origins of Life*, *12*(2), 181–204.
4. Bedian, V. (2001). Self-description and the origin of the genetic code. *Biosystems*, *60*(1–3), 39–47.
5. Beer, R. D. (1995). A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, *72*(1–2), 173–215.
6. Bickhard, M. H. (2002). The biological emergence of representation. In T. Brown, & L. Smith (Eds.), *Emergence and reduction: Proceedings of the Twenty-Ninth Annual Symposium of the Jean Piaget Society* (pp. 105–131). Mahwah, NJ: Erlbaum.
7. Cassirer, E. (1957). *Language and myth*. Mineola, NY: Dover.
8. Clark, A. (1996). Happy couplings: Emergence and explanatory interlock. In M. Boden (Ed.), *The Philosophy of Artificial Life* (pp. 262–281). Oxford, UK: Oxford University Press.
9. Conrad, M. (1993). Adaptability theory as a guide for interfacing computers and human society. *Systems Research*, *10*(4), 3–24.
10. Crick, F. (1968). The origin of the genetic code. *Journal of Molecular Biology*, *38*, 367–379.
11. Crutchfield, J. P., & Hanson, J. E. (1993). Turbulent pattern bases for cellular automata. *Physica D*, *69*, 279–301.
12. Crutchfield, J. P., & Mitchell, M. (1995). The evolution of emergent computation. *Proceedings of the National Academy of Sciences of the U.S.A.*, *92*, 10,742–10,746.
13. Crutchfield, J. P., Mitchell, M., & Das, R. (2002). The evolutionary design of collective computation in cellular automata. In J. P. Crutchfield, & P. K. Schuster (Eds.), *Evolutionary dynamics: Exploring the interplay of selection, neutrality, accident, and function* (pp. 361–412). Oxford, UK: Oxford University Press.
14. Das, R., Crutchfield, J. P., Mitchell, M., & Hanson, J. E. (1995). Evolving globally synchronized cellular automata. In L. J. Eshelman (Ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms* (pp. 336–343). San Mateo, CA: Morgan Kaufmann.
15. Das, R., Mitchell, M., & Crutchfield, J. P. (1994). A genetic algorithm discovers particle-based computation in cellular automata. In Y. Davidor, H.-P. Schwefel, & R. Manner (Eds.), *Parallel problem solving from nature—PPSN III: International Conference on Evolutionary Computation: The third conference on parallel problem solving from nature, 9–14 Oct. 1994, Jerusalem, Israel* (pp. 344–353). Berlin: Springer-Verlag.
16. Dretske, F. (1995). *Naturalizing the mind*. Cambridge, MA: MIT Press.
17. Eigen, M. (1992). *Steps towards life: A perspective on evolution*. Oxford, UK: Oxford University Press.
18. Emmeche, C., & Hoffmeyer, J. (1991). From language to nature—The semiotic metaphor in biology. *Semiotica*, *84*(1/2), 1–42.



19. Fontana, W. (1991). Algorithmic chemistry. In C. G. Langton, C. Taylor, J. D. Farmer, & S. Rasmussen (Eds.), *Artificial Life II* (pp. 159–209). Reading, MA: Addison-Wesley.
20. Freeman, W. J., & Skarda, C. A. (1990). Chaotic dynamics versus representationalism. *Behavioral and Brain Sciences*, 13(1), 167–1990.
21. Garrett Millikan, G. (1984). *Language, thought, and other biological categories*. Cambridge, MA: MIT Press.
22. Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.
23. Goodwin, B. (1994). *How the leopard changed its spots: The evolution of complexity*. New York: Charles Scribner's Sons.
24. Hanson, J. E., & Crutchfield, J. P. (1992). The attractor-basin portrait of a cellular automaton. [Journal of Statistical Physics](#), 66(5/6), 1415–1462.
25. Harnad, S. (1990). The symbol grounding problem. [Physica D](#), 42(1–3), 335–346.
26. Haugeland, J. (1991). Representational genera. In W. M. Ramsey, S. P. Stich, & D. E. Rumelhart (Eds.), *Philosophy and Connectionist Theory* (pp. 61–89). Mahwah, NJ: Erlbaum.
27. Hoffman, G. (1975). The stochastic theory of the origin of the genetic code. In H. Eyring, C. J. Christianson, & H. S. Johnson (Eds.), *Annual Reviews of Physical Chemistry*, Vol. 26. San Diego, CA: Academic Press.
28. Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.
29. Hordijk, W., Crutchfield, J. P., & Mitchell, M. (1998). Mechanisms of emergent computation in cellular automata. In A. E. Eiben, T. H. Back, M. Schoenauer, & H.-P. Schwefel (Eds.), *Parallel problem solving from nature* (pp. 613–622). New York: Springer-Verlag.
30. Huang, C. F., & Rocha, L. M. (2003). Exploration of RNA editing and design of robust genetic algorithms. In R. Sarker (Ed.), *Proceedings of the 2003 IEEE Congress on Evolutionary Computation. Canberra, Australia, December 2003* (pp. 2799–2806). Piscataway, NJ: IEEE Press.
31. Jackendoff, R. (2003). *Foundations of language: Brain, meaning, grammar, evolution*. Oxford, UK: Oxford University Press.
32. Juillé, H., & Pollack, J. B. (1998). Coevolving the “ideal” trainer: Application to the discovery of cellular automata rules. In J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, & R. L. Riolo (Eds.), *Genetic Programming Conference (GP-98), 22–25 July 1998, Madison, WI, USA*. San Mateo, CA: Morgan Kaufmann.
33. Kauffman, S. (1993). *The origins of order: Self-organization and selection in evolution*. Oxford, UK: Oxford University Press.
34. Langton, C. G. (1986). Studying artificial life with cellular automata. [Physica D](#), 22(1–3), 120–149.
35. Langton, C. G. (1989). Artificial life. In C. Langton (Ed.), *Artificial life* (pp. 1–47). Reading, MA: Addison-Wesley.
36. McClelland, J. L., Rumelhart, D. E., & Hinton, G. E. (1986). The appeal of parallel distributed processing. In D. E. Rumelhart, & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1: Foundations, Vol. 1* (pp. 3–44). Cambridge, MA: MIT Press.
37. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115–133.
38. Mitchell, M. (1996). *An introduction to genetic algorithms*. Cambridge, MA: MIT Press.
39. Mitchell, M. (1998). A complex-systems perspective on the “computation vs. dynamics” debate in cognitive science. In M. A. Gernsbacher, & S. J. Derry (Eds.), *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 710–715).
40. Moreno, A., Etxeberria, A., & Umerez, J. (1994). Universality without matter? In R. Brooks, & P. Maes (Eds.), *Artificial Life IV: Fourth International Workshop on the Syntheses and Simulation of Living Systems* (pp. 406–410). Cambridge, MA: MIT Press.
41. Newell, A., & Simon, H. A. (1976). Computer science as empirical inquiry—Symbols and search. [Communications of the ACM](#), 19(3), 113–126.

42. Nieselt-Struwe, K., & Wills, P. R. (1997). The emergence of genetic coding in physical systems. *Journal of Theoretical Biology*, 187(1), 1–14.
43. Pattee, H. H. (1982). Cell psychology: An evolutionary approach to the symbol-matter problem. *Cognition and Brain Theory*, 5(4), 191–200.
44. Pattee, H. H. (1995). Artificial life needs a real epistemology. In F. Moran, A. Moreno, J. J. Merelo, & P. Merelo (Eds.), *Advances in Artificial Life* (pp. 23–38). Berlin: Springer-Verlag.
45. Pattee, H. H. (1995). Evolving self-reference: Matter, symbols, and semantic closure. *Communication and cognition—Artificial intelligence*, 12(1–2), 9–27.
46. Pattee, H. H. (2001). The physics of symbols: Bridging the epistemic cut. *Biosystems*, 60(1–3), 5–21.
47. Pollack, R. (1994). *Signs of life: The language and meanings of DNA*. Boston: Houghton Mifflin.
48. Ramsey, W. M., Stich, S. P., & Rumelhart, D. E. (1991). *Philosophy and connectionist theory*. Mahwah, NJ: Erlbaum.
49. Rocha, L. M. (1995). Contextual genetic algorithms: Evolving developmental rules. *Lecture Notes in Artificial Intelligence*, 929, 368–382.
50. Rocha, L. M. (1996). Eigenbehavior and symbols. *Systems Research*, 13(3), 371–384.
51. Rocha, L. M. (1998). Selected self-organization and the semiotics of evolutionary systems. In S. Salthe, G. vandeVijver, & M. Delpo (Eds.), *Evolutionary systems: Biological and epistemological perspectives on selection and self-organization* (pp. 341–358). Dordrecht, The Netherlands: Kluwer Academic.
52. Rocha, L.M. (1998). Syntactic autonomy. In *Joint Conference on the Science and Technology of Intelligent Systems ISIC/CIRA/ISAS; September 14–17, 1998; NIST; Gaithersburg, MD* (pp. 706–711). Piscataway, NJ: IEEE Press.
53. Rocha, L. M. (2000). Syntactic autonomy: Why there is no autonomy without symbols and how self-organizing systems might evolve them. *Annals of the New York Academy of Sciences*, 901, 207–223.
54. Rocha, L. M. (2001). Evolution with material symbol systems. *Biosystems*, 60(1–3), 95–121.
55. Rocha, L. M., & Joslyn, C. (1998). Simulations of embodied evolving semiosis: Emergent semantics in artificial environments. *Simulation Series*, 30(2), 233–238.
56. Rosen, R. (1993). Bionics revisited. In H. K. A. Haken, & U. Svedin (Eds.), *The Machine as a Metaphor and Tool* (pp. 87–100). New York: Springer-Verlag.
57. Thelen, E., Schoner, G., Scheier, C., & Smith, L. B. (2000). The dynamics of embodiment: A field theory of infant perseverative reaching. *Behavioral and Brain Sciences*, 24(1).
58. Thompson, E. (1997). Symbol grounding: A bridge from artificial life to artificial intelligence. *Brain and Cognition*, 34(1), 48–71.
59. Umerez, J. (1995). Semantic closure: A guiding notion to ground artificial life. *Lecture Notes in Artificial Intelligence*, 929, 77–94.
60. Van Gelder, T. (1998). The dynamical hypothesis in cognitive science. *Behavioral and Brain Sciences*, 21, 615–665.
61. Van Gelder, T., & Port, R. (1995). It's about time: An overview of the dynamical approach to cognition. In R. Port & T. Van Gelder (Eds.), *Mind as motion: Explorations in the dynamics of cognition* (pp. 1–43). Cambridge, MA: MIT Press.
62. Varela, F., Thompson, F. E., & Rosch, E. (1991). *The embodied mind: Cognitive science and human experience*. Cambridge, MA: MIT Press.
63. von Neumann, J. (1966). *The theory of self-reproducing automata*. University of Illinois Press.
64. Wheeler, M., & Clark, A. (1999). Genic representation: Reconciling content and causal complexity. *British Journal for the Philosophy of Science*, 50(1), 103–135.
65. Whitehead, A. N. (1927). *Symbolism: Its meaning and effect*. New York: Macmillan.
66. Wills, P. R. (2001). Autocatalysis; Information and coding. *Biosystems*, 60(1–3), 49–57.
67. Wuensche, A., & Lesser, M. (1992). *The global dynamics of cellular automata: An atlas of basin of attraction fields of one-dimensional cellular automata*. Reading, MA: Addison-Wesley.