

Generative Network Automata: A Generalized Framework for Modeling Complex Dynamical Systems with Autonomously Varying Topologies

Hiroki Sayama

Department of Bioengineering, Binghamton University, State University of New York
P.O. Box 6000, Binghamton, NY 13902-6000 sayama@binghamton.edu

Abstract—We propose a new modeling framework “Generative Network Automata (GNA)” that can uniformly describe both state transitions and autonomous topology transformations of complex dynamical networks. GNA is formulated as an extension of existing complex dynamical network models to include a new set of generative update rules that determine how local network topologies will change based on the current local network states and topologies. This paper introduces basic concepts of GNA, its formal definitions, its generality to represent other dynamical systems models, and some preliminary results of an exhaustive sweep of possible dynamics found in elementary binary GNA with restricted updating rules.

I. INTRODUCTION

A variety of modeling tools have been proposed and utilized in complex systems research [1], [2]. Typical tools include equation-based dynamical systems such as ordinary or partial differential equations and iterative maps [3], and dynamical networks such as artificial neural networks [4], [5], random Boolean networks [6], [7], [8] and cellular automata [9], [10]. While they are capable of producing strikingly complex and even biological-like behaviors [11], [12], [13], [14], [15], these tools commonly assume a system made of a fixed number of components, or variables. Their dynamics are considered as trajectories of system states in a confined phase space with time-invariant dimensions.

The rapidly growing network theory has demonstrated yet another approach of complex systems modeling [16], [17]. It has provided a way to consider dynamic addition, modification and removal of components and their interactions (i.e., nodes and edges) and their implications for system-level properties. Among the most actively investigated issues in this field is how statistical properties of network topology will be affected by random and/or preferential addition (growth) and removal (errors or attacks) of nodes and edges, and in particular, how networks can be more robust against the latter [18], [19], [20], [21], [22]. Component addition and removal are typically assumed as perturbations coming from external sources, not incorporated into the dynamics of the system itself.

Researchers have recently started investigating dynamical properties of networks with nontrivial complex topologies [23], [24], [25], [26], [27], [28]. These studies are the forerunners that attempt to merge the above two fields, considering how states of networks with complex topologies will evolve over time and respond to exogenous perturbations, though they are still limited to networks with fixed topologies so far. Understanding the coupling between dynamic changes of states and

This work is supported in part by the Binghamton University Individual Development Award, UUP.

topologies of networks is now recognized as one of the most important problems to address [19].

When looking into real-world complex systems, one can find many instances of dynamical networks whose topology (or phase space dimensions and constitutions) keeps changing due to the system’s own dynamical rules (see Table I). In all of these systems, state transitions of each component and topological transformations of networks are strongly intertwined with each other. Neither conventional dynamical systems theory nor modern network theory is sufficient to fully describe the dynamics of such systems. There is a significant need for comprehensive models of systems with dynamically varying topologies in many cutting-edge fields, including system biology, human social dynamics, and communication infrastructure design.

We aim to address the above-mentioned lack of fundamental theories for modeling complex systems by extending existing complex dynamical network models to include a new set of generative update rules that determine how local network topologies will change based on the current local network states and topologies. We call this model framework “*generative network automata (GNA)*.” The name indicates the integration of knowledge accumulated in dynamical systems theory, network theory, and graph grammar theory.

In the following sections we will introduce basic concepts of GNA, its formal definitions, its generality to represent other dynamical systems models, and some preliminary results of an exhaustive sweep of possible dynamics found in elementary binary GNA with restricted updating rules.

II. BACKGROUND: GENERATIVE GRAPH GRAMMARS

The key characteristic of GNA is that it should have rules for transformations of local network topologies as well as transitions of local states. Such network transformations may be modeled as a rewriting process of local network configurations. We will therefore introduce methods and techniques developed in graph grammar theory [29], [30], [31], [32], [33] to construct general formulations of GNA.

Graph grammars, studied since late 1960’s in theoretical computer science, are an extension of formal grammars in computational linguistics to discuss similar rule-based rewriting processes of graphs, or networks. They are used to model generative processes of networks through repeated node and/or (hyper-)edge replacements. “Generative” means that these replacements are driven by local topological features of the network itself, and not by external sources of perturbation as typi-

TABLE I
REAL-WORLD EXAMPLES OF DYNAMICAL NETWORKS WHOSE TOPOLOGY CHANGES DUE TO THE SYSTEM'S OWN DYNAMICAL RULES.

System	Nodes	Edges	Example of states of nodes	Example of phase space changes
Organism	Cells	Intercellular communication channels	Gene/protein activities	Fission and death of cells during the developmental period
Ecological community	Species	Interspecific relationships	Population	Introduction and extinction of species
Human society	Individual	Social relationships	Social, economical, political and/or cultural statuses	Entry and withdrawal of individuals
Communication network	Terminals, hubs	Cables, wireless connections	Information stored and transacted	Addition and removal of terminal or hub nodes

cally assumed in modern network theory.

A classic, and probably most widely known, example of graph grammars is the Lindenmayer system, or L-system [34]. It is a simple rewriting system that can produce self-similar recursive structures in a sequential string (in this sense, the L-system remains within the range of classic formal grammars). What makes this system outstanding is that it comes with an interpretation algorithm that converts a resultant string into a tree-like topological structure, which may appear just like a natural tree if parameters are appropriately chosen. This example shows the capability of graph grammars to describe the emergence of natural complex structures using a set of small local rules.

Although their relevance to biology was initially recognized [29], [35], application of graph grammars has so far mostly remained within the range of computer science, such as pattern recognition, compiler design, and data type specification, with a primary focus on context-free rewriting rules [30], [31], [32], [33]. It is also limited in that it does not consider dynamics of network states. There has been some exceptional work done in this field though [36], [37], where dynamical networks were applied to solve graph-theoretic problems such as diameter measurement and closure detection. However, these studies did not consider generative aspects of networks.

Most recently, a few models have been developed for generative processes of networks with some dynamical aspects included [38], [39], [40]. These models are fairly complicated and heuristically designed for specific problems, with neither clean-cut theoretical foundations nor broad applicability.

To the best of our knowledge, our GNA framework is the first to systematically integrate rewriting rules for transformations of local network topologies into dynamical networks. Our aim is to develop generalized theory of GNA that can be broadly applied to the modeling of various complex systems.

III. DEFINITIONS OF GNA

A. Configuration

GNA is a network made of dynamical nodes and directional edges between them. Undirectional edges can also be represented by symmetrically placing two directional edges between nodes. Each node takes one of the (finitely or infinitely many) possible states defined by a state set S . The edges describe

referential relationships between the nodes, specifying how the nodes affect each other in state transition and topology transformation. A configuration of GNA at a specific time t is a combination of states and topologies of the network, which can be given by the following:

- V_t : A finite set of nodes. This determines the phase space dimensions of the system at time t . While usually assumed as time-invariant in conventional dynamical systems theory, this set can dynamically change in the GNA framework due to addition and removal of nodes.
- $C_t : V_t \rightarrow S$: A map from the node set to the local state set. This describes the global state of the system at time t . If local states are scalar numbers, C_t can be represented as a vector with its size potentially varying over time.
- $L_t : V_t \rightarrow V_t^*$: A map from the node set to its own closure. This specifies a set of destinations of outgoing edges originating from a specific node, representing the global topology of the system at time t , which is also potentially varying over time.

B. Dynamics

States and topologies of GNA are updated through a repetitive rewriting process that includes the following three steps:

1. Extraction of part of the GNA (subGNA) that will be subject to change.
2. Production of a new subGNA that will replace the subGNA selected above.
3. Embedding of the new subGNA into the rest of the whole GNA.

The temporal dynamics of GNA can therefore be formally defined by the following triplet $\langle E, R, I \rangle$:

- E : An extraction mechanism that determines which part of the GNA is selected for the updating. It may be implemented as a function that takes the whole GNA configuration and returns a subGNA configuration to be replaced. It may be deterministic or stochastic.
- R : A replacement mechanism that produces a new subGNA from the subGNA selected by E and also specifies the correspondence of nodes between the old and new subGNAs. It may be implemented as a function that takes a subGNA configuration and returns a pair of a new subGNA configuration and a mapping between nodes in the

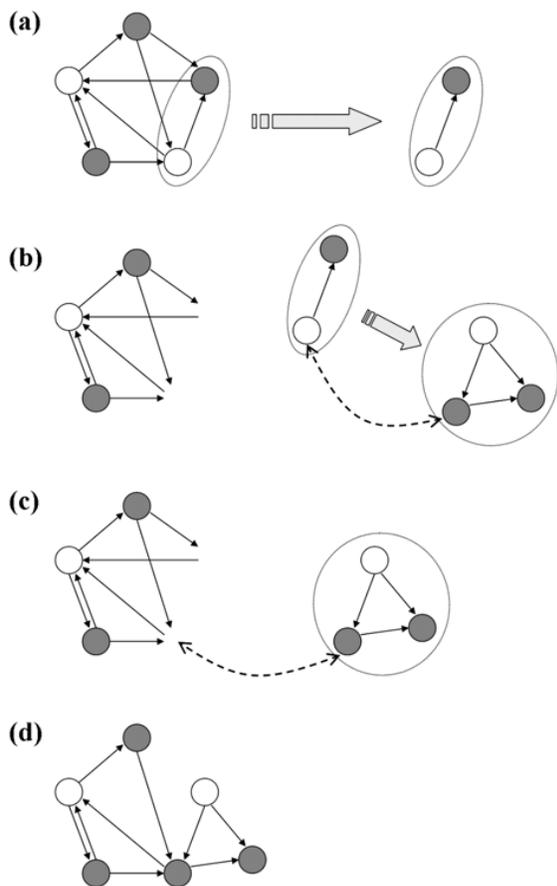


Fig. 1. GNA rewriting process. (a) The extraction mechanism E selects part of the GNA. (b) The replacement mechanism R produces a new subGNA as a replacement of the old subGNA and also specifies the correspondence of nodes between old and new subGNAs (dashed line). This process may involve both state transition of nodes and transformation of local topologies. The “bridge” edges that used to exist between the old subGNA and the rest of the GNA remain unconnected and open. (c) The new subGNA produced by R is embedded into the rest of the GNA according to the node correspondence also specified by R . In this particular example, the top gray node in the old subGNA has no corresponding node in the new subGNA, so the bridge edges that were connected to that node will be removed. (d) The updated configuration after this rewriting event.

old subGNA and nodes in the new subGNA. It may be deterministic or stochastic.

- I : An initial configuration of GNA.

To fully implement the algorithm for simulating GNA dynamics, one needs a couple more commonly used procedures such as the removal of the selected subGNA from the whole GNA and the re-connection of “bridge” edges (edges that were between the old subGNA and the rest of the GNA) when embedding the new subGNA. Because the workings of these procedures are fairly obvious, we omit detailed explanations for them. The above E, R, I are sufficient to define specific GNA models. The entire rewriting process is illustrated in Fig. 1, which visually shows how these mechanisms work together.

Note that this rewriting process, in general, may not be applied synchronously to all nodes or subGNAs in a network, be-

cause simultaneous modifications of local network topologies at more than one places may cause conflicting results that are inconsistent with each other. This limitation will not apply though when there is no possibility of such topological conflicts, e.g., when the rewriting rules are all context-free, or when GNA is used to simulate conventional dynamical networks that involve no topological changes.

Although the definitions given above could be one of the simplest possible formulations of GNA, it already has considerable complexity compared to conventional dynamical systems models. The possibility of temporal changes of V_t and L_t particularly makes it difficult to investigate its dynamical properties analytically. While defying analytical treatments, however, the updating process of GNA is algorithmically described and hence their dynamics can be easily experimented through computer simulation. We developed a preliminary GNA simulation/visualization program in Wolfram Research Mathematica for this purpose¹. The results presented in this paper are obtained using this simulator program.

IV. GENERALITY OF GNA

The GNA framework is highly general and flexible so that many existing dynamical network models can be represented and simulated within this framework.

For example, if R always conserves local network topologies and modifies states of nodes only, then the resulting GNA is a conventional dynamical network model, including cellular automata, neural networks, and random Boolean networks (Fig. 2 (a), (b)). A straightforward application of GNA typically comes with asynchronous updating schemes, as introduced in the previous section. Since asynchronous automata networks can emulate any synchronous automata networks [41], the GNA framework covers the whole class of dynamics that can be produced by conventional dynamical network models. Moreover, as mentioned earlier, synchronous updating schemes could also be implemented in GNA for this particular class of models, because they involve no topological transformation.

On the other hand, if R causes no change in local states of nodes, the resulting GNA is a purely generative model based on traditional graph grammars, which may also be a network growth model in modern network theory if appropriate assumptions are implemented in the subGNA extraction mechanism E (Fig. 2 (c)).

V. COMPUTATIONAL EXPLORATION OF POSSIBLE DYNAMICS OF SIMPLE GNA

We conducted computational exploration of potential dynamics found in a restricted subset of simple GNA models, which is intended to play a key priming role in our ongoing GNA research similar to that of the work by Wolfram [9] in cellular automata research. The preliminary results are summarized below.

¹ The Mathematica-based simulation program is still under active development but may be available upon request.

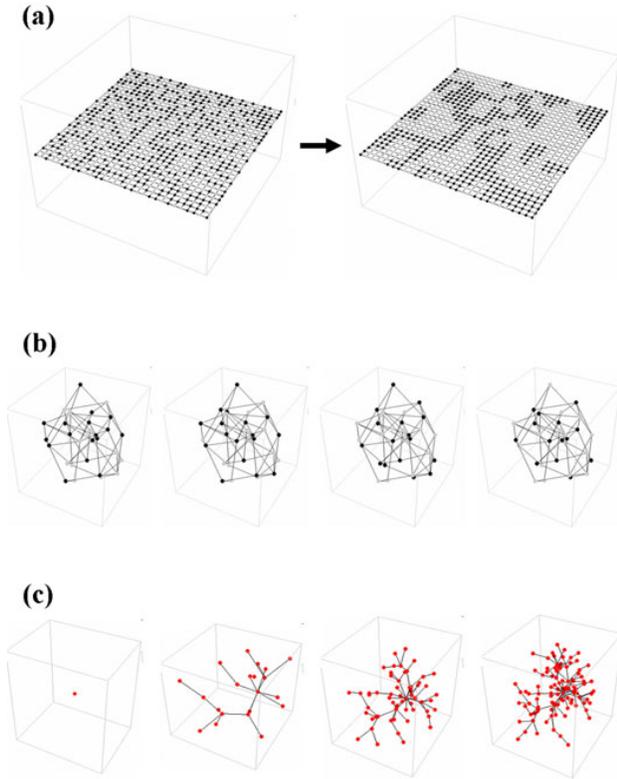


Fig. 2. Various dynamical network models simulated using GNA. These examples were represented in the same format of $\langle E, R, I \rangle$ (see text) and simulated using the same simulator program. (a) Simulation of asynchronous 2-D binary cellular automata with von Neumann neighborhoods and local majority rules. Space size: 100×100 . (b) Simulation of an asynchronous random Boolean network with $N = 30$ and $K = 2$. Time flows from left to right. Nodes of random Boolean networks are generally inhomogeneous, i.e., they obey different state-transition rules. Here each node's own state-transition rule is embedded as part of its state, and the replacement mechanism R refers to that information when calculating the next state of a node. (c) Simulation of a network growth model with the Albert-Barabási preferential attachment scheme. Time flows from left to right. Each new node is attached to the network with one edge. The extraction mechanism E is implemented so that it determines the place of attachment preferentially based on the node degrees, which causes the formation of a scale-free network in the long run.

A. Assumptions

We first need to note that there are infinitely many possible mechanisms for E and R because there are no theoretical upperbounds in terms of the size of the old subGNA selected by E (it could be infinitely large as the GNA grows) and the new subGNA produced by R (it could be arbitrarily large by the design of R). Making reasonable assumptions to restrict the possibility of mechanisms for E and R is critical to facilitate systematic study on the dynamics of GNA. Here we make the following assumptions (Fig. 3):

1. States are binary (0 or 1).
2. Edges are undirectional (or more precisely, every connection between nodes is represented by two symmetric directional edges).
3. The extraction mechanism E always selects a subGNA by (a) randomly picking one node u from the entire GNA (Fig.

3(a)),

(b) taking all the destination nodes of its outgoing edges $L_t(u)$ (Fig. 3(b)), and

(c) producing a subGNA “induced” by these nodes $\{u\} \cup L_t(u)$, i.e., a subGNA that includes all these nodes as well as all the edges present between them (Fig. 3(c)).

4. The replacement mechanism R only refers to the state of the central node in the selected subGNA and the local majority state within it. If there are equal numbers of 0's and 1's within the selected subGNA, one of them is randomly chosen. This two-bit information will be used to determine what will happen to the local configuration (Fig. 3(d)). The following seven possible outcomes are made available:

(0) The central node disappears.

(1) Everything remains in the same condition.

(2) The state of the central node is inverted.

(3) The central node divides into two with the state preserved.

(4) The central node divides into two with the state inverted.

(5) The central node divides into three with the state preserved.

(6) The central node divides into three with the state inverted.

(7) The initial condition I consists of a single node with state 0.

In cases where node division occurs, the edges that were connected to the central node is evenly distributed to its daughter nodes (Fig. 3(e)).

B. Methods

We carried out an exhaustive sweep of all the possible rewriting rules that satisfy the assumptions discussed above. Since the extraction mechanism E is uniquely defined, it is only the replacement mechanism R that needs to be varied. Here R is a function that maps one of the seven possible actions to each of the two-bit input patterns. Therefore the number of all the possible R 's is $7^2 = 2401$. To indicate a specific R , we will use its “Rule Number” $rn(R)$ that is defined by

$$rn(R) = a_{11} \times 7^3 + a_{10} \times 7^2 + a_{01} \times 7^1 + a_{00} \times 7^0, \quad (1)$$

where a_{ij} is a numerical representation (numbers shown in *italic* in the previous section) of the choice that R will make when the state of the central node is i and the local majority state is j .

We simulated the GNA dynamics using asynchronous updating scheme for 200 rewriting events for each rn ranging from 0 to 2400. In each simulation run we recorded the temporal changes of the number of nodes N_t and the average node degree K_t once in every half unit of time. We then calculated the average growth rate g and the temporal average of the average node degree k as characteristic quantities of each rn . g was obtained by averaging instantaneous growth rates $(N_{t+0.5}/N_t)^2$ over the entire time series data. k was obtained by averaging K_t over the second half of the time series data. The first half was omitted to avoid the effect of initial large fluctuations due to the small number of nodes. For each rn three independent simulation runs were conducted and the average of their results were used for the following analysis.

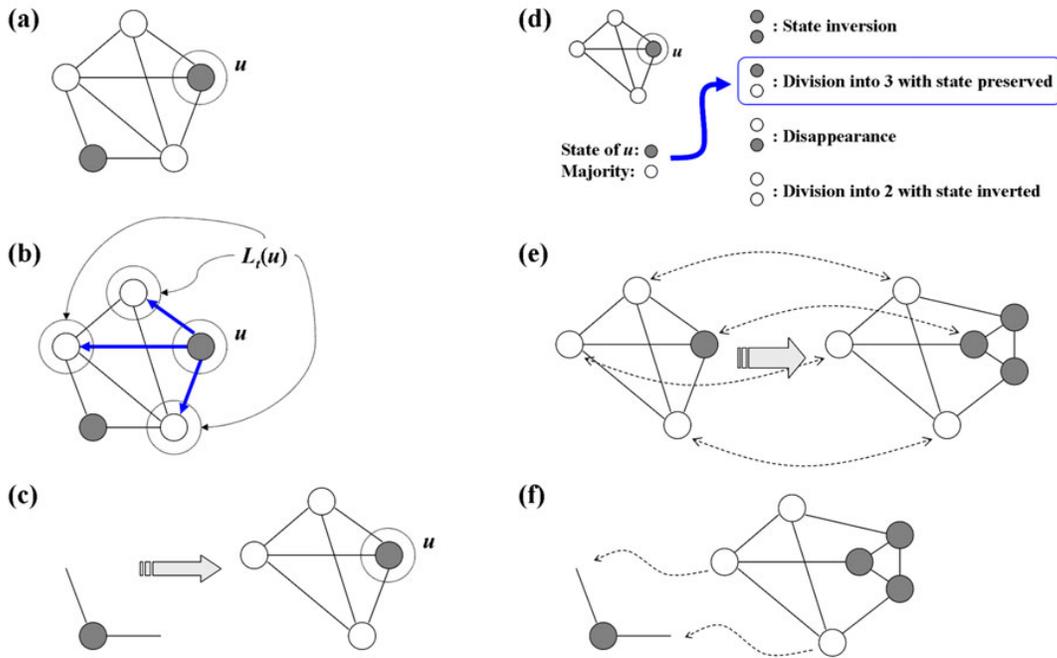


Fig. 3. Simplified GNA rewriting process used for the exhaustive sweep experiments. The extraction mechanism E (a) randomly picks one node u , (b) takes all the destination nodes of its outgoing edges $L_t(u)$, and (c) produces a subGNA induced by those nodes $\{u\} \cup L_t(u)$. The replacement mechanism R (d) refers to the state of the central node u in the selected subGNA and the local majority state within it to determine what happens to the local configuration, (e) produces a new subGNA as well as the correspondence of nodes between the old and new subGNAs based on the choice made in (d), and then (f) embeds the new subGNA into the rest of the GNA.

It should be noted that there are two different ways of counting time steps in asynchronous simulations. One is simply to count one rewriting event as one time step, which we call *computational time steps*. The other is to measure the progress of virtual time in a simulated world between each discrete events by considering one rewriting event as taking $1/N_t$ of the unit of time. This is based on the assumption that every node is updated once, on average, per unit of time, which is a reasonable and useful assumption especially when one wants to compare results of asynchronous simulations with those of synchronous ones. We call the latter notion of time *simulated time steps*. All the t 's in this paper represent this simulated time steps.

C. Results

Figure 4 shows the distribution of characteristic quantities of all the 2401 simple GNAs in the g - k space, where the large diversity of possible dynamics is demonstrated. The growth rate g is positively correlated with the average node degree k , because using divisions into three nodes will both speed up the growth of GNA and increase the node degrees. The distribution becomes broader and more sparse for smaller g , and ends at $g = 1$ which is the necessary growth rate to maintain the structure. Below this value is the regime where GNAs become extinct quickly, hence no positive k values are calculated. It is also visible that there is a dense horizontal cluster around the area of $1 < g < 3$ and $k = 1.9$. It was found that the generated GNAs were typically string-shaped in this area, where most nodes had two edges, resulting in k close to 2. Several samples of ac-

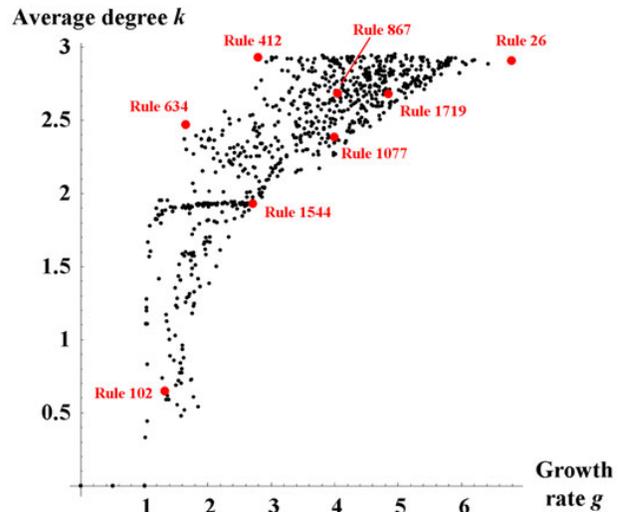


Fig. 4. Distribution of characteristic quantities (average growth rate g and average node degree k) of all the 2401 simple GNAs plotted in a g - k space. Sample cases shown in Fig. 5 are indicated with large dots.

tual network topologies and their temporal changes are shown in Fig. 5, which confirms topological diversity generated by the GNA framework.

The difference of dynamics between different rules can also be seen in the time series plot of the node number N_t (Fig. 6),

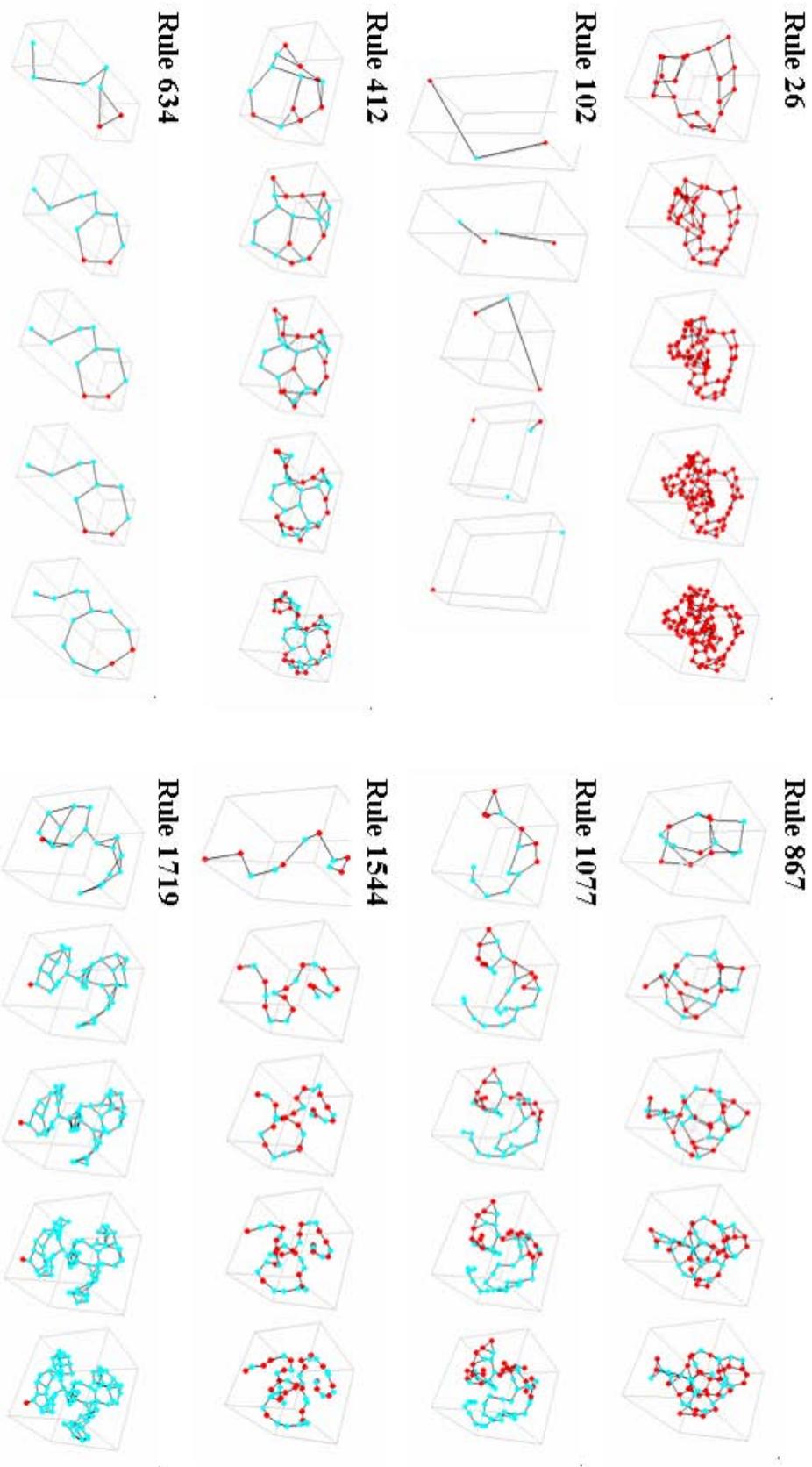


Fig. 5. Several samples of actual network growth processes taken from the distribution in Fig. 4. Time flows from left to right in all cases. Dark gray (red) dots represent nodes in state 0, while light gray (blue) ones in state 1. Some GNAs develop complex three-dimensional structure, while others remain almost in a linear structure (e.g., Rule 1544).

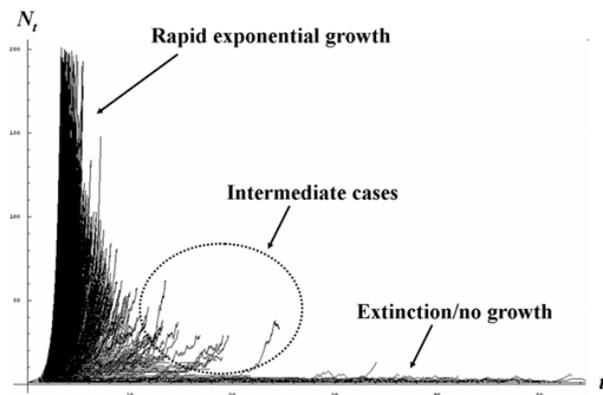


Fig. 6. Growth curves of all the 2401 simple GNAs superposed. Curves that go along the t axis indicate that the GNA for these cases did not grow. Many other rules showed rapid exponential growth processes. Between these two, there are relatively fewer intermediate cases that exhibit slow and fluctuating growth.

where the rules may be roughly classified into (1) extinction or no growth, (2) rapid exponential growth, and (3) intermediate cases that exhibit slow and fluctuating growth. The last cases may be particularly interesting, because their growth processes could be sensitive to local conditions, sitting in the critical regime at the boundary between two different dynamics, or possibly regulated by some intricate mechanisms to slow down.

Figure 7 shows temporal changes of network topologies found in this intermediate cases. Their behaviors are characterized by relatively slow growth and occasional production of separated network fragments due to a disappearance of nodes, or a nonhomogeneous development of a linear GNA strand. The former may be viewed as a precursory process of biological replication. The latter may be qualitatively similar to Wolfram's Class IV dynamics, here taking place in expanding 1-D CA arising on a GNA. These remain no more than a speculation at this time, and they will need further investigation.

VI. CONCLUSION

We proposed Generative Network Automata as a new generalized modeling framework that allows us to uniformly describe the dynamics of various complex systems that have been handled with different modeling techniques so far. We hope that it will help formulate many distinct complex systems in the same "format", enabling one to compare those systems systematically, to identify their commonness and uniqueness, and to actively exchange knowledge between different fields. The preliminary results obtained in the exhaustive sweep of simple binary GNA dynamics revealed a large diversity of possible outcomes that may be modeled and simulated using this framework.

As our research on GNA is still at its launching stage, there are many things to be done in the near future. The effectiveness of GNA must be evaluated through its applications to the modeling of real biological, social, and/or engineered complex systems. Specific areas of application we are currently planning to include are the modeling of decentralized control mechanisms of multicellular organisms, team development and deci-

sion making processes in complex organizations, and dynamical hierarchies in evolutionary systems.

One problem that needs to be solved immediately is the computational inefficiency of our current GNA simulator. To resolve this problem we plan to develop a specialized modeling and simulation platform software that runs as a native application rather than a script implemented on another software. This new platform will be used for interactive modeling, analysis and visualization of results, as well as computationally efficient large-scale simulations of GNA. It would provide substantial horsepower for GNA research, similar to *DDLab* for dynamical systems research [42].

REFERENCES

- [1] Bar-Yam, Y. (1997) *Dynamics of Complex Systems*. Westview Press.
- [2] Boccaro, N. (2004) *Modeling Complex Systems*. Springer-Verlag.
- [3] Strogatz, S. H. (1994) *Nonlinear Dynamics and Chaos*. Westview Press.
- [4] McCulloch, W. S. & Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* 5:115-133.
- [5] Hopfield, J. J. (1982) Neural networks and physical systems with emergent collective computational abilities. *PNAS* 79:2554-2558.
- [6] Kauffman, S. A. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* 22:437-467.
- [7] Derrida, B. & Pomeau, Y. (1986) Random networks of automata: A simple annealed approximation. *Europhys. Lett.* 1:45-49.
- [8] Kauffman, S. A. (1993) *The Origins of Order*. Oxford University Press.
- [9] Wolfram, S. (1984) Universality and complexity in cellular automata. *Physica D* 10:1-35.
- [10] Ilachinski, A. (2001) *Cellular Automata: A Discrete Universe*. World Scientific.
- [11] May, R. M. (1976) Simple mathematical models with very complicated dynamics. *Nature* 261:459-467.
- [12] Berlekamp, E. R., Conway, J. H., & Guy, R. K. (1982) *Winning Ways for Your Mathematical Plays Vol. 2: Games in Particular*, Chapter 25: "What is Life?". Academic Press.
- [13] Pearson, J. E. (1993) Complex patterns in a simple system. *Science* 261:189-192.
- [14] Sayama, H. (1999) A new structurally dissolvable self-reproducing loop evolving in a simple cellular automata space. *Artificial Life* 5:343-365.
- [15] Salzberg, C. & Sayama, H. (2004) Complex genetic evolution of artificial self-replicators in cellular automata. *Complexity* 10(2): 33-39.
- [16] Watts, D. J. & Strogatz, S. H. (1998) Collective dynamics of 'small-world' networks. *Nature* 393:440-442.
- [17] Strogatz, S. H. (2001) Exploring complex networks. *Nature* 410:268-276.
- [18] Albert, R., Jeong, H. & Barabási, A.-L. (2000) Error and attack tolerance of complex networks. *Nature* 406:378-382.
- [19] Albert, R. & Barabási, A.-L. (2002) Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74:47-97.
- [20] Shargel, B., Sayama, H., Epstein, I. R. & Bar-Yam, Y. (2003) Optimization of robustness and connectivity in complex networks. *Phys. Rev. Lett.* 90:068701.
- [21] da Fontoura Costa, L. (2004) Reinforcing the resilience of complex networks. *Phys. Rev. E* 69:066127.
- [22] Beygelzimer, A., Grinstein, G. M., Linsker, R. & Rish, I. (2005) Improving network robustness by edge modification. *Physica A* 357:593-612.
- [23] Aldana, M. & Cluzel, P. (2003) A natural class of robust networks. *PNAS* 100:8710-8714.
- [24] Bar-Yam, Y. & Epstein, I. R. (2004) Response of complex networks to stimuli. *PNAS* 101:4341-4345.
- [25] Hütt, M.-Th. & Lüttge, U. (2004) The interplay of synchronization and fluctuations reveals connectivity levels in networks of nonlinear oscillators. *Physica A* 350:207-226.
- [26] Motter, A. E. (2004) Cascade control and defense in complex networks. *Phys. Rev. E* 93:098701.
- [27] de Aguiar, M. A. M. & Bar-Yam, Y. (2005) Spectral analysis and the dynamic response of complex networks. *Phys. Rev. E* 71:016106.
- [28] Zhou, H. & Lipowsky, R. (2005) Dynamic pattern evolution on scale-free networks. *PNAS* 102:10052-10057.
- [29] Claus, V., Ehrig, H. & Rozenberg, G., eds. (1979) *Proceedings of the International Workshop on Graph-Grammars and Their Application to Computer Science and Biology*, October 30-November 3, 1978, Bad Honnef, Germany. Lecture Notes in Computer Science 73, Springer-Verlag.

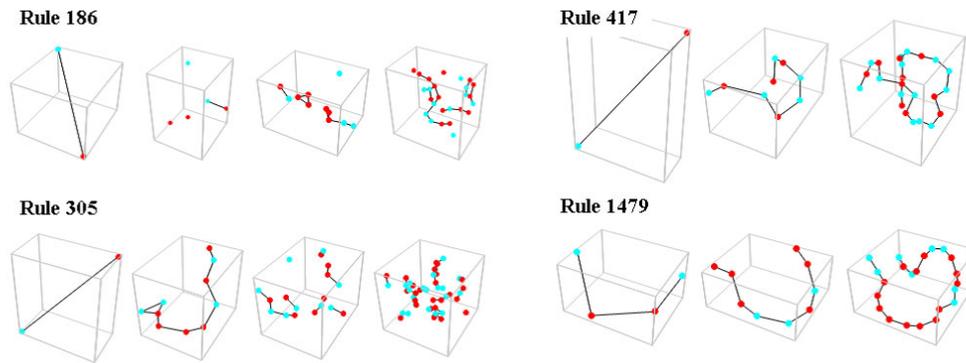


Fig. 7. Characteristic network growth processes found in the intermediate cases indicated in Fig. 6. Rule 186 and 305 are typical examples that show occasional production of GNA fragments. 417 and 1479 are another set of examples that show nonhomogeneous growth of linear GNA strands.

- [30] Ehrig, H., Nagl, M. & Rozenberg, G., eds. (1983) *Proceedings of the Second International Workshop on Graph-Grammars and Their Application to Computer Science*, October 4-8, 1982, Haus Ohrbeck, Germany. Lecture Notes in Computer Science 153, Springer-Verlag.
- [31] Ehrig, H., Nagl, M., Rozenberg, G. & Rosenfeld, A., eds. (1987) *Proceedings of the Third International Workshop on Graph-Grammars and Their Application to Computer Science*, December 2-6, 1986, Warrenton, VA. Lecture Notes in Computer Science 291, Springer-Verlag.
- [32] Ehrig, H., Kreowski, H.-J. & Rozenberg, G., eds. (1991) *Proceedings of the Fourth International Workshop on Graph-Grammars and Their Application to Computer Science*, March 5-9, 1990, Bremen, Germany. Lecture Notes in Computer Science 532, Springer-Verlag.
- [33] Rozenberg, G., ed. (1997) *Handbook of Graph Grammars and Computing by Graph Transformation Volume 1: Foundations*. World Scientific.
- [34] Lindenmayer, A. (1968) Mathematical models for cellular interaction in development I. Filaments with one-sided inputs. *J. Theor. Biol.* 18:280-289.
- [35] Doi, H. (1984) Graph-theoretical analysis of cleavage pattern: Graph developmental system and its application to cleavage pattern of ascidian egg. *Development, Growth & Differentiation* 26:49-60.
- [36] Wu, A. & Rosenfeld, A. (1979) Cellular graph automata I. basic concepts, graph property measurement, closure properties. *Information and Control* 42:305-329.
- [37] Wu, A. & Rosenfeld, A. (1979) Cellular graph automata II. graph and sub-graph isomorphism, graph structure recognition. *Information and Control* 42:330-353.
- [38] Tomita, K., Kurokawa, H. & Murata, S. (2002) Graph automata: Natural expression of self-replication. *Physica D* 171:197-210.
- [39] Salzberg, C., Sayama, H. & Ikegami, T. (2004) A tangled hierarchy of graph-constructing graphs. In Pollack, J., Bedau, M., Husbands, P., Ikegami, T. & Watson, R. A., eds., *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, September 12-15, 2004, Boston, MA, pp.495-500. MIT Press.
- [40] Sano, K. & Sayama, H. (2006) Wriggraph: A kinetic graph model that uniformly describes ontogeny and motility of artificial creatures. In Rocha, L. M., Jaeger, L. S., Bedau, M. A., Floreano, D., Goldstone, R. L. & Vespignani, A., eds., *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, June 3-7, 2006, Bloomington, IN. MIT Press. In press.
- [41] Nehaniv, C. L. (2004) Asynchronous automata networks can emulate any synchronous automata network. *Int. J. Algebra & Computation* 14:719-739.
- [42] Wuensche, A. (2002) Discrete Dynamics Lab: Tools for investigating cellular automata and discrete dynamical networks. *Kybernetes* 32:77-104. Software freely available from <http://www.ddlab.com/>.