LA-UR-98-757

CONF-980669--

RECEIVED
JUL 31 1998
OSTI

TITLE: **MATTER AND SYMBOLS OF THE ARTIFICIAL**

AUTHOR(S): Luis M. Rocha

SUBMITTED TO: External Distribution - Hard Copy

Los Alamos

Los Alamos National Laboratory
Los Alamos New Mexico 87545

DISCLAIMER

Matter and Symbols of the Artificial

Luis Mateus Rocha
Los Alamos National Laboratory, MS P990

e-mail: rocha@lanl.gov or rocha@santafe.edu
URL: http://www.c3.lanl.gov/~rocha

Abstract.
The study of complex systems should be based on a systems-theoretic framework which requires both self-organizing and symbolic dimensions. An inclusive framework based the notion of semiotics is advanced to build models capable of representing, as well as evolving in their environments, with implications for Artificial Life. Such undertaking is pursued by discussing the ways in which symbol and matter are irreducibly intertwined in evolutionary systems. The problem is thus phrased in terms of the semiotic categories of syntax, semantics, and pragmatics. With this semiotic view of matter and symbols the requirements of semiotic closure are expressed in models with both self-organizing and symbolic characteristics. Situated action and recent developments in the evolution of cellular automata rules to solve non-trivial tasks are discussed in this context. Finally, indirect encoding schemes for genetic algorithms are developed which follow the semiotic framework here proposed.

Keywords: Evolutionary Systems, Semiotics, Emergence, Embodiment, Self-Organization, Situated Action, Cellular Automata, Genetic Algorithms, Fuzzy Sets.

1. SEMANTIC EMERGENCE AND SELECTED SELF-ORGANIZATION

1.1 Self-Organization and Matter

Self-organization is seen as the process by which systems of many components tend to reach a particular state, a set of cycling states, or small volumes of their state space (attractor basins), with no external interference. Therefore, the system's order cannot be imposed by special initial conditions, which would amount to a special creation. Thus, self-organizing behavior is the spontaneous formation of well organized structures, patterns, or behaviors, from random initial conditions. The systems used to study this behavior computationally are referred to as dynamical systems or state-determined systems. They possess a large number of elements or variables, and thus high-dimensional state spaces. Examples of computational dynamical systems are boolean networks and cellular automata. Computational self-organization is often used to study physical matter. The state-determined transition rules are interpreted as the laws of some physical or chemical system. It follows from this that there is a propensity for matter to self-organize (e.g. the work of Prigogine [1985] and Kauffmann [1993]). In this sense, matter is described by the laws of physics and the emergent characteristics of self-organization. In the following, whenever the words matter and materiality are used, they should be understood this way.

1.2 Emergent Classification and Semantic Emergence

Self-organizing attractor values 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). Self-organizing systems can only classify those aspects of their interaction with and environment which result in the maintenance of some internally stable state or attractor. The process of obtaining novel classifications of an environment by a self-organizing system, which can only be achieved by structural changes to its attractor landscape (e.g. weight changes in a neural network), can be referred to generally as *emergent classification* [for more details on this argument please refer to Rocha, 1996, 1997] Emergent because it is the result of the local interaction of the basic components of the self-organizing system and not from a global controller. In the following, I will refer to systems that are capable of emergent classification as *classifying systems*. I will also restrict the term *agent* to such systems.

There are three levels that need to be addressed when dealing with the notion of emergent phenomena in self-organizing systems, in particular, of emergent classification. First, there is the material, dynamical, substrate, which will be the causal basis for all other levels that we may further distinguish. Second, we have the attractor behavior of this dynamics. Finally, we have the (possible) utilization of the set of attractors as referents for some aspects of the interaction of the dynamical system itself with its environment (e.g. the pattern recognition abilities of neural networks). No physical or formal description of the dynamical system and its attractors alone will completely explain the "standing-for", or semantic, dimension [Pattee, 1995a]. The emergence of the level of attractor behavior from the level of dynamics, and of the level classification from the level of attractor behavior is based on *explanatory emergence* [Rocha, 1996; Clark, 1996] defined as the existence of complementary modes of description. However, the emergence of classification from attractor behavior introduces a more specific form of semantic emergence as it establishes a representational relation between the classifying system and its environment.

1.3 Selected Self-Organization

For a dynamic system to observe genuine emergence of new classifications, that is, to be able to accumulate useful variations, it must change its structure. One way or another, this structural change leading to efficient classification (not just random change), has only been achieved through some external influence on the self-organizing system. Artificial neural networks discriminate by changing the structure of their connections through an external learning procedure. Evolutionary strategies rely on internal random variation which must ultimately be externally selected. In other words, the self-organizing system must be structurally coupled [Maturana and Varela, 1987] to some external system which acts on structural changes of the first and induces some form of explicit or implicit selection of its dynamic representations: *selected self-organization* [Rocha, 1996, 1997, 1998a].

Now, for a self-organizing system to observe an evolving or adapting emergent classification of its own interaction with an environment, it must be able to change its structure, and subsequently its attractor basins, explicitly or implicitly. For selection to occur we must have some internal vehicle for classification — there must be different alternatives. The attractor landscape offers these alternatives. One way of conceptualizing this, is to think of the attractor landscape as a distributed [Van Gelder, 1992] memory bank. Each attractor basin storing a given classification function or morphology. Whatever the form of selection, this selected self-organization must be able to classify its interaction with an environment by utilizing its own memory. Hence, it explicitly emphasizes a second dimension of a material (embodied) semiosis of situated self-organizing systems with their environments. If classification implies semantic emergence, selection implies pragmatic environmental influence. In fact, these two dimensions of semiosis cannot be separated, the meaning of the classifications of a self-organizing system does not make sense until it is grounded on the repercussions it triggers in its environment. The structural coupling, or situation, of an agent in its environment is the source of meaning. In this sense, meaning is not private to the agent but can only be understood in the context of the agent's situation in an environment. More on situated semiosis ahead.

1.4 Von Neumann and the Symbolic Advantage

Von Neumann's [1966] model of self-replication is a systems-theoretic criteria for open-ended evolution [for a detailed discussion of this model see Rocha, 1996, 1998a]. Based on the notion of universal construction and description it provides a threshold of complexity after which systems that observe it can for ever more increase in complexity (open-ended evolution). This model clearly does not rely on a distributed but on a local kind of memory. Descriptions entail a symbol system on which construction commands are cast. These commands are not distributed over patterns of activation of the components of a dynamic system, but instead localized on "inert" structures which can be used at any time — a sort of random access memory. By "inert" I mean material structures with many dynamically equivalent states, in other words, the semantic relation, or what the structures are used to refer to, must possess a large degree of arbitrariness so that certain representations are not much more probable than others. For instance, in the genetic system (which Von Neumann's model conceptually describes), most any sequence of nucleotides is possible, and its informational value is almost completely independent of the particular dynamic behavior of DNA or RNA.

Von Neumann showed that there is an advantage of local memory over purely dynamic because if we do not have symbolic descriptions directing self-replication, then an organism must replicate through material self-inspection of its parts. Clearly, as systems grow in complexity, self-inspection becomes more and more difficult [Pattee, 1995a]. The existence of a language, a symbol system, allows a much more sophisticated form of communication. Functional, dynamic structures do not need to replicate themselves, they are simply constructed from physically non-functional (dynamically inert) descriptions. For instance, for an enzyme to replicate itself, it would need to have this intrinsic property of self-replication "by default", or it would have to be able to assemble itself from a pool of existing parts. But for this, it would have to "unfold" so that its internal portions could be reconstituted for the copy to be produced [Pattee, 1995a]. With the genetic code, however, none of these complicated gimmicks are necessary: functional molecules can be simply folded from inert messages. This method is by far more general since any functional molecule can be produced from a description, not merely those that either happen to be able to self-reproduce, or those that can unfold and fold at will to be reproduced from available parts.

The symbol system, with its utilization of inert structures, opens up a whole new universe of functionality which is not available for purely dynamical self-replication. In this sense, it can evolve functions in an open-ended fashion. It introduces the third level of an evolving, embodied semiosis of autonomous systems with their environments, syntax, as defined by a construction code. This syntactic level of evolutionary systems is the foundation of the neo-Darwinist position and of all genetic based schemes found in evolutionary computation.

2. EMBODIED, EVOLVING, SEMIOSIS

2.1 Semiotics and Material Codes

Semiotics concerns the study of signs/symbols in three basic dimensions: *syntactics* (rule-based operations between signs within the sign system), *semantics* (relationship between signs and the world external to the sign system), and *pragmatics* (evaluation of the sign system regarding the goals of their users) [Morris, 1946]. When Von Neumann's universal constructor interprets a description to construct some automaton, a *semiotic code* is utilized to map instructions into physical actions to be performed. When the copier copies a description, only its *syntactic* aspects are replicated. Now, the language of this code presupposes a set of material primitives (e.g. parts and processes) for which the instructions are said to "stand for". In other words, descriptions are not universal as they refer to some material constituents which cannot be changed without altering the significance of the descriptions. We can see that a self-reproducing organism following this scheme is an entanglement of *symbolic controls* and *material constraints* which is closed on its semantics only through its repercussions in an environment. Howard Pattee [1982, 1995a] calls such a principle of self-organization *semantic closure*. Perhaps a better description would be to refer to it as *semiotic closure* since then the three semiotic dimensions of semantics, pragmatics and syntax would be accounted.

The semiotic code is defined by a small, finite, number of symbols (e.g. codons in DNA), which can encode a finite number of primitive parts (e.g. aminoacids). There is a finite number of functional structures which may be constructed with a given set of parts. This defines the representational power of a given symbol system. From coded messages, potentially, a trans-computational number of products can be constructed. However, since the products are dynamic and not symbolic structures, they will have different dynamic characteristics (for which they are ultimately selected). Moreover, the messages encoded stand for some arrangement of parts (strings of aminoacids) and not just the parts themselves. An arrangement of dynamic structures, however simple, tends to form a complex dynamic compound which will self-organize according to universal laws. These self-organized, coded, compounds can then interact with one another in many levels of organization which establish the hierarchical nature of evolution [Pattee, 1973; Laszlo, 1987]. In the computational lingo of Artificial Life, we can say that there is not a linear mapping of coded messages to functional products, rather messages encode dynamic structures which are then left to self-organize.

2.2 Embodiment and the Parts Problem

A particular materiality is tied to specific construction building blocks. The richer the parts, the smaller the required descriptions, but also the smaller the number of classifiable categories or constructed morphologies [Rocha, 1996, 1998a]. For instance, Von Neumann used simple building blocks such as "and" and "or" gates to build his automaton, which in turn required a 29 state cellular automata lattice and very complicated descriptions. Arbib[1966, 1967] was able to simplify von Neumann's model greatly by utilizing more complicated logical building blocks. Likewise, the genetic system does not need to describe all the chemical/dynamical characteristics of a "desired" protein, it merely needs to specify an aminoacid chain which will itself self-organize (fold) into a functional configuration with some reactive properties. A given set of parts such as amino acids, provides intrinsic dynamic richness which does not have to be specified by the symbol system on which construction commands are cast [Moreno, et al, 1994] making descriptions much smaller. Embodiment thus provides this kind of *material information compression.*

The other side of Embodiment, is that it also constrains the universe of possible constructions. Living organisms are morphologically restricted to those forms that can be made out of aminoacid chains through the genetic code, while in principle, a formal symbol system, stripped as it is from any materiality, can describe anything whatsoever. Of course, this 'in principle' is seriously, and easily, constrained by computational limits, as formal descriptions are much larger than material ones. A complete formal description of a protein would have to include all of its physical characteristics from the atomic to the chemical level, while a gene needs only a description of an aminoacid sequence. These two sides of embodiment can be classified as enabling and restraining constraints defined by a given materiality. On one hand embodiment enables emergent classification and smaller descriptions of this classification, and on the other it restrains the universe of possible classification.

2.3 Embodied, Evolving, Semiosis: Semiotic Closure and Syntactic Autonomy

Semiotics is a particularly intuitive way of thinking about Selected Self-Organization. It reminds us that the essential attribute of complex systems with emergent classification is the symbolic, that is, the existence of memory tokens that stand for dynamical configurations. The syntactic dimension can be equated with whatever type of memory tokens are utilized to refer to aspects of the complex system's environment. The semantic dimension refers to actual (self-organizing) dynamical configurations, with repercussions in an environment, and their relation to the memory tokens. The pragmatics dimension refers naturally to the selection of agents according to their behavior in an environment, which as discussed previously also defines semantics. Thus, selected self-organization refers to agents that observe a *semiotic closure* with their environments leading to an *embodied evolving semiosis* (EES) which can be open-ended if the natural symbols systems they implement are symbolic and follow von Neumann's scheme. Closure implies the existence of a functional loop or situational loop. That is, evolution, function, and meaning require an agent/environment coupling of structure that is grounded and evolves by pragmatic environmental selection.

Biological systems have developed a system of structural perturbation of their self-organization clearly based on a (genetic) code that virtually implements Von Neumann's scheme above. It is undeniable that this syntactic code is completely specified within organisms since its reading and constructing machinery is found within each cell: an autonomous code. Even though environmental conditions clearly affect what is decoded in different circumstances [Rocha, 1995], the code itself remains fixed. The ability to generate such a powerful system of structural perturbation is the one defining characteristic for all known life forms, which somehow evolved this extra level of *syntactic emergence* that is autonomous. Thus, the concept of autonomy is not enough to characterize living organisms, unless by that we mean, in addition to material autonomy (organizational closure), also *syntactic autonomy*. In other words, EES is based on organizational closure (self-organization), structural or semantic openness by virtue of a coupling to an environment, and syntactic autonomy.

3. SEMIOTICS OF THE ARTIFICIAL

3.1 Situated Action: Semantic Emergence

Artificial Life, mostly through the work of Brooks [1991], whose behavior language replaced the traditional high-level control of robots by a scheme of functional modularization by behavior generating modules, changed all this. Instead of a high-level computation of behavior, the bottom-up (emergentist) self-organization of simpler components produces a variety of behaviors depending on the interaction of a robot with its environment. "Situated" does not mean merely material, but inseparably interdependent. The material (structural) coupling of the robot with its environment is the source of behavior, and not just the robot control system alone. In other words, the modeling of living and cognitive systems is moved to the dynamics of self-organization of a network of components and its interaction with an environment (selected self-organization). Whichever way situated robots solve a problem, it is done by the construction of their own classifications, given the set of low level components they have available, as they interact with their environment, and not by externally imposed rules: emergent semantics.

One shortcoming of situated robotics in general, is that they implement solely the (reactive) semantic and pragmatic aspects of selected self-organization. The robots are indeed selected by their behaviors vis a vis a desired task, but they do not have a mechanism to implement open-ended evolution: the von Neumann scheme or syntactic autonomy. That is, they do not implement "efficient memory controlled *constructions* of real life that self-assemble at all-levels, from polymer folding to multicellular development". [Pattee, 1995b, page 36] Naturally, this is what situated robotics is trying to move into, though still with many difficulties to solve [Cariani, 1992]. In any case, situated robotics offers an excellent example of putting into practice some of the ideas behind EES and selected self-organization.

3.2 Emergent Particle Computation: Syntactic Autonomy

A very interesting problem that Genetic Algorithms (GA's) have been used successfully in, is the evolution of Cellular Automata (CA) rules for the solution of non-trivial tasks[1]. CA's are often used to study the behavior of complex systems, since they capture the richness of self-organizing dynamical systems which from local rules amongst their components, observe emergent behavior. Certain CA rules are capable of solving global tasks assigned to their lattices, even though their transition rules are local. One such tasks is usually referred to as the *density task*: given a randomly initialized lattice configuration, the CA should converge to a global state where all its cells are turned "ON" if there is a majority of "ON" cells in the initial configuration (IC), and to an all "OFF" state otherwise. This rule is not trivial because the local rules of the component cells do not have access to the entire lattice, but can only act on the state of their immediate neighborhood.

Crutchfield and Mitchell [1995] used a GA to evolve the one-dimensional CA rules for such a task. The GA found a number of fairly interesting rules, but 7 out of the 300 runs evolved very interesting rules (with high fitness) which create an intricate system of lattice communication. Basically, groups of adjacent cells propagate certain patterns across the lattice, which as they interact with other such patterns "decide" on the appropriate solutions for the lattice as a whole. An intricate system of signaling patterns and its communication syntax has been identified, and can be said to observe the emergence of embedded-particle computation in evolved CA's [Ibid; Hordijk, Crutchfield, Mitchell, 1996]. The emergent signals (or embedded particles) refer to the borders of the different patterns that develop on the space-time diagrams. If the areas inside these patterns are removed, their boundaries can be identified as system of signals with a definite syntax, or emergent logic grammar. This syntax is based on a small number of discrete signals, α, β, δ, γ, η, and μ, and a small number or rules such as: $\alpha + \delta \rightarrow \mu$, meaning that when signals α and δ collide, the μ signal results. Please refer to the references above for more details.

---

[1] This work has been mainly pursued by the Adaptive Computation Group of the Santa Fe Institute. I am indebted to Melanie Mitchell and Wim Hordijk for making much of their unpublished research available.

These experiments are very interesting because from the interaction of self-organization (CA's) and selection (GA) a very simple semantics emerges from the pragmatics of the GA: it either classifies its IC correctly or incorrectly. Now, most CA rules evolved with this set up show a very simple space-time patterns: they try to solve the problem by block-expansion, that is, when large neighborhoods of either try "ON" or "OFF" states exist in the IC, they are expanded. However, a few of these rules (7 of about 300) evolved a more complicated way of dealing with the problem by coming up with the system of particle computation. In other words, these CA rules rely on a system of personal emergent signals used to communicate across the lattice and compute the answer to the task. These signals effectively establish an autonomous sign system that granted great selective advantage to the rules capable of developing it.

These signals, even though being a small set of discrete entities, are not full-fledged symbols in the senses described in section 2, because they do not possess the degree of arbitrariness required of pure symbols. If we were able to evolve a CA rule that, with the same signal grammar, could solve different tasks, then we would have the necessary degree of arbitrariness to refer to them at symbols. In any case, these experiments possess the intertwined semantics and pragmatics of selected self-organization, plus a primordial autonomous syntax in an artificial environment (though a far cry from Von Neumann's scheme). In this sense they are a case of a purely computational semiotic closure, which represents a truly exciting new development in complex systems research and artificial life. We should now empower them with richer dynamics in order to try to achieve the degree of arbitrariness required of true symbols that emerge from artificial matter.

4. CONTEXTUAL GENETIC ALGORITHMS: RICHER GENETIC SEMIOTICS

One important difference between evolutionary computation and biological genetic systems, lies on the connection between descriptions and solutions, between signifier (genotype) and signified (phenotype). In genetic algorithms the relation between the two is linear and direct: one description, one solution. In the biological genetic system, on the other hand, there exists a multitude of processes, taking place between the transcription of a description and its expression, responsible for the establishment of an one-to-many relation between signifier and signified. In order to expand GA's to model more aspects of the semiotics of the genetic system[2], we can act on both sides of this code (figure 1). Since the systems which implement the development of solutions and the manipulation of chromosomes can receive inputs other than chromosomes (e.g. environmental observables), genetic transcription and solution development may be dependent on contextual factors. Hence, I refer to these expanded GA's as *Contextual Genetic Algorithms* (CGA's) [Rocha, 1995, 1997, 1998b].

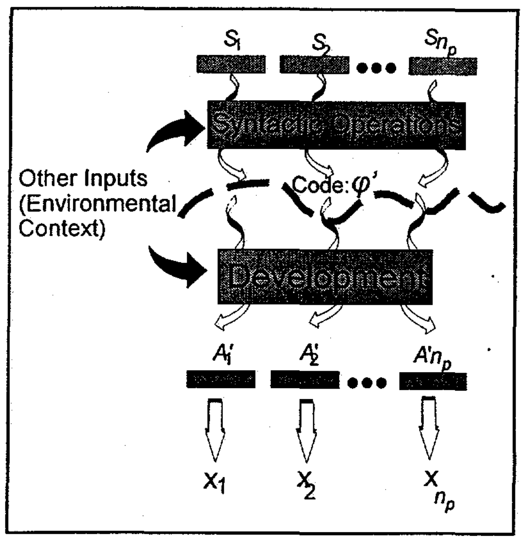

Figure 1: Indirect Encoding and Extended Syntax in Contextual Genetic Algorithms

The genetic semiotics described in Rocha [1995, 1997, 1998a] expands the syntax of the traditional genetic semiotics by postulating richer symbolic interactions than mere DNA/RNA transcription. If a second type of symbols exists, which operates on genetic messages and in so doing change the latter's encoded meaning, their access to environmental information can provide the genetic system real-time control of genetic expression according to context. In Rocha [1995, 1997] it was shown that RNA Editing offers the genetic system a richer syntax that facilitates contextual development and gives it the ability to link changes in the environment with internal parameters, which is useful for phenotypical development in changing environments. A formal model of this system was also proposed based on CGA's.

[2] The semiotics of the genetic system has been explored conceptually in Rocha [1995, 1997, 1998a] here I am exploring this issue computationally.

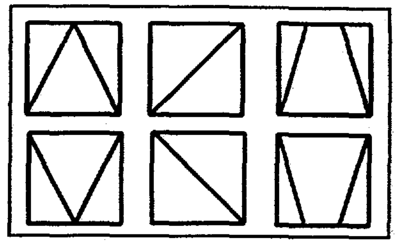

Figure 2: Pool of Simple Fuzzy Set Shapes

Fuzzy sets [Zadeh, 1965] are unconstrained mappings to the unit interval. A fuzzy set is able to capture the state of any system whose components's states are defined by numerical values, since any real interval can be mapped to the unit interval. Given a universal set X containing all the elements (variables) x of system X, a fuzzy set A is defined by the (membership) function $A(x): X \rightarrow [0,1]$. It represents a particular state of system X. Consider an initial fuzzy set $A_0(x) = 0.5$ for all x of X, it is the initial condition of our system X. Consider now a sequence of n fuzzy sets $F_1, F_2, \cdots, F_n$, which are applied to A_0 with the sequence of n fuzzy operations $\odot_1, \odot_2, \cdots, \odot_n$. This sequence of n fuzzy sets F_i and operations $\odot_i$ is a program to develop $A_n(x)$from $A_0(x)$ in n steps: a *fuzzy development program* (FDP). Notice that if some of the operations $\odot_i$ are non-commutative, then the sequence of the FDP matters, which is desired of developmental models.

Consider a small pool of n_F typical fuzzy set shapes referred to as $\mathscr{F}$ (figure 2, details in Rocha [1997, 1998b]). Consider a small pool of n_O fuzzy set operations referred to as $\mathcal{O}$. These operations range from commutative operations such as fuzzy union ($\cup$) and intersection ($\cap$) to non-commutative operations such as $A \cap \overline{B}$ and $\overline{A} \cup B$. A FDP is a sequence (string) S of n fuzzy sets $F_i \in \mathscr{F}$ and n operations $\odot_i \in \mathcal{O}$: $S = \odot_1 F_1 \odot_2 F_2 \cdots \odot_n F_n$ which are to be applied in sequence to the initial state $A_0(x)$ of system X. Consider a partition of X in an even number n_X of parts. If $n_X = 8$, X is divided in equal octants. Each F_i of the FDP S is associated with a specific part of X: $p = 1, \cdots n_X$. One other parameter is $s = 1, \cdots n_X/2$, which represents the number of parts of X that shape F_i should be stretched over. Figure 3 shows the universal set X divided in octants ($n_X = 8$), and a triangular fuzzy shape (dotted line) being applied to the sixth octant ($p = 6$) with a stretch of two octants for each side of the sixth octant ($s = 2$). A final parameter $d = 1, \cdots n_X/2$ is defined which represents the number of times shape F_i is going to be repeated in the portion of X given by p and s. In figure 3, the dark line represents $d = 1$, meaning that the triangular shape is repeated once. If d were 2, then the triangular shape would be narrowed in half, and repeated twice over the interval of X given by parts 4 to 8 (p=6, s=2), in figure 3 this is shown by the lighter line.

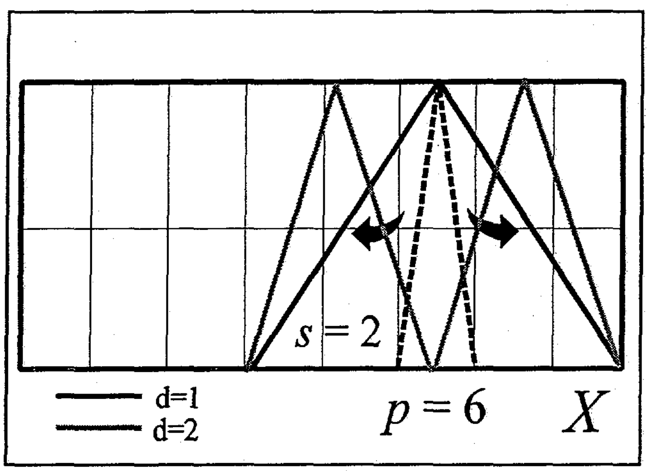

Figure 3: Triangular Fuzzy Set Shape Applied to Sixth Octant of X (p=6) and stretched 2 octants to each side (s=2). Shown for two division parameter (d=1 and d=2).

Each fuzzy set F_i of the FDP S is defined by a fuzzy set shape from $\mathscr{F}$, the part of X, p, the stretch s, and the division factor d. Thus, the information required to describe the FDP S does not depend on the size of X but on the parameters n_X, n_F, and n_O. To specify the part of X chosen (e.g. octant), no information regarding the elements of X in that part is required. We need only $\log_2 n_F$ and $\log_2 n_O$ bits of information to identify n_F fuzzy set shapes and n_O operations from $\mathscr{F}$ and $\mathcal{O}$ respectively. We further need $\log_2 n_X$ bits to describe the position parameter p, and $2 \times \log_2 (n_X/2)$ bits to describe the stretch parameter s and division factor d. Therefore, $\log_2 n_F + \log_2 n_X + 2 \times \log_2 (n_X/2)$ bits are required to identify a fuzzy set F_i in a FDP S, and $\log_2 n_O$ bits to identify its associate operation. For example, if there are 16 possible fuzzy set shapes ($|\mathscr{F}|=16$) and 16 possible fuzzy logic operations ($|\mathcal{O}|=16$), and X is divided in 16 parts, then $\log_2 16 + \log_2 16 + 2 * \log_2 8 + \log_2 16 = 4 + 4 + 6 + 4 = 18$ bits are required for each pair fuzzy set/operation in the FDP. If the length of the FDP is $n = 8$, then the FDP S requires 144 bits to be described, that is, a 144 long bit string. Notice that this value is independent of the cardinality of X or its parts. In summary, a FDP S requires the following number of bits of information to be described for any X:

$$v = n\left[\log_2 n_F + \log_2 n_X + 2\cdot\log_2\left(\frac{n_X}{2}\right) + \log_2 n_O\right] \quad (1)$$

Utilizing the scheme above, states $A(x)$ of system X can be evolved (regarding some fitness function) through a genetic algorithm (GA) which codes not for the states themselves, but for FDP's whose sequences of operations produce $A(x)$. The chromosomes of the GA are bit strings of length v encoding FDP programs, which develop into (continuous) solutions $A(x)$ that are not directly encoded in the bit strings (figure 4). In the computational realm, we can ease the chromosomes of a GA from having to describe every detail of the solutions through an indirect encoding scheme as the CGA. However, some form of that description is unavoidable somewhere else in a computer implementation as everything must be specified in computational environments. EES in the computer realm, requires the simulation of materiality in order to implement selected self-organization. In this case, the material dynamic constraints are simulated with FDP's.

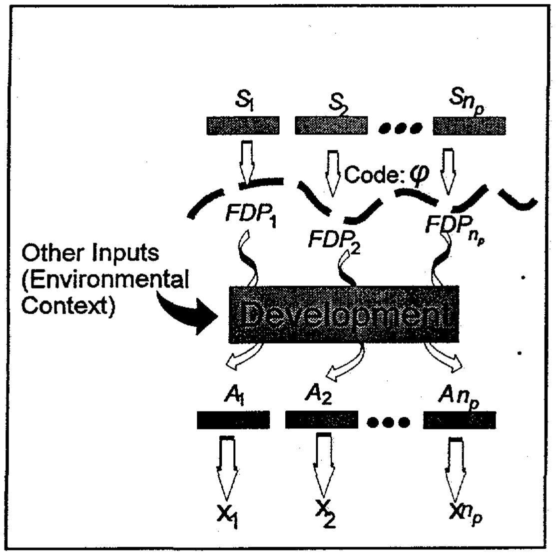

Figure 4: CGA with Developmental Stage based on FDP's. Chromosomes encode FDP's which develop into Fuzzy Sets, standing for Solutions.

It is therefore important to have as simple as possible a description of the dynamic building blocks for the indirectly encoded GA solutions. The fuzzy set scheme above is an attempt precisely at that. If a true computational dynamic system, such as boolean networks, is used, every time a chromosome is decoded into a set of rules to build the network that will self-organize into a solution, the network will actually have to be implemented and run for a number of cycles in all its details. Thus at each step of the GA, the evaluation of a chromosome relies on a computationally demanding evaluation procedure that must implement and observe the dynamic behavior of a network. By contrast, the FDP is not a dynamical system, it is a sequence dependent (non-commutative) procedure for constructing the state of a dynamical system without actually running it. FDP's depend on a small vocabulary of simple parts, such as fuzzy set shapes and operations, which are described by very simple logical equations.

FDP's offer a simulation of true developmental dynamics which preserves some important characteristics of development such as being constructed from a small pool of (simulated material) parts which interact with one-another through a development program in order to define a final state. Since the operations between these parts can be non-commutative, the order of application of parts is important as it leads to different final results. They allow a transformation from a small discrete, boolean, domain, to a large continuous domain without very computationally expensive implementations of true dynamical systems. Furthermore, they can represent the state of any dynamical system, discrete or continuous. Coupling them to GA's, allows the establishment of computational counterparts of EES with the immediate advantage of tremendous genetic information compression. Good results were obtained in the application of CGA's to demanding problems such as the evolution of neural network weights and cellular automata rules [Rocha, 1997, 1998b]

5. Artificial Matter and Symbols

Different approaches to including a more accurate picture of the interaction of matter and symbols that we find in real evolutionary systems into artificial realms were discussed above. After a conceptualization of this issue in terms of semiotic closure and EES, three approaches were discussed: situated action, embedded particle computation in CA's, and CGA's. The first of these approaches relies on the utilization of real material constraints of real environments, that is, the emergent semantics situated robotics are capable of developing (see section 3) is real. The challenge is to construct robots that are capable of semiotic closures with more complicated syntactic autonomies, namely a sort of genetic autonomy, so that they become capable of open-ended evolutionary potential. In this sense, situated robots are not yet (semiotically) autonomous as biological organisms are.

The experiments leading to the emergence of particle computation in CA's, represents the emergence of a limit-case semiotic autonomy in a completely computational environment. As such, it represents an abstract model of how signs can emerge from purely dynamical interactions under an artificial EES. Surely the dynamics can be made more complex and realistic, but such experiments should open the fields of complex systems and artificial life to seek those rules than may be able to show more characteristics desired of pure symbols (see section 4). These experiments seem to indicate that it is possible to evolve symbols from artificial matter.

The CGA is an instance of the EES ideas presented in sections 1 and 2. The developmental stage is in effect simulating some specific dynamic constraints on evolution posed by a simulated embodiment of the evolving semiotic system. Fuzzy indirect encoding captures computationally the advantages of materiality by reducing genetic descriptions, which may be very relevant in practical domains such as data mining. However, it does also capture the reverse side of embodiment, that is the limiting constraints that a given materiality poses on evolving systems [For more details see Rocha, 1997, 1998b]. Given a specific simulated embodiment defined by the particular fuzzy set shapes and operations used, the algorithm cannot reach all portions of space of solutions. It can only reach those solutions that can be constructed from the manipulation of the allowed fuzzy sets and operations – the building blocks. This echoes my previous observation, that the genetic system is similarly not able to evolve anything whatsoever, but only forms that can be built out of aminoacid chains.

Another way to think of this, is that in traditional GA's, each position of the solution vector maps to only one position in the chromosomes (allele), which can be independently mutated. In other words, the mutation of one bit in the chromosome will affect only one component in the solution vector. Whereas in the indirect encoding scheme, each bit of the chromosomes affects several elements of the solution vector non-linearly. In the scheme here utilized, flipping one bit in the FDP may result in changing a fuzzy set operation to another, thus causing the fuzzy sets it operates on to be connected in a totally different manner for all its elements. Thus, one single bit of indirectly encoded chromosomes can affect many, potentially all, elements of the solution vector as the development program is changed. This introduces *epistasis* to evolutionary computation, which we know exists in natural genetic systems.

All of these aspects of indirectly encoded GA's, both enabling constraints such as genetic information compression, and limiting constraints such as reduction of the space of solutions, are desired of models of evolutionary systems. But what does it mean for practical applications? Genetic compression is obviously a plus, but the limiting constraints may be problematic if the reduced search space includes only mediocre solutions to our problems. When genetic descriptions are not very large, the only reason to use indirect encoding is if we wish to avoid very random solution vectors that tend to be produced with GA's. This was the case of the evolution of neural network weights in section 5. The indirect encoding scheme was able to produce better cross-validation results. Due to its intrinsic order, its solutions did not overly adapt to the patterns in the learning set. When the genetic descriptions are very large, or require real-encoding, then it is advantageous to use indirect encoding. Sometimes, even if the reduced solution space is mediocre, such less that optimum solutions might be all that we can hope to find in huge solution spaces, inaccessible to standard GA's due to computational limitations.

References

Arbib, M.A. [1966]."A simple self-reproducing universal automaton." *Information and Control,* 9, pp. 177-189.

Arbib, M.A. [1967]."Automata theory and development: Part I." *Theoretical Biology.* Vol. 14, pp. 131-156.

Brooks, R.A. [1991]."Intelligence without reason." In: *Proceedings of the 12th International Conference on Artificial Intelligence, San Mateo 2992.*. Morgan Kaufmann.

Cariani, Peter [1992]."Emergence and Artificial Life." In: *Artificial Life II.* C. Langton, C. Taylor, J.D. Farmer, and S. Rasmussen (eds.). SFI Series in the Sciences of Complexity. Addison-Wesley, pp. 775-797.

Clark, Andy [1996]."Happy couplings: emergence and explanatory interlock." In: *The Philosophy of Artificial Life.* M. Boden (ed.). Oxford University Press, pp. 262-281.

Crutchfield, J.P. and M. Mitchell [1995]."The evolution of emergent computation." *Proc. National Acadamy of Sciences, USA, Computer Sciences.* Vol. 92, pp. 10742-10746..

Dellaert, Frank and R.D. Beer [1994]."Toward an evolvable model of development for autonomous agent synthesis." In: *Artificial Life IV.* R. Brooks and P. Maes (eds.). MIT Press. pp. 246-257.

Holland, J.H. [1975]. *Adaptation in Natural and Artificial Systems.* University of Michigan Press.

Hordijk, W., J.P. Crutchfield, and M. Mitchell [1996]."Embedded-particle computation in evolved cellular automata." In: *Proceedings of Physics and Computation 96.* . In Press.

Kauffman, S. [1993]. *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press.

Kitano, H. [1994]."Evolution of metabolism for morphogenesis." In: *Artificial Life IV.* R.A. Brooks and P. Maes (eds.). MIT Press. pp 49-58.

Kitano, H. [1995]."Cell differentiation and neurogenesis in evolutionary large scale chaos." In: *Advances in Artificial Life.* F. Moran et al (Eds.). Springer-Verlag. pp. 341-352..

Laszlo, E. [1987]. *Evolution: The Grand Synthesis.* New Science Library.

Maturana, H. and F. Varela [1987]. *The Tree of Knowledge: The Biological Roots of Human Understanding.* New Science Library.

Mitchell, Melanie [1996]. *An Introduction to Genetic Algorithms.* MIT Press.

Pattee, Howard H. (Ed.) [1973]. *Hierarchy Theory: The Challenge of Complex Systems.* George Braziller.

Pattee, Howard H. [1982]."Cell psychology: an evolutionary approach to the symbol-matter problem." *Cognition and Brain Theory.* Vol. 5, no. 4, pp. 191-200.

Pattee, Howard H. [1995a]."Evolving self-reference: matter, symbols, and semantic closure." *Communication and Cognition - Artificial Intelligence.* Vol. 12, nos. 1-2, pp. 9-27.

Pattee, Howard H. [1995b]."Artificial life needs a real epistemology." In: *Advances in Artificial Life.* F. Moran, et al. Springer-Verlag. pp 23-38.

Rocha, Luis M. [1995]."Contextual genetic algorithms: evolving developmental rules." In: *Advances in Artificial Life.* J. Moran, A. Moreno, J.J. Merelo, and P. Chacon (Eds.). Springer -Verlag, pp. 368-382.

Rocha, Luis M. [1996]."Eigenbehavior and symbols." *Systems Research.* Vol. 13, No. 3, pp. 371-384.

Rocha, Luis M. [1997]. *Evidence Sets and Contextual Genetic Algorithms: Exploring Uncertainty, Context and Embodiment in Cognitive and biological Systems.* PhD. Dissertation. State University of New York at Binghamton.

Rocha, Luis M. [1998a]."Selected self-organization and the Semiotics of Evolutionary Systems." In: *Evolutionary Systems.* S. Salthe and G. Van de Vijver. Kluwer Academic Publishers. (in press).

Rocha, Luis M. [1998b]."Semiotic Closure and Contextual Genetic Algorithms." *Complex Systems.* (Submitted). .

van Gelder, Tim [1991]."What is the 'D' in 'PDP': a survey of the concept of distribution." In: *Philosophy and Connectionist Theory.* W. Ramsey et al. Lawrence Erlbaum.

Varela, F.J. and P. Bourgine [1992]. *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life.* MIT Press.

von Neumann, J. [1966]. *The Theory of Self-Reproducing Automata.* University of Illinois Press.

Zadeh, Lofti A. [1965]."Fuzzy Sets." *Information and Control.* Vol. 8, pp. 338-353.