

# Hierarchical Self-Organization in the Finitary Process Soup

Olof Görnerup<sup>\*,\*\*,‡</sup>  
Chalmers University of Technology  
Santa Fe Institute

James P. Crutchfield<sup>†,‡</sup>  
University of California, Davis  
Santa Fe Institute

**Abstract** Current analyses of genomes from numerous species show that the diversity of their functional and behavioral characters is not proportional to the number of genes that encode the organism. We investigate the hypothesis that the diversity of organismal character is due to hierarchical organization. We do this with the recently introduced model of the *finitary process soup*, which allows for a detailed mathematical and quantitative analysis of the population dynamics of structural complexity. Here we show that global complexity in the finitary process soup is due to the emergence of successively higher levels of organization, that the hierarchical structure appears spontaneously, and that the process of structural innovation is facilitated by the discovery and maintenance of relatively noncomplex, but general, individuals in a population.

## Keywords

Pre-biotic evolution, structural complexity, hierarchy, self-organization, networks, population dynamics

## I Introduction

Recent estimates have shown that the genomes of many species consist of a surprisingly similar number of genes despite some species being markedly more sophisticated and diverse in their behaviors. Humans have only 30% more genes than the worm *Caenorhabditis elegans*; humans, mice, and rats have nearly the same number [15, 18]. Moreover, many of those genes serve to maintain elementary processes and are shared across species, which greatly reduces the number of genes available to account for diversity. One concludes that individual genes cannot directly code for the full array of individual functional and morphological characters of a species, as genetic determinism would have it. From what, then, do the sophistication and diversity of organismal form and behavior arise?

Here we investigate the hypothesis that these derive from a hierarchy of interactions between genes and between interacting gene complexes. A hierarchy of gene interactions, being composed of subsets of available genes, allows for an exponentially larger range of functions and behaviors than direct gene-to-function coding. We will use a recently introduced pre-biotic evolutionary model—the *finitary process soup*—of the population dynamics of structural complexity [6]. Specifically, we will show that global complexity in the finitary process soup is due to the emergence of successively higher levels of organization. Importantly, hierarchical structure appears spontaneously and is

\* Corresponding author

\*\* Department of Physical Resource Theory, Chalmers University of Technology, 412 96 Göteborg, Sweden. E-mail: olof.gornerup@fy.chalmers.se

† Center for Computational Science & Engineering and Physics Department, University of California, Davis, One Shields Avenue, Davis CA 95616. E-mail: chaos@cse.ucdavis.edu

‡ Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501.

facilitated by the discovery and maintenance of relatively noncomplex, but general individuals in a population. These results, in concert with the minimal assumptions and simplicity of the finitary process soup, strongly suggest that an evolving system's sophistication, complexity, and functional diversity derive from its hierarchical organization.

## 2 Modeling Pre-biology

Prior to the existence of highly sophisticated entities acted on by evolutionary forces, replicative objects relied on far more basic mechanisms for maintenance and growth. However, these objects managed to transform not only themselves, but also indirectly the very transformations by which they changed [20], eventually supporting the mechanisms of natural selection. How did the transition from raw interaction to evolutionary change take place? Is it possible to pinpoint generic properties, however basic, that would have enabled a system of simple interacting objects to take the first few steps toward biotic organization?

To explore these questions in terms of structural complexity we developed a theoretical model borrowing from computation theory [13] and computational mechanics [9, 5]. In this system—the *finitary process soup*—elementary objects, as represented by  $\epsilon$ -machines, interact and generate new objects in a well-stirred flow reactor.

Choosing  $\epsilon$ -machines as the interacting, replicating objects, it turns out, brings a number of advantages. Most particularly, there is a well-developed theory of their structural properties found in the framework of computational mechanics. In contrast with individuals in previous, related pre-biotic models—such as machine language programs [16, 17, 19, 1], tags [10, 2],  $\lambda$ -expressions [11], and cellular automata [7],  $\epsilon$ -machines have a well-defined (and calculable) notion of structural complexity. For the cases of machine language and  $\lambda$  calculus, in contrast, it is known that algorithms do not even exist to calculate such properties, since these representations are computation-universal [3]. Another important distinction with prior pre-biotic models is that the individuals in the finitary process soup do not have two separate modes of operation—one of representation or storage and one for functioning and transformation. The individuals are simply objects whose internal structure determines how they interact. The benefit of this when modeling prebiotic evolution is that there is no assumed distinction between gene and protein [21, 24] or between data and program [16, 17, 19, 1, 14].

## 3 $\epsilon$ -Machines

Individuals in the finitary process soup are objects that store and transform information. In the vocabulary of information theory, they are *communication channels* [4]. Here we focus on a type of finite-memory channel, called a finitary  $\epsilon$ -machine, as our preferred representation of an evolving information-processing individual. To understand what this choice captures, we can think of these individuals in terms of how they compactly describe stochastic processes.

A *process* is a discrete-valued, discrete-time stationary stochastic information source [4]. A process is most directly described by the bi-infinite sequence it produces of random variables  $S_t$  over an alphabet  $\mathcal{A}$ :

$$\overleftrightarrow{S} = \dots S_{t-1} S_t S_{t+1} \dots \quad (1)$$

and the distribution  $P(\overleftrightarrow{S})$  over those sequences. At each moment  $t$ , we think of the bi-infinite sequence as consisting of a history  $\overleftarrow{S}_t = \dots S_{t-3} S_{t-2} S_{t-1}$  and a future  $\overrightarrow{S}_t = S_t S_{t+1} S_{t+2} \dots$  subsequence:  $\overleftrightarrow{S} = \overleftarrow{S}_t \overrightarrow{S}_t$ .

A process stores information in its set  $\mathcal{S}$  of *causal states*. Mathematically, these are the members of the range of the map  $\epsilon : \overleftarrow{\mathcal{S}} \mapsto 2^{\overleftarrow{\mathcal{S}}}$  from histories to sets of histories,

$$\epsilon(\overleftarrow{s}) = \{ \overleftarrow{s}' \mid P(\overleftarrow{\mathcal{S}} \mid \overleftarrow{\mathcal{S}} = \overleftarrow{s}) = P(\overleftarrow{\mathcal{S}} \mid \overleftarrow{\mathcal{S}} = \overleftarrow{s}') \}, \tag{2}$$

where  $2^{\overleftarrow{\mathcal{S}}}$  is the power set of histories  $\overleftarrow{\mathcal{S}}$ . That is, the causal state  $\mathcal{S}$  of a history  $\overleftarrow{s}$  is the *set* of histories that all have the same probability distribution of futures. The transition from one causal state  $\mathcal{S}_i$  to another  $\mathcal{S}_j$  while emitting the symbol  $s \in \mathcal{A}$  is given by a set of labeled transition matrices:  $\mathcal{T} = \{ T_{ij}^{(s)} : s \in \mathcal{A} \}$ , in which

$$T_{ij}^{(s)} \equiv P(\mathcal{S}' = \mathcal{S}_j, \overleftarrow{s}^1 = s \mid \mathcal{S} = \mathcal{S}_i), \tag{3}$$

where  $\mathcal{S}$  is the current casual state,  $\mathcal{S}'$  its successor, and  $\overleftarrow{s}^1$  the next symbol in the sequence.

A process'  $\epsilon$ -*machine* is the ordered pair  $\{ \mathcal{S}, \mathcal{T} \}$ . Finitary  $\epsilon$ -machines are stochastic finite-state machines with the following properties [9]: (i) All recurrent states form a single strongly connected component. (ii) All transitions are deterministic in the specific sense that a causal state together with the next symbol determines a unique next state. (iii) The set of causal states is finite and minimal.

In the finitary process soup we use the alphabet  $\mathcal{A} = \{ 0 \mid 0, 0 \mid 1, 1 \mid 0, 1 \mid 1 \}$  consisting of pairs *in* \ | *out* of input and output symbols over a binary alphabet  $\mathcal{B} = \{ 0, 1 \}$ . When used in this way,  $\epsilon$ -machines read in strings over  $\mathcal{B}$  and emit strings over  $\mathcal{B}$ . Accordingly, they should be viewed as mappings from one process  $\overleftrightarrow{\mathcal{S}}_{\text{input}}$  to another  $\overleftrightarrow{\mathcal{S}}_{\text{output}}$ . They are, in fact, simply functions, each with a domain (the set of strings that can be read) and with a range (the set of strings that can be produced). In this way, we consider  $\epsilon$ -machines as models of objects that store and transform information. In the following we will take the transitions from each causal state to have equal probabilities. Figure 1 shows several examples of simple  $\epsilon$ -machines.

Given that  $\epsilon$ -machines are transformations, one can ask how much processing they do—how much structure do they add to the inputs when producing an output? Due to the properties mentioned above, one can answer this question precisely. Ignoring input and output symbols, the state-to-state transition probabilities are given by an  $\epsilon$ -machine's *stochastic connection matrix*:  $\mathbf{T} \equiv \sum_{\mathcal{S} \in \mathcal{A}} T^{(\mathcal{S})}$ . The causal-state probability distribution  $p_{\mathcal{S}}$  is given by the left eigenvector of  $\mathbf{T}$  associated with eigenvalue 1 and normalized in probability. If  $M$  is an  $\epsilon$ -machine, then the amount of information storage that it has, and that it can add to an input process, is given by  $M$ 's *structural complexity*

$$C_{\mu}(M) \equiv - \sum_{v \in \mathcal{S}} p_{\mathcal{S}}(v) \log_2 p_{\mathcal{S}}(v). \tag{4}$$

### 4 $\epsilon$ -Machine Interaction

$\epsilon$ -machines interact by functional composition. Two machines  $T_A$  and  $T_B$  that act on each other result in a third  $T_C = T_B \circ T_A$ , where  $T_C$  (i) has the domain of  $T_A$  and the range of  $T_B$  and (ii) is

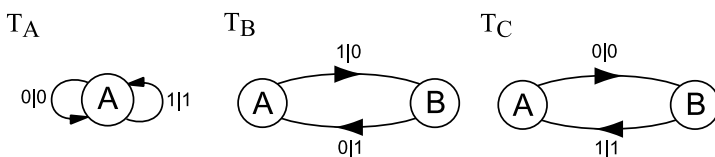


Figure 1. Example  $\epsilon$ -machines:  $T_A$  has a single causal state and, according to its transition labels, is the identity function.  $T_B$  consists of causal states  $A$  and  $B$  and two transitions.  $T_B$  accepts two input strings, either  $1010\dots$  or  $0101\dots$ , and flips 0s to 1s and vice versa as it produces an output string. Note that the function's domain and range are the same.  $T_C$  has the same domain and range as  $T_B$ , but does not exchange 0s and 1s.

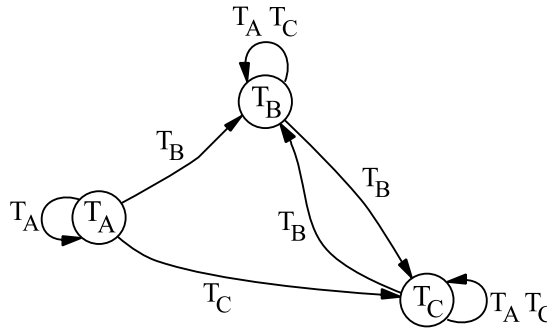


Figure 2. Interaction network for the  $\epsilon$ -machines of Figure 1. It is a meta-machine.

minimized. If  $T_A$  and  $T_B$  are incompatible (e.g., the domain of  $T_B$  does not overlap with the range of  $T_A$ ), the interaction produces nothing—it is considered elastic. During composition the size of the resulting  $\epsilon$  machine can grow very rapidly (geometrically):  $|T_C| \leq |T_B| \times |T_A|$ .

### 4.1 Interaction Network

We monitor the interactions of objects in the soup via the *interaction network*  $\mathcal{G}$ . This is represented as a graph whose nodes correspond to  $\epsilon$ -machines and whose transitions correspond to interactions. If  $T_k = T_j \circ T_i$  occurs in the soup, then the edge from  $T_i$  to  $T_k$  is labeled  $T_j$ . One can represent  $\mathcal{G}$  with the binary matrices:

$$\mathcal{G}_{ij}^{(k)} = \begin{cases} 1 & \text{if } T_k = T_j \circ T_i, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

For the set of  $\epsilon$ -machines in Figure 1, for example, we have the interaction graph shown in Figure 2 that is given by the matrices

$$\mathcal{G}^{(A)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{G}^{(B)} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \text{and} \quad \mathcal{G}^{(C)} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

To measure the diversity of interactions in a population we define the *interaction network complexity*

$$C_{\mu}(\mathcal{G}) = - \sum_{f_i, f_j, f_k > 0} \frac{v_{ij}^k}{V^k} \log_2 \frac{v_{ij}^k}{V^k}, \tag{6}$$

where

$$v_{ij}^k = \begin{cases} f_i f_j & \text{if } T_k = T_j \circ T_i \text{ has occurred,} \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

$V^k = \sum^{lm} v_{lm}^k$  is a normalizing factor, and  $f_i$  is the fraction of  $\epsilon$ -machine type  $i$  in the soup. In order to emphasize our interest in actual reproduction pathways, we consider only those that have occurred in the soup.

### 4.2 Meta-machines

Given a population  $\mathcal{P}$  of  $\epsilon$ -machines, we define a *meta-machine*  $\Omega \subset \mathcal{P}$  to be a connected set of  $\epsilon$ -machines that is invariant under composition. That is,  $\Omega$  is a meta-machine if and only if (i)  $T_j \circ T_i \in \Omega$  for all  $T_i, T_j \in \Omega$ , (ii) for all  $T_k \in \Omega$ , there exists  $T_i, T_j \in \Omega$  such that  $T_k = T_j \circ T_i$ , and (iii) there is a nondirected path between every pair of nodes in  $\Omega$ 's interaction network  $\mathcal{G}_\Omega$ . The interactions in Figure 2 describe a meta-machine of Figure 1's  $\epsilon$ -machines.

The meta-machine captures the notion of a self-replicating and autonomous entity and is consistent with Maturana and Varela's *autopoietic set* [23], Eigen and Schuster's *hypercycle* [22], and Fontana and Buss' *organization* [12].

## 5 Population Dynamics

We employ a continuously stirred flow reactor with an influx rate  $\Phi_{in}$  that consists of a population  $\mathcal{P}$  of  $N$   $\epsilon$ -machines. The dynamics of the population is iteratively ruled by compositions and replacements as follows:

### 1. $\epsilon$ -machine *Generation*:

- (a) With probability  $\Phi_{in}$  generate a random  $\epsilon$ -machine  $T_R$  (*influx*).
- (b) With probability  $1 - \Phi_{in}$  (*reaction*):
  - i. Select  $T_A$  and  $T_B$  randomly.
  - ii. Form the composition  $T_C = T_B \circ T_A$ .

### 2. $\epsilon$ -machine *Outflux*:

- (a) Select an  $\epsilon$  machine  $T_D$  randomly from the population.
- (b) Replace  $T_D$  with either  $T_C$  or  $T_R$ .

Below,  $T_R$  will be uniformly sampled from the set of all two-state  $\epsilon$ -machines. This set is also used when initializing the population. The insertion of  $T_R$  corresponds to the influx, while the removal of  $T_D$  corresponds to the outflux. The latter keeps the population size constant. Note that there is no spatial dependence in this model;  $\epsilon$ -machines are picked uniformly from the population for each replication. The finitary process soup here is a well-stirred gas of reacting objects.

When there is no influx ( $\Phi_{in} = 0$ ) and the population is closed with respect to composition, the population dynamics is described by a finite-dimensional set of equations:

$$\mathbf{f}_t^{(k)} = \mathbf{f}_{t-1} \cdot \mathcal{G}_{ij}^{(k)} \cdot \mathbf{f}_{t-1}^T Z^{-1}, \tag{8}$$

where  $\mathbf{f}_t^{(k)}$  is the frequency of  $\epsilon$ -machine type  $k$  at time  $t$ , and  $Z^{-1}$  is a normalization factor.

In addition to capturing the notion of self-replicating entities, meta-machines also describe an important type of invariant set of the population dynamics. Formally, we have

$$\Omega = \mathcal{G} \circ \Omega, \tag{9}$$

where  $\Omega$  is a meta-machine and  $\mathcal{G}$  is the interaction network. These invariant sets can be stable or unstable under the population dynamics. Note that the meta-machine of Figure 2 is unstable: Only  $T_A$  produces  $T_A$ s. Thus, over time the population dynamics will decay to the meta-machine of Figure 3, which describes a soup consisting only of  $T_B$ s and  $T_C$ s. This example also happens to illustrate that copying—implemented here by the identity object  $T_A$ —need not dominate the population and so does not have to be removed by hand, as done in several prior pre-biotic models. It can decay away due to the intrinsic population dynamics.

### 6 Simulations

A system constrained by closure forms one useful base case that allows for a straightforward analysis of the population dynamics. It does not permit, however, for the innovation of structural novelties in the soup on either the level of individual objects ( $\epsilon$ -machines) or the level of their interactions. What we are interested in is the possibility of open-ended evolution of  $\epsilon$ -machines and their meta-machines. When enabled as an open system, both with respect to composition and influx, the soup constitutes a constructive dynamical system and the population dynamics of Equation 8 do not strictly apply. (The open-ended population dynamics of epochal evolution is required [8].)

We first set the influx rate to zero in order to study dynamics that are ruled only by compositional transformations. One important first observation is that almost the complete set of machine types that are represented in the soup’s initial random population is replaced over time. Thus, even at the earliest times, the soup generates genuine novelty. The population-averaged individual complexity  $\langle C_\mu(T) \rangle$  increases initially, as Figure 4a ( $\Phi_{in} \approx 0$ ) from [6] shows. The  $\epsilon$ -machines are to some extent shaped by the selective pressure coming from outflux and by geometric growth due to composition. The turnover is due to the dominance of nonreproducing  $\epsilon$ -machines in the initial population.  $\langle C_\mu(T) \rangle$  subsequently declines, since it is favorable to be simple in that it takes a more extensive stochastic search to find reproductive interactions that include more complex  $\epsilon$ -machines.

Note (Figure 4b,  $\Phi_{in} \approx 0$ ) that the run-averaged interaction complexity  $\langle C_\mu(\mathcal{G}) \rangle$  reaches a significantly higher value than  $\langle C_\mu(T) \rangle$ , implying that the population’s structural complexity derives from its network of interactions rather than the complexity of its constituent individuals.  $\langle C_\mu(\mathcal{G}) \rangle$  continues to grow while compositional paths are discovered and created. A maximum is eventually reached after which  $\langle C_\mu(\mathcal{G}) \rangle$  declines and settles down to zero when one single type of self-reproducing  $\epsilon$ -machine takes over the whole population.

By monitoring the individual run values of  $C_\mu(\mathcal{G})$  rather than the ensemble average, one sees that they form plateaus as shown in Figure 5. The plateaus—at  $C_\mu(\mathcal{G}) = 4$  bits and, most notably, at  $C_\mu(\mathcal{G}) = 2$  bits and 0 bits—are determined by the largest meta-machine that is present at a given time. Being a closed set, the meta-machine does not allow any novel  $\epsilon$ -machines to survive, and this gives the upper bound on  $C_\mu(\mathcal{G})$ . As one  $\epsilon$ -machine type is removed from  $\Omega$  by the outflux, the meta-machine

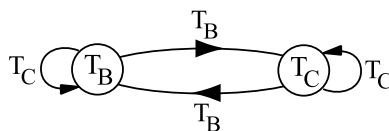


Figure 3. The meta-machine to which that in Figure 2 decays under the population dynamics of Equation 8.

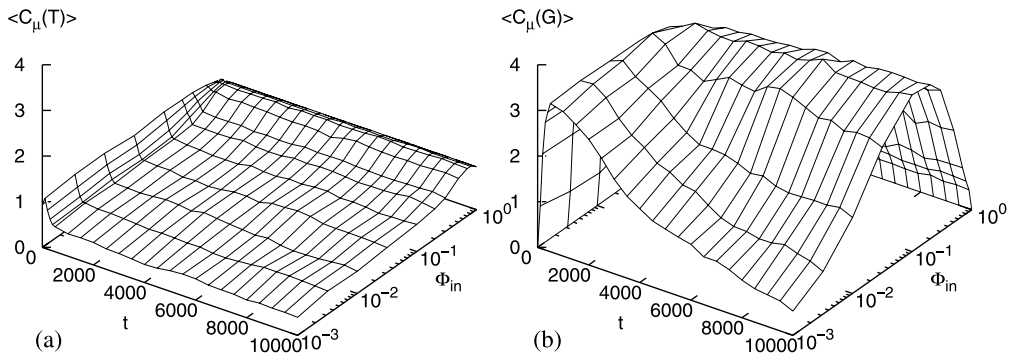


Figure 4. (a) Population-averaged  $\epsilon$ -machine complexity  $\langle C_{\mu}(T) \rangle$  and (b) run-averaged interaction network complexity  $\langle C_{\mu}(\mathcal{G}) \rangle$  as a function of time  $t$  and influx rate  $\Phi_{in}$  for a population of  $N = 100$  objects. (Reprinted with permission from [6].)

decomposes and the upper bound on  $C_{\mu}(\mathcal{G})$  lowers. This produces a stepwise and irreversible succession of meta-machine decompositions.

Thus, in the case of zero influx, one sees that the soup moves from one extreme to another. It is completely disordered initially, generates structural complexity in its individuals and in its interaction network, runs out of resources (poorly reproducing  $\epsilon$ -machines that are consumed by outflux), and decomposes down to a single type of simple self-reproducing  $\epsilon$ -machine.

Although Figure 5 shows only three plateaus, there is in principle one plateau for every meta-machine that at some point is the largest one generated by the soup. The diagram in Figure 6 summarizes our results from a more extensive and systematic survey of meta-machine hierarchies from a series of runs with  $N = 500$ . It gives one illuminating example of how the soup spontaneously generates hierarchies of meta-machines.

Leaving closed soups behind, we now investigate the effects of influx. Recall the population-averaged  $\epsilon$ -machine complexity  $\langle C_{\mu}(T) \rangle$  and the run-averaged interaction network complexity  $\langle C_{\mu}(\mathcal{G}) \rangle$

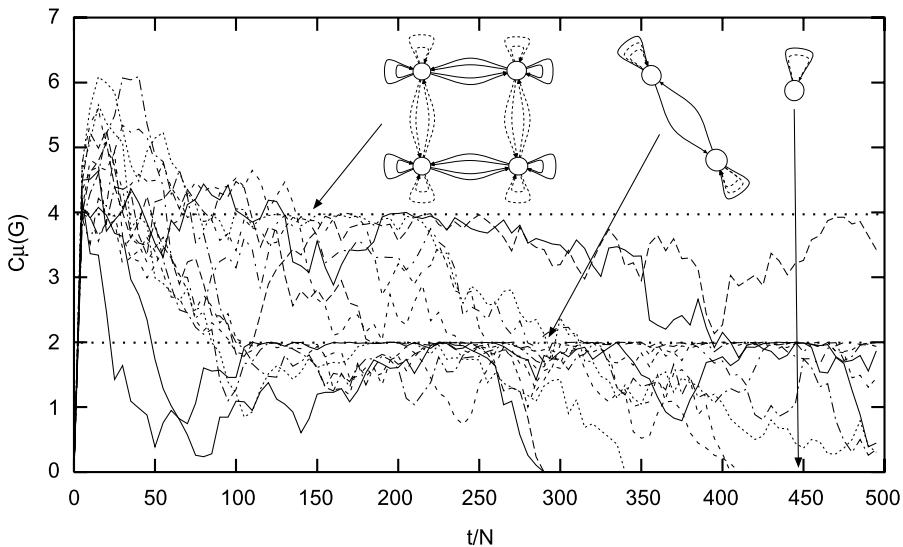


Figure 5. Meta-machine decomposition in a closed soup: 15 separate runs with  $N = 500$  that illustrate the typical behaviors. While the minimal four-element meta-machine  $\Omega_4$  (shown) dominates the soup,  $C_{\mu}(\mathcal{G})$  is bounded by 4 bits. Once outflux removes one of its  $\epsilon$ -machines,  $\Omega_4$  decays rapidly to  $\Omega_2$ , a two-element meta-machine (shown). ( $\Omega_4$  does not contain a sub-meta-machine of three  $\epsilon$ -machines.) At this point  $C_{\mu}(\mathcal{G})$  is bounded by 2 bits. After some period of time,  $\Omega_2$  decays to  $\Omega_1$ , a single self-reproducing  $\epsilon$  machine (shown), and  $C_{\mu}(\mathcal{G})$  is fixed at 0.



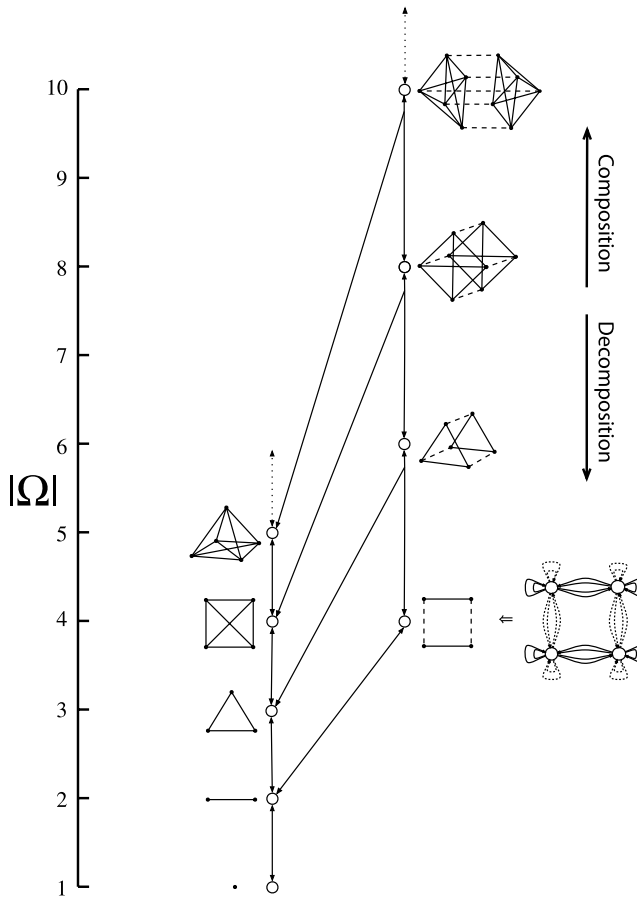


Figure 6. Meta-machine hierarchy of dynamical composition and decomposition. Dots denote  $\epsilon$ -machines. An isolated dot denotes a self-replicating  $\epsilon$ -machine. Solid lines denote  $T_A \xrightarrow{T_B} T_C$  transitions. Dashed lines denote  $T_B \xrightarrow{T_A} T_C$  transitions. Although all possible transitions are used by the meta-machines shown, they are represented in a simplified way according to  $\Omega_4$ ; cf. Figure 5.

as a function of  $t$  and  $\Phi_{in}$  shown in Figure 4. Over time,  $\langle C_\mu(T) \rangle$  behaves similarly for  $\Phi_{in} > 0$  to the way it does when  $\Phi_{in} = 0$ . It increases rapidly initially, reaches a peak, and declines to a steady state. Notably, the emergence of complex organizations of interaction networks occurs where the average structural complexity of the  $\epsilon$ -machines is low. Stationary  $\langle C_\mu(T) \rangle$  is instead maximized at a relatively high influx rate ( $\Phi_{in} \approx 0.75$ ), at which  $\langle C_\mu(\mathcal{G}) \rangle$  is small compared to its maximum. As  $\Phi_{in}$  is increased, so is  $\langle C_\mu(\mathcal{G}) \rangle$  at large times.  $\langle C_\mu(\mathcal{G}) \rangle$  is maximized around  $\Phi_{in} \approx 0.1$ . For higher influx rates, individual novelty has a deleterious effect on the sophistication of a population's interaction network. Existing reproductive paths do not persist due to the low rate of successful compositions of highly structured, and therefore specialized, individuals. We found that the maximum network complexity  $\widehat{C}_\mu(\mathcal{G})$  grows slowly and linearly over time at  $\approx 7.6 \times 10^{-4}$  bits/replication.

### 7 Summary and Conclusions

To understand the basic mechanisms driving the evolutionary emergence of structural complexity in a quantitative and tractable pre-biotic setting, we investigated a well-stirred soup of  $\epsilon$ -machines (finite-memory communication channels) that react with each other by composition and so generate new  $\epsilon$ -machines. When the soup is open with respect to composition and influx, it spontaneously builds structural complexity on the level of transformative relations among the  $\epsilon$ -machines rather



than in the  $\epsilon$ -machine individuals themselves. This growth is facilitated by the use of relatively non-complex individuals that represent general and elementary local functions rather than highly specialized individuals. The soup thus maintains local simplicity and generality in order to build up hierarchical structures that support global complexity.

Novel computational representations are intrinsically introduced in the form of meta-machines that, in turn, are interrelated in a hierarchy of composition and decomposition. Computationally powerful local representations are thus not necessary (nor effective) in order for the emergence and growth of complex replicative processes in the finitary process soup.

Meta-machines in closed soups eventually decay. For  $C_\mu(\mathcal{G})$  to maintain itself and grow, the soup must be fed with novel material in the form of random  $\epsilon$ -machines. Otherwise, any spontaneously generated meta-machines are decomposed (due to finite-population sampling), and the population eventually consists of a single type of trivially self-reproducing  $\epsilon$ -machine. At an intermediate influx rate, however, the interaction network complexity is not only maintained but grows linearly with time. This, then, suggests the possibility of open-ended evolution of increasingly sophisticated organizations.

### Acknowledgments

This work was supported at the Santa Fe Institute under the Network Dynamics Program funded by the Intel Corporation and under the Computation, Dynamics, and Inference Program via SFI's core grants from the National Science and MacArthur Foundations. Direct support was provided by NSF grants DMR9820816 and PHY9910217 and DARPA Agreement F306020020583. O.G. was partially funded by PACE (Programmable Artificial Cell Evolution), a European Integrated Project in the EU FP6-IST-FET Complex Systems Initiative.

### References

1. Adami, C., & Brown, C. T. (1994). Evolutionary learning in the 2D artificial life system 'Avida'. In *Artificial Life 4* (pp. 377–381). Cambridge, MA: MIT Press.
2. Bagley, R. J., Farmer, J. D., Kauffman, S. A., Packard, N. H., Perelson, A. S., & Stadnyk, I. M. (1989). Modeling adaptive biological systems. *Biosystems*, *23*, 113–138.
3. Brookshear, J. G. (1989). *Theory of computation: Formal languages, automata, and complexity*. Benjamin/Cummings.
4. Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley-Interscience.
5. Crutchfield, J. P. (1994). The calculi of emergence: Computation, dynamics, and induction. *Physica D*, *75*, 11–54.
6. Crutchfield, J. P., & Görnerup, O. (2006). Objects that make objects: The population dynamics of structural complexity. *Journal of the Royal Society Interface*, *3*, 345–349; doi:10.1098/rsif.2006.0114.
7. Crutchfield, J. P., & Mitchell, M. (1995). The evolution of emergent computation. *Proceedings of the National Academy of Sciences of the U.S.A.*, *92*, 10742–10746.
8. Crutchfield, J. P., & van Nimwegen, E. (2002). The evolutionary unfolding of complexity. In L. F. Landweber & E. Winfree (Eds.), *Evolution as computation* (pp. 67–94). New York: Springer-Verlag; Santa Fe Institute Working Paper 99-02-015; adap-org/9903001.
9. Crutchfield, J. P., & Young, K. (1989). Inferring statistical complexity. *Physical Review Letters*, *63*, 105–108.
10. Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D*, *2*, 187–204.
11. Fontana, W. (1991). Algorithmic chemistry. In C. Langton, C. Taylor, J. D. Farmer, & S. Rasmussen (Eds.), *Artificial Life II* (pp. 159–209). Reading, MA: Addison-Wesley.
12. Fontana, W., & Buss, L. W. (1996). The barrier of objects: From dynamical systems to bounded organizations. In S. Casti & A. Karlqvist (Eds.), *Boundaries and barriers* (pp. 56–116). Reading, MA: Addison-Wesley.
13. Hopcroft, J. E., & Ullman, J. D. (1979). *Introduction to automata theory, languages, and computation*. Reading, MA: Addison-Wesley.

14. Ikegami, T., & Hashimoto, T. (1997). Replication and diversity in machine-tape coevolutionary systems. In C. G. Langton & K. Shimohara (Eds.), *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems* (pp. 426–433). Cambridge, MA: MIT Press.
15. Lynch, M., & Conery, J. S. (2003). The origins of genome complexity. *Science*, *302*, 1401–1404.
16. Rasmussen, S., Knudsen, C., Feldberg, P., & Hindsholm, M. (1990). The coreworld: Emergence and evolution of cooperative structures in a computational chemistry. In S. Forrest (Ed.), *Emergent Computation* (pp. 111–134). Amsterdam: North-Holland.
17. Rasmussen, S., Knudsen, C., & Feldberg, R. (1992). Dynamics of programmable matter. In *Artificial Life II: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*. Reading, MA: Addison-Wesley.
18. Rat Genome Sequencing Project Consortium. (2004). Genome sequence of the brown Norway rat yields insights into mammalian evolution. *Nature*, *428*, 493–521.
19. Ray, T. S. (1991). An approach to the synthesis of life. In C. Langton, C. Taylor, J. D. Farmer, & S. Rasmussen (Eds.), *Artificial Life II* (pp. 371–408). Reading, MA: Addison-Wesley.
20. Rössler, O. (1979). Recursive evolution. *BioSystems*, *11*, 193–199.
21. Schrödinger, E. (1967). *What is life? Mind and matter*. Cambridge, UK: Cambridge University Press.
22. Schuster, P. (1977). The hypercycle: A principle of natural self-organization. *Naturwissenschaften*, *64*, 541–565.
23. Varela, F. J., Maturana, H. R., & Uribe, R. (1974). Autopoiesis: The organization of living systems. *BioSystems*, *5*(4), 187–196.
24. von Neumann, J. (1966). *Theory of self-reproducing automata*. Bloomington, IL: University of Illinois Press.