Complexity Scaling of a Minimal Functional Chemistry

Chris Salzberg
Department of General Systems Studies, Graduate School of Arts and Sciences
The University of Tokyo, 3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, Japan
chris@sacral.c.u-tokyo.ac.jp

Abstract

An abstract artificial chemistry is proposed in the form of a transformation on a space of directed graph structures. The transformation is derived by generalizing the reading process of a finite-state machine from transitions between *states* to transitions between *pairs of coupled states* (*composite states*). The state-transition graph generated by the transformation (*composite-state machine*) may be seen as a "phase space" of possible reading trajectories on the original graph. Such a phase-space is always exponentially larger than the original state-transition graph from which it is derived. Hierarchical *complexity scaling* is demonstrated, wherein complexity of the product of the transformation is shown to expand proportionately with the size of the graph (number of nodes). Simple examples evidence a richness of topological diversity and seemingly open-ended complexity increase. A classification scheme is introduced to group simple graphs on the basis of the topologies that they generate, uncovering certain basic trends. Relation of the findings to earlier artificial chemistry models and directions for future work are discussed.

Introduction

Within the fields of complex systems and artificial life, the search for simple systems capable of expressing open-ended increases in complexity has resulted in a number of computational models taking on a variety of forms. Differing types of complexity have been demonstrated in such models. The potential of certain cellular automata systems, for instance, to continuously generate complex temporal patterns is by now well-known (Ilachinski, 2001; Wolfram, 2002). Systems of nonlinear differential equations readily evidence complexity in their phase-space trajectories (Prigogine and Stengers, 1984). Many other systems, both discrete and continuous, exhibit other types of complexity (Bar-Yam, 1997).

The field of Artificial Chemistry (AC) also concerns itself with complexity, but in the context of uncovering basic mechanisms governing organization, self-maintenance, and self-construction (Dittrich et al., 2001). Here, complexity may for instance be observed in the construction pathways that sustain complex chemical networks. Among the most famous AC models exhibiting this type of complexity is the λ-calculus system proposed by Fontana and Buss (Fontana, 1991; Fontana and Buss, 1994) In their model, Fontana and Buss demonstrate the emergence of complex self-sustaining constructive networks made up of interacting λ-calculus expressions. Similar complexity has also been discovered in systems of co-constructing machines and tapes (Ikegami and Hashimoto, 1995). In each case, topological complexity of the network that emerges is not readily reducible to the components (expressions, machines/tapes) upon which it is defined. Many other AC models have been devised that deal with related forms of complexity and emergence (Banzhaf, 1994; Dittrich and Banzhaf, 1998; McCaskill, 1988; Thürk, 1993; Yamamoto and Kaneko, 2003).

The goal of the work presented in this paper is to: (a) to devise an abstract artificial chemistry that is *minimal* in terms of its formal specification, and (b) to explore the constructive potential of this minimal chemistry. A transformation on a space of directed graph structures is proposed satisfying (a), properties of which distinguish it from earlier AC and complex systems models. Whereas existing models focus mainly on dynamics within a fixed state-space, the proposed transformation applies directly to the state-space itself, generating a larger, more expansive, and more complex "composite" space. While this new space is similar to the constructive networks of earlier artificial chemistry models, a key distinction is that in this case the components and network are of the same type: both are state-transition graphs. The process of generating composite spaces (state-composition) may thus be applied indefinitely, generating a hierarchy of graphs of exponentially-increasing size and complexity. Examples of such hierarchies are investigated.

In the next section, the basic format of the system is outlined. Section III describes simple exploration experiments giving an indication of the system's complexity. A classification scheme is also introduced to group the set of minimal two-node graphs on the basis of the topologies that they generate. Conclusions are summarized in Section IV.

Formulation

The starting point for our formulation is the traditional concept of a finite-state machine (FSM) from computer science

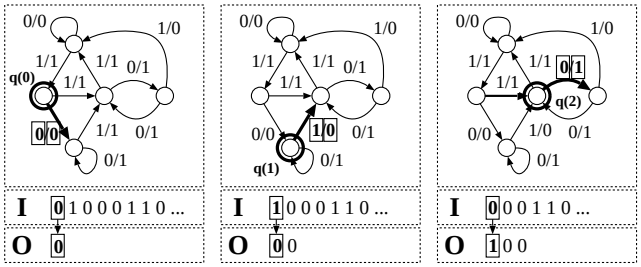

Figure 1: Three steps in a reading process on a finite-state machine. Input and output streams are included below state-transition graphs. Encircled node denotes the current state $q(t)$, thick arrow highlights the transition to the next state $q(t+1)$, and black squares mark the input value $i(t)$ and output value $o(t)$ of the transition. Output functions are included beside transitions in the form $i/\varphi_q(i)$, for an input $i \in \mathscr{A}$ from a node $q \in Q$.

theory (Minsky, 1967). A FSM is described in terms of a state-transition graph $G = (Q, \mathscr{A}, \delta, \varphi)$, where Q is a set of states, $\mathscr{A}$ is a finite alphabet, $\delta : \mathscr{A} \times Q \to Q$ is a transition function, and $\varphi : \mathscr{A} \times Q \to \mathscr{A}$ is an output function. We refer to transition and output functions for an input $i \in \mathscr{A}$ from a state $q \in Q$ in the form $\delta_q(i) \equiv \delta(i,q)$ and $\varphi_q(i) \equiv \varphi(i,q)$, respectively. The FSM constitutes a minimal abstraction of a physical machine: a collection of states, transitions between states, and elementary input/output mappings. Given an input $i \in \mathscr{A}$, a FSM in state $q \in Q$ follows the transition to state $\delta_q(i)$ and returns an output $\varphi_q(i)$.

For what follows, we employ a dynamic picture of a FSM, where $i(t) \in \mathscr{A}$ is an input stream, $q(t) \in Q$ is a dynamic state, and $o(t) \in \mathscr{A}$ is an output stream, defined at time step t. The next state is then defined as $q(t+1) := \delta_q(i)$, and the output as $o(t) := \varphi_q(i)$, with i and q short for $i(t)$ and $q(t)$, respectively. Fig. 1 illustrates three steps in a reading process of a simple FSM using this dynamic picture. The reader is encouraged to familiarize themselves with this process and with the associated notation before moving on.

We consider an extension of the traditional FSM, referred to as a *composite-state machine* (CSM) and defined in terms of a pair of states $s(t)$ (sender) and $r(t)$ (receiver) in Q. Output and transition functions at the state $s(t)$ for an input $i \in \mathscr{A}$ are denoted as $\varphi_s(i)$ and $\delta_s(i)$, respectively, for short; functions at the state $r(t)$ are denoted as $\varphi_r(i)$ and $\delta_r(i)$, respectively. The reader should think of the pair of states $s(t)$ and $r(t)$, referred to together as a *composite state*, as analogous to the dynamic state $q(t)$ of the FSM.

We define output and transition functions for the CSM again in analogy to the FSM. The output $o(t)$ of the CSM is derived by taking the value of the output function at $s(t)$ and feeding it to the output function at $r(t)$:

$$o(t) \quad = \quad \varphi_r(\varphi_s(i)). \tag{1}$$

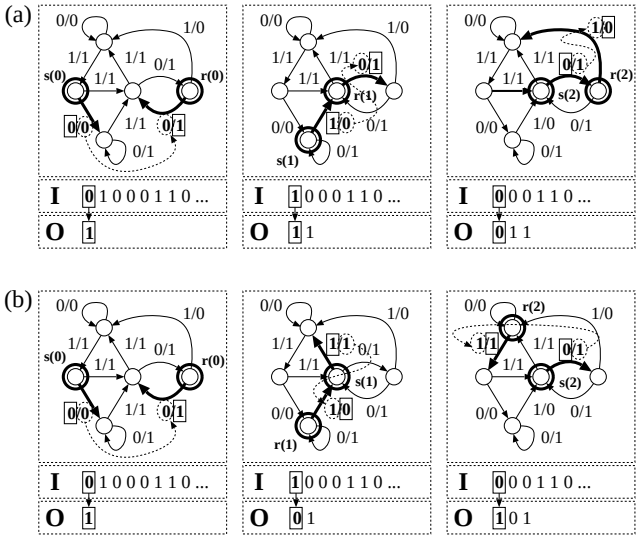

Figure 2: Three steps in a reading process on (a) unidirectional and (b) bidirectional CSM. Encircled nodes indicate the current states $s(t)$ and $r(t)$, thick arrows highlight transition to the next state, and black squares mark the input value $i(t)$ and output value $o(t)$ of the transitions.

A transition function on a CSM takes a state-pair $(s(t), r(t))$ and returns a next state-pair $(s(t+1), r(t+1))$. Two sets of transition functions are defined, each set describing a different type of CSM. The first set we refer to as *unidirectional*:

$$\begin{aligned} s(t+1) &= \delta_s(i), \\ r(t+1) &= \delta_r(\varphi_s(i)). \end{aligned} \tag{2}$$

where s, r, and i are short for $s(t)$, $r(t)$ and $i(t)$, respectively. A reading process of a unidirectional CSM is shown in Fig. 2(a). The term "unidirectional" is used to highlight the set of dependencies that are involved: the value of $s(t+1)$ is dependent on the input $i(t)$ and on $s(t)$, whereas the value of $r(t+1)$ is dependent on $i(t)$, $s(t)$ and $r(t)$. We may thus say that information "flows" from i, to s, to r.

We also define a *bidirectional* transition, identical to (2) except that assignments to $s(t+1)$ and $r(t+1)$ are swapped:

$$\begin{aligned} s(t+1) &= \delta_r(\varphi_s(i)), \\ r(t+1) &= \delta_s(i). \end{aligned} \tag{3}$$

A reading process of a bidirectional CSM is shown in Fig. 2(b). Note that in this case both dynamic states $s(t+1)$ and $r(t+1)$ are co-dependent: both are functions of earlier values of $i(t)$, $s(t)$, and $r(t)$. $r(t+1)$, for instance, is a function of $s(t)$ and $i(t)$, and $s(t)$ is a function of $r(t-1)$.

Given an initial state $(s(0), r(0))$, the combination of the output function (1) and either (2) (unidirectional) or (3) (bidirectional) transition functions describes subsequent dynamics of state-pairs on a composite-state machine. In Fig. 2

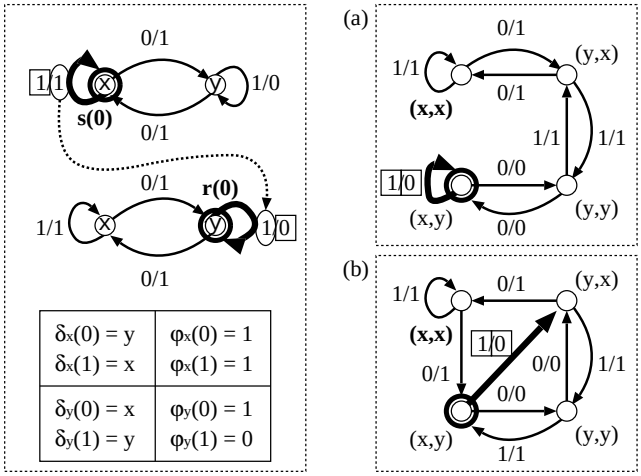

Figure 3: Composite state-transition diagrams for a simple two-node graph. Left: two diagrams of an identical two-node graph (top) and its tabular representation (bottom). Right: (a) unidirectional and (b) bidirectional state-compositions. The highlighted transition begins from the state $(s(0),r(0)) = (x,y)$ for an input $i(0) = 1$, and advances to a state (x,y) (unidirectional) and (y,x) (bidirectional), in both cases returning an output of $\varphi_y(\varphi_x(0)) = \varphi_y(1) = 0$.

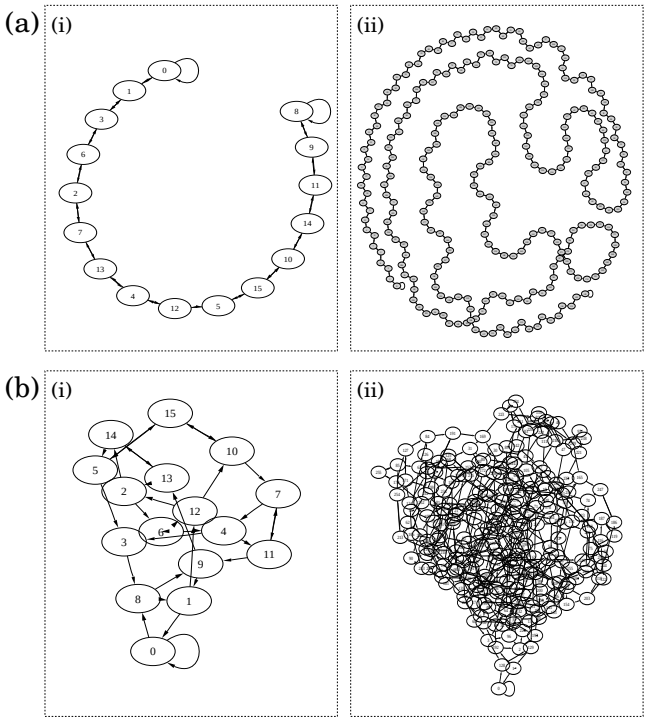

Figure 4: State compositions of Fig. 3(a) and Fig. 3(b). Node arrangement is calculated using a method proposed by Kamada and Kawai (Kamada and Kawai, 1989) (see Acknowledgments). Input/output values have been omitted to emphasize structure.

these dynamics are depicted as the motion of two state pointers (s and r) on a single shared machine. However, we may also think of these dynamics as the motion of a *single* state pointer on a larger diagram, one that contains states in $Q \times Q$ and transitions between them. The latter diagram may be thought of as an expanded "phase-space" containing all possible reading trajectories of state-pairs on the original graph.

This view of a CSM is shown in Fig. 3 for a simple two-node graph. On the left, two copies of the graph are shown. As an example, transitions and input/output values for an initial state $(s(0),r(0)) = (x,y)$ and an input $i(0) = 1$ are highlighted. On the right, these transitions are compacted to a single transition on a larger (composite) graph. From equation (1), the output of this single transition is determined as:

$$o(1) = \varphi_y(\varphi_x(1)) = \varphi_y(1) = 0. \tag{4}$$

In the case of a unidirectional CSM (a), the transition function is defined by the pair of assignments in equation (2):

$$\begin{aligned} s(1) &= \delta_x(1) = x, \\ r(1) &= \delta_y(\varphi_x(1)) = \delta_y(1) = y. \end{aligned} \tag{5}$$

whereas in the case of a bidirectional CSM, these assignments are exchanged (eq. (3)):

$$\begin{aligned} s(1) &= \delta_y(\varphi_x(1)) = \delta_y(1) = y, \\ r(1) &= \delta_x(1) = x. \end{aligned} \tag{6}$$

This difference in transition functions is reflected in the topologies of Fig. 3(a) and Fig. 3(b): in the former case, the 1-transition from (x,y) points to itself, whereas in the latter case it points to (y,x). The reader is encouraged to work through other combinations of inputs and initial positions to verify that the composite diagrams of Fig. 3(a,b) are correct.

We refer to the transformations shown in Fig. 3(a) and Fig. 3(b) as *unidirectional* and *bidirectional state-compositions*, denoted as $\mathcal{C}_u(G)$ and $\mathcal{C}_b(G)$, respectively, of the graph G. Since the product of these transformations is itself a state-transition graph, state-composition may be applied repeatedly, resulting in increasingly large graphs. This expansion procedure constitutes the basis of the minimal chemistry to be explored in this paper.

There is a fundamental distinction between $\mathcal{C}_u$ and $\mathcal{C}_b$, noted earlier in relation to equations (2) and (3). To appreciate the distinction, consider a trajectory $\{(s(0),r(0)),(s(1),r(1)),\ldots,(s(n),r(n))\}$ on both graphs. In the case of $\mathcal{C}_u(G)$, the state $s(t)$, for $t \leq n$, is dependent on earlier values of s and on the input, but not on values of r; information "flows" from s to r. This is very different from $\mathcal{C}_b(G)$, in which values of $s(t)$ and $r(t)$ are *co-dependent*, a codependency that arises from the "flipping" of s and r in equation (3). Thus to determine the value of $s(t)$, a previous value of r must be known, and conversely to know $r(t)$

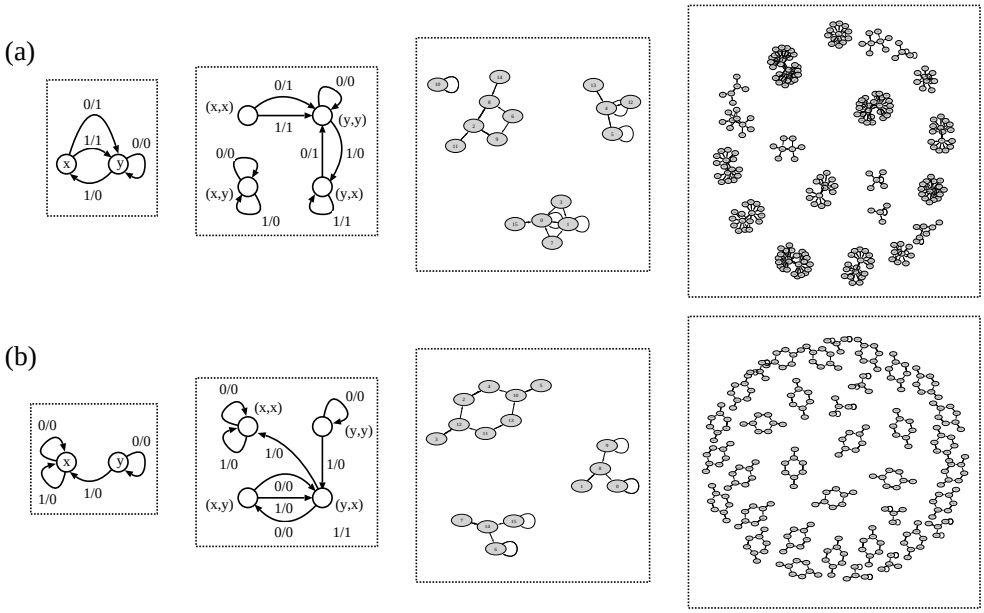

Figure 5: State compositions of some simple two-node graphs. (a) Open and (b) closed self-replicating systems. Input/output values in third and fourth frames have been omitted to emphasize structure.

a previous value of s must be known (in addition to earlier input values).

The distinction in dependencies between $\mathcal{C}_u$ and $\mathcal{C}_b$ is reflected in the structure of the state-composition graphs that they generate. In the case of $\mathcal{C}_u$, graphs are regular, with simple, repeated structures. Unidirectional state compositions of Fig. 3(a), shown in Fig. 4(a), demonstrate this regularity. In this particular case, state-composition produces an increasingly long chain structure of the same basic form as the four-node chain in Fig. 3(a). Complexity of the structure does not increase with application of $\mathcal{C}_u$. In the case of bidirectional state-composition, however, graphs may exhibit rapid complexity increase, as demonstrated in Fig. 4(b). Although the overall structure of the graph is difficult to discern from this plot, it is evident that novel topological features arise with each application of $\mathcal{C}_b$. This increase in complexity, which is roughly proportional to the increase in size of the graph, we refer to as *complexity scaling*.

For the reasons noted above, we employ the bidirectional state-composition transformation as the basis of our abstract chemistry. It is important to appreciate the minimality of the formalism underlying this transformation. In terms of mathematics, we have employed only two basic operations: function composition (in the output function (1)) and permutation (the "flipping" in equation (3)). This is in contrast to other popular formulations, such as for example cellular automata, which rely on a set of state-specific rules, specified externally, to define dynamics of the system. Analogous "rules" are difficult to isolate in the minimal chemistry we have proposed. Indeed, apart from the choice of unidirectional or bidirectional state-composition, arbitrariness is completely confined to the initial graph upon which the transformation is applied. This unique property underlies the rapid complexity increase shown in Fig. 4(b) and further explored in the following section.

Results

Given the space constraints of this paper, our aim in this section is simply to convince the reader, via a set of simple examples, of the complexity and diversity induced by state-composition; more detailed analyses are planned for a follow-up publication. To achieve this aim, we performed an exhaustive search of all bidirectional state-compositions of simple two-node graphs and grouped the results according to the types of topologies generated. There are a total of $2^8 = 256$ 2-node graphs, half of which may be ignored by fixing one output value of one link, bringing the total down to 128 graphs. First, second, and third state-compositions of each of these graphs were produced, generating a surprising diversity of topologies. A number of different classes of topologies were distinguished. Firstly, results were classified as either: (a) replicating systems or (b) complex networks. Category (a) constitutes cases in which state-composition graphs comprise multiple disjoint subgraphs, whereas (b) corresponds to cases in which state-composition graphs are fully connected. Fig. 4(b) is an example of a category (b) topology.

Fig. 5 depicts two examples of category (a) topologies. Two subclasses are shown here: Fig. 5(a) is an example of an *open* replicating system, whereas Fig. 5(b) is an example

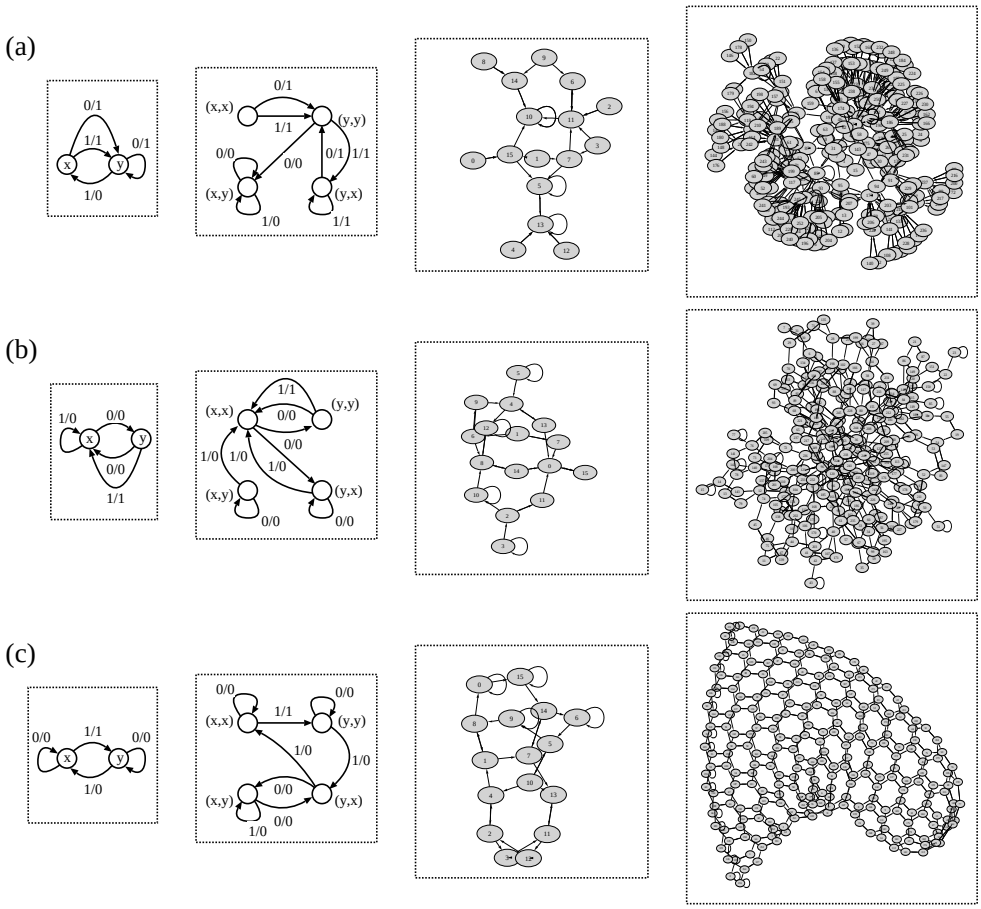

Figure 6: State compositions of some simple two-node graphs. (a) Complex network with multiple hubs. (b) Complex network with no central hub. (c) "Membrane" topology. Input/output values in third and fourth frames have been omitted to emphasize structure.

of a *closed* replicating system. In the first case, subgraphs increase in topological diversity and complexity in an open-ended manner. This series of graphs begins with a simple two node graph G, followed by two subgraphs in $\mathcal{C}_b(G)$, followed by four subgraphs in $\mathcal{C}_b(\mathcal{C}_b(G))$, and finally 21 subgraphs in $\mathcal{C}_b(\mathcal{C}_b(\mathcal{C}_b(G)))$. While evidencing a high degree of self-similarity, the majority of graphs in the last set are topologically distinct. Among the 128 2-node graphs, a handful of other examples of such open-ended complexity scaling have been observed, with distinct topologies.

Note that there is a combinatorial nature to this complexity increase: in generating a new graph, state-composition considers *all* possible pairings of nodes (states). This process may also be considered at the level of subgraphs: all pairings of subgraphs contribute to the state-composition. A new subgraph type at one level of the hierarchy may generate many new subgraph types at the next higher level of the hierarchy. These new types may themselves generate new types at next higher levels, and so on and so forth. This "cascading" underlies the complexity of Fig. 5(a).

In contrast, the diversity of subgraphs in Fig. 5(b) is limited to two basic types, a four-node graph and an eight-node graph. Here, the relations between each subgraph type (as dictated by state-composition) do not generate new types, thus the process of complexity scaling is limited. In our study of all two-node graphs, replicator systems with only a single subgraph type have also been observed; systems with more than two subgraph types have yet to be found.

In the category of complex networks, three basic topology classes have been observed in two-node graphs. The first class consists of networks with multiple "hub" nodes, each having numerous incoming links. Fig. 6(a) is an example of one such network. Other networks, such as the one as shown in Fig. 6(b), have no such central hub and are more decentralized. Finally, a third class of complex networks, very distinct from the first two, is shown in Fig. 6(c). We refer to this last class as a "membrane" topology, in reference to its three-dimensional structure. Two types of membranes have been observed, a "cone"-shaped topology (shown in Fig. 6(c)) and a "net"-shaped topology (not shown).

Although the above categories summarize the topologies observed, a great deal of diversity has been found even within each class. Note also that increasing the number of nodes in the initial graph vastly increases the diversity of topologies generated; we have focused here on the minimal set as a logical starting point for future work.

Conclusions

In this paper, we presented an abstract chemistry based on the concept of state-composition. Given the diversity of results obtained, and the minimality of the underlying formalism, we believe that this chemistry exemplifies a novel form of complexity not previously observed in other models. Systems such as cellular automata, difference equations, and dynamical systems emphasize rule-based (or law-based) dynamics in a fixed state-space. Complexity emerges in such models in the structure of state-space trajectories or in pattern formation. In the system we have presented it is organizational structure (graph topology), as opposed to dynamics, that is the focus of interest. The complexity that emerges from this structure does so as the space itself expands.

It may be argued that the chemistry we have presented is distinctly "unphysical" in the way it treats a computational phase-space as the product of a reaction. While one may imagine the composition process as analogous to the bonding of atoms or molecules, it is difficult coming from such a perspective to find any analogy for the cleavage of bonds or molecular decay; and yet the latter process is characteristic of the simplest living systems. This is a fair and valid criticism, and indeed the metaphor of a "chemistry" is quite loose. It may be more natural to think of the system as an organizational hierarchy of interacting parts or agents. Taking this perspective, the root of the complexity observed in Fig. 5 and Fig. 6 may be traced back to the interaction of many agents, each described by a simple state-transition graph. From this organizational viewpoint, this work relates very strongly to the studies by Fontana and Buss on interacting λ-calculus expressions, as well as to Rosen's MR-system (Rosen, 1971). The formalism presented here, however, is arguably simpler than the λ-calculus, and is certainly more constructive in nature than MR-systems.

One of the aspects of the system that has been emphasized in this paper is the scaling of complexity associated with the state-composition process. This complexity scaling, however, has not been rigorously defined in terms of a single "complexity metric". One goal for future work would be to experiment with different metrics with the aim of pinpointing in which instances (i.e. for which graphs) complexity "increases", and for which it does not. This would establish a more concrete basis for claims of complexity increase, as well as potentially provide a better understanding of the state-composition process.

A number of other pathways for future research also follow from the work presented here. At the level of the underlying formalism, it would be advantageous to establish connections with other computational models. A direct correspondence with certain cellular automata systems has already been found, to be discussed in a future publication. Secondly, a more extensive exploration of state-compositions of simple graphs is also planned. Finally, a long-term goal of this work is to find a match between the topologies of Fig. 5 and Fig. 6 and a real physical or organizational system. It remains an open question, however, whether such a match exists.

Acknowledgments

Research was supported by a scholarship from the Japanese Ministry of Education, Culture, Sports, Science and Technology. Graph visualization was performed using the software "GraphViz", which is based on a method proposed by Kamada and Kawai (Kamada and Kawai, 1989). Source code for generating state-composition graphs is available upon request from the author.

References

Banzhaf, W. (1994). Self-organisation in a system of binary strings. In Brooks, R. and Maes, P., editors, *Proceedings of Artificial life IV*, pages 109–118, Cambridge, MA. MIT Press.

Bar-Yam, Y. (1997). *Dynamics of Complex Systems*. Perseus Books Group, Cambridge, MA.

Dittrich, P. and Banzhaf, W. (1998). Self-evolution in a constructive binary string system. *Artificial Life*, 4:203–220.

Dittrich, P., Ziegler, J., and Banzhaf, W. (2001). Artificial chemistries — A review. *Artificial Life*, 7(3):225–275.

Fontana, W. (1991). Algorithmic chemistry. In Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S., editors, *Artificial Life II: SFI Studies in the Sciences of Complexity*, volume X. Addison-Wesley.

Fontana, W. and Buss, L. W. (1994). "The arrival of the fittest": Toward a theory of biological organization. *Bull. Math. Biol.*, 56:1–64.

Ikegami, T. and Hashimoto, T. (1995). Active mutation in self-reproducing networks of machines and tapes. *Artificial Life*, 2:305–318.

Ilachinski, A. (2001). *Cellular Automata: A Discrete Universe*. World Scientific Publishing Co. Pte. Ltd.

Kamada, T. and Kawai, S. (1989). An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15.

McCaskill, J. S. (1988). Polymer chemistry on a tape: A computational model for emergent genetics. Max-Planck-Institut für biophysikalische Chemie.

Minsky, M. (1967). *Computation: Finite and Infinite Machines*. Prentice-Hall, New Jersey.

Prigogine, I. and Stengers, I. (1984). *Order Out of Chaos: Man's new dialogue with nature*. Bantam Books, Toronto.

Rosen, R. (1971). Some realizations of (M,R)-systems and their interpretation. *Bull. Math. Biophys.*, 33:303–319.

Thürk, M. (1993). *Ein Modell zur Selbstorganisation von Automatenalgorithmen zum Studium molekularer Evolution*. PhD thesis, Friedrich-Schiller-Univerität, Jena.

Wolfram, S. (2002). *A New Kind of Science*. Wolfram Media.

Yamamoto, T. and Kaneko, K. (2003). Tile automaton in the well-mixed medium. *Physica D*, 181:252–273.