State Clustering and Declustering of 3-Regular Graphs with Structural Rewriting

Kohji Tomita and Haruhisa Kurokawa

National Institute of Advanced Industrial Science and Technology (AIST)
Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568 Japan
{k.tomita,kurokawa-h}@aist.go.jp
http://staff.aist.go.jp/k.tomita

Abstract. We consider a variant of graph developing system and show various behavior with one type of graph-rewriting. This system is based on rewriting of 3-regular graphs with two possible states per node. We focus on a simple case of fixed number of nodes. The development processes include interaction among rather stable subgraphs. Some simple behaviors, such as clustering and declustering of states, are shown by simulation.

Keywords: cellular automata, graph-rewriting, graph automata, adaptive network.

1 Introduction

Recently, there has been increasing attention on adaptive networks [2]. There, co-evolution of topology (network) and states has been studied in various contexts such as complex networks. In modeling, analyzing or designing systems comprising many elements in full detail, it is important to clarify the dynamics of each element and the relations among elements. In most cases, structures and states are coupled closely in the sense that the global structure constrains the behavior of each element and the behaviors of the elements affect on the structure. One of the interesting behavior of such graph development in our concern is emergence of hierarchy such that graph is decomposed appropriately into subgraphs, and the overall dynamics is described by the interaction among subgraphs and dynamics in subgraphs.

We have been studying a particular class of graph dynamics called graph-rewriting automata [4]. It is an extension of cellular automata to dynamic graph structures. As in cellular automata, the number of neighbors of each cell is unchanging. Graphs are rewritten synchronously according to several types of local rules with changing capabilities of the number of nodes. Though our framework only treats 3-regular graphs [1], it is sufficiently general to represent various behaviors of network evolution such as self-organization or self-reconfiguration including self-replication.

In this paper, we restrict our attention to the behavior of planar graphs with fixed number of nodes and focus on cluster formation. The model is based on

J. Suzuki and T. Nakano (Eds.): BIONETICS 2010, LNICST 87, pp. 475–482, 2012.

rewriting of 3-regular planar graphs with two possible states per node. As for structural rewriting, we adopt a variant of a rule called commutation in the previous study [4]. It changes local connective relation of a pair of connected nodes. As a simple cluster, we adopt a connected subgraph all of whose nodes have the same state, connected to other subgraphs with different states. We see clusters are formed by a rewriting rule through simulation. In the development process, interaction occurs among clusters whose structure is rather stable.

In the following, formulation of our framework is given, and then simulation setting and results are shown. Finally, the conclusion follows.

2 Graph-Rewriting Automata

Let us introduce the framework that we consider in this paper. It is a variant of graph-rewriting systems, called a graph-rewriting automaton. (Evolution of networks based on similar rewriting rules is discussed in [7].) We assume that the base graph structure is a 3-regular planar graph: each node has three neighbor nodes. Different from ordinary graphs, a cyclic order of links is defined on each node. Each node has an internal state chosen from a finite set. More formally, it is defined as follows. The set of all two element subsets of a set A is denoted by $\mathcal{P}_2(A)$, i.e., $\mathcal{P}_2(A) = \{\{x, y\} | x, y \in A \text{ and } x \neq y\}$. Let $I = \{0, 1, 2\}$.

Definition 1. A base graph G is a triplet $\langle V, E, \xi \rangle$, where V is a finite set of vertices, E is a set of edges defined in the following, and $\xi : V \to S$ is a function that assigns a state to each vertex. Each edge specifies two incident vertices with link indices I; more formally, E is a subset of $\mathcal{P}_2(V \times I)$ such that for every $\langle u, i \rangle \in V \times I$ there exists just one $\langle v, j \rangle \in V \times I$ such that $\{\langle u, i \rangle, \langle v, j \rangle\} \in E$.

This definition permits multiple edges or self-edges (loops). Hereafter, base graphs are also called graphs for simplicity. In this paper, we use only two states 0 and 1, i.e., $S = \{0, 1\}$, denoted by white and black nodes respectively in the figures. Also, three incident links of a node are drawn clockwise on this order around the node in sections 2 and 3.

Isomorphism between two graphs are defined by, in addition to the usual condition, that states of the nodes are the same and that cyclic order of the links defined on each node are the same. (As for link indices, only the order rather than the number is in concern.) In the following, isomorphic graphs are identified. An example of non-isomorphic graphs due to the link order is shown in Fig.1.

We allow two types of rewriting, *cw* and *ccw*, as shown in Fig. 2. The above cyclic order is used to uniquely determine the nodes that are connected to the involved nodes in each structural rewriting.

A link is called applicable if its incident nodes have different states. At each time step, one applicable link is chosen randomly and one rewriting is applied. When the target link is one of double links, a loop is generated as in Fig. 3, in which rewriting is applied to the bold link of the left graph.

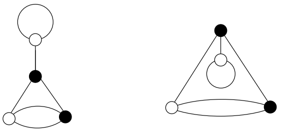


Fig. 1. Non-isomorphic graphs by link order

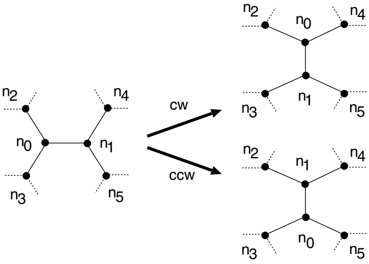


Fig. 2. Structural rewriting

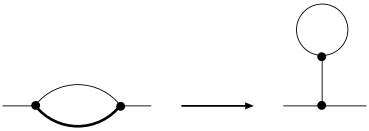


Fig. 3. Application to one of double links

This rewriting is complete for planar connected graphs with the same number of nodes in the following sense. If we omit the states and focus only on the connective relation among nodes, we can rewrite any connected graphs into any connected graph with the same number of nodes by choosing an appropriate sequence of links to which the rewriting is applied.

Depending on the states of target and neighbor nodes, a rewriting rule specifies which rewriting is executed. We use two kinds of rewriting rules, called *phobic* and *philic*, as defined below. In the following, $s(n)$ is the state of node n. Let $s_u = s(n_2) + s(n_4)$ and $s_l = s(n_3) + s(n_5)$.

- *phobic rule*:

$$\begin{cases} \text{cw, if } s(n_0) = 0 \text{ and } s_u < s_l \text{ or } s(n_0) = 1 \text{ and } s_u > s_l, \\ \text{ccw, if } s(n_0) = 0 \text{ and } s_u > s_l \text{ or } s(n_0) = 1 \text{ and } s_u < s_l, \\ \text{randomly chosen, if } s_u = s_l. \end{cases}$$

- *philic rule*:

$$\begin{cases} \text{cw, if } s(n_0) = 0 \text{ and } s_u > s_l \text{ or } s(n_0) = 1 \text{ and } s_u < s_l, \\ \text{ccw, if } s(n_0) = 0 \text{ and } s_u < s_l \text{ or } s(n_0) = 1 \text{ and } s_u > s_l, \\ \text{randomly chosen, if } s_u = s_l. \end{cases}$$

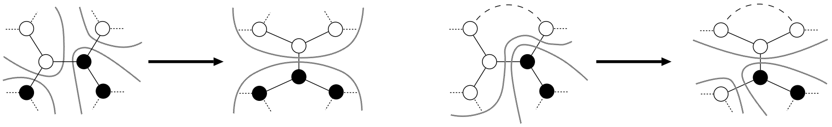


Fig. 4. Cluster merging and separation by phobic rule

The phobic rule tends to phase separation (like oil and water), whereas the philic rule tends to mixing.

3 Mesoscopic View

Graph development processes are generally complicated when a large number of nodes and intricate connective relation among nodes are involved. Instead of viewing the whole graph at the level of individual nodes, viewing it at an appropriate level of abstraction might be useful, if possible. One desirable description would be such that the whole graph is divided into parts, i.e., subgraphs, so that the dynamics of the whole graph is determined by the interaction among subgraphs, and that the interaction is described by a kind of states of the subgraphs. Then the whole behavior could be described at the level of subgraphs. Such will lead to hierarchical description of the graph development processes. This is suitable when graphs are developed in the hierarchical manner like living things with some organizing principle. Similar idea was partly explored as a meta-node in [5], but it was complex and given a priori.

In this paper, we try the simplest case for such description by using two kinds of rules: phobic and philic. As a simple cluster, we adopt a connected subgraph all of whose nodes have the same state, connected to other subgraphs with different states. In the development process, nodes within clusters have the same state, and hence are rather stable because structural rewriting is not applied between the nodes with the same node state. On the other hand, at the interface of two clusters, structural rewriting is performed, which leads to cluster merging and separation (see Fig. 4). The rewriting has different effects to the clusters depending on the global connective relation. In the following, the effect of these rules is examined through simulation.

4 Simulation

We conduct simulation for 10 initial graphs $G_1, \ldots, G_{10}$ with varying number of nodes.[1] Each graph includes equal number of nodes with state 0 and 1. G_1 and G_2 are shown, together with G_0, in Fig. 5 at the individual node level. The

[1] Each graph G_i is obtained by the system in [4] using a rule set 'div 0, (0, 1, 1), 0', 'div 1, (0, 0, 1), 1', and 'com 0, 1' from an initial graph G_0 (in Fig. 5), composed of four nodes, at i-th step.

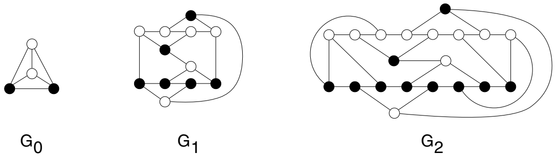


Fig. 5. Graphs for simulation

Table 1. The number of nodes and clusters in initial graphs

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
#nodes	12	20	28	52	84	148	260	428	692	1140
#clusters	6	6	14	18	26	42	66	102	174	282

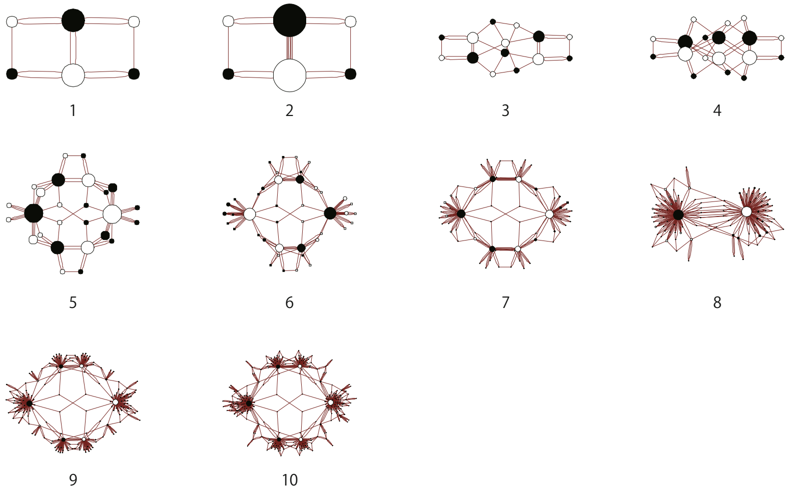


Fig. 6. Initial graphs $G_1, \ldots, G_{10}$ at the cluster level

number of nodes and clusters are shown in Table 1. Figure 6 shows the graphs in the cluster level. Each black or white circle indicates one cluster, and its size is proportional to the number of its member nodes.

We conducted simulation of phobic and philic rules for 100 trials from these initial graphs in 10,000,000 steps. Figure 7 indicates the change of average number of clusters by phobic rule. In the development process by phobic update, clustering is performed, i.e., the number of clusters tend to decrease. Depending on the total number of nodes, the number is different in the simulated steps.

The above figure only shows the average number for 100 trials, and the number deviates by each trial. Figure 8 details the result for G_5 with error bars. The bars indicate minimum and maximum numbers. Figure 9 shows an example of cluster

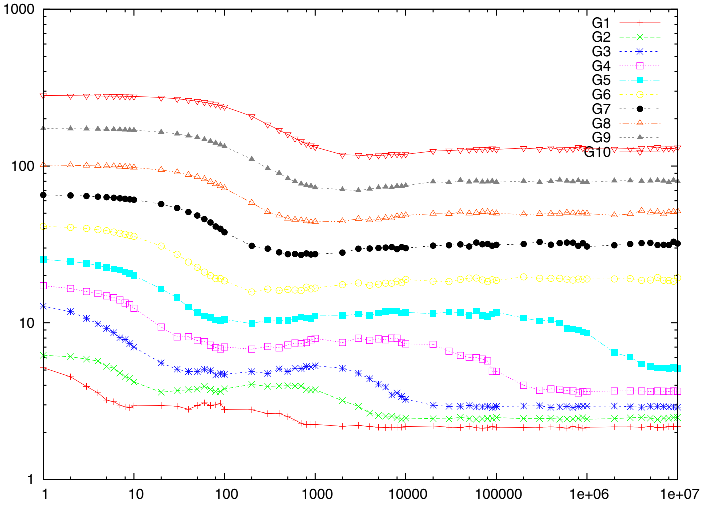


Fig. 7. The average number of clusters by phobic rule

change in one execution at steps 10^i for $i = 1, \ldots, 6$. The number of clusters is shown in parentheses. We can observe by these results in Figs. 7, 8 and 9 that the number of clusters once grows before reducing.

Fig. 10 shows the change of average number of clusters for 100 trials by the philic rule. The number of clusters tend to increase by the philic rule. Depending on the total number of nodes, the number is different. In this simulation setting, the final number of applicable links was about 65–70% of the total number of links. Some of the obtained graphs from G_5 by philic rule are shown as clusters in Fig. 11.

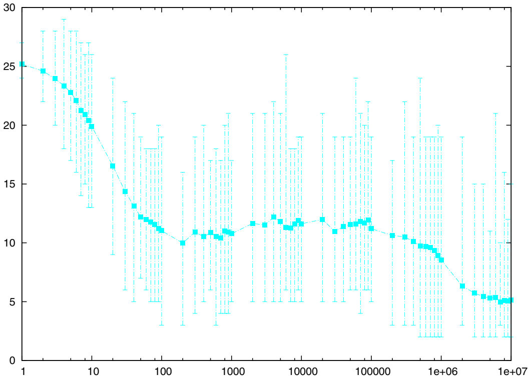


Fig. 8. The number of clusters obtained by phobic rule from G_5. Error bars indicate minimum and maximum.

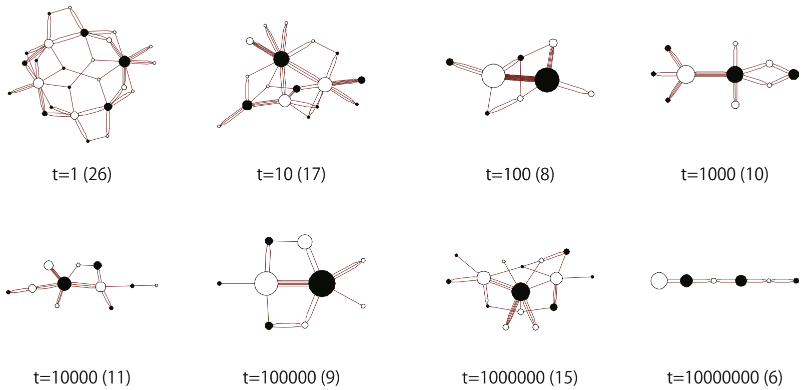

t=1 (26)
t=10 (17)
t=100 (8)
t=1000 (10)
t=10000 (11)
t=100000 (9)
t=1000000 (15)
t=10000000 (6)

Fig. 9. Clusters obtained by phobic rule from G_5

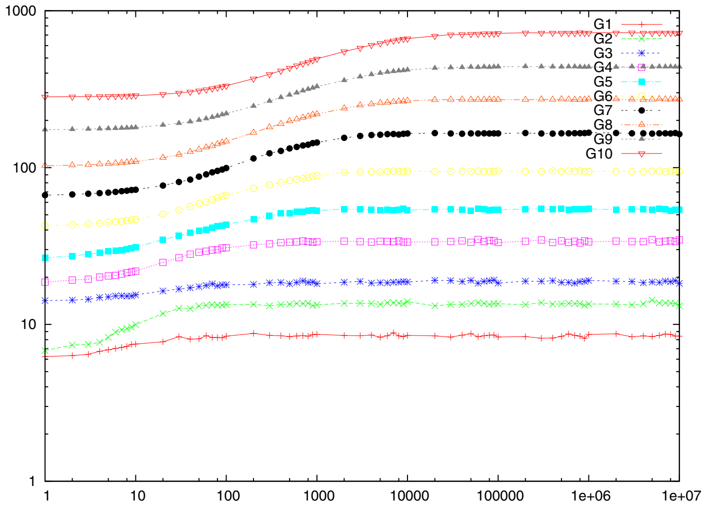
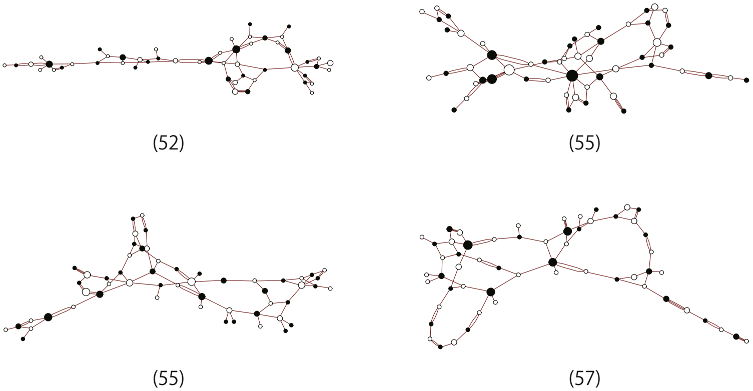


Fig. 10. The average number of clusters by philic rule

5 Conclusion

We have considered a variant of graph developing system and showed behaviors with phobic and philic graph-rewriting rules. This system is based on rewriting of 3-regular graphs with two possible state per node. We focused on a simple case of fixed number of nodes. Simple behaviors, such as clustering and declustering of graphs, are shown by simulation. The phobic rule appears to resulting in clustering, and the philic rule in mixing, regardless of graph and starting state.


(52)

(55)

(55)

(57)

Fig. 11. Examples of obtained clusters by philic rule from G_5

There is much future work. We need to investigate more complex processes. In many cases the phobic rule leaded to two giant clusters in the simulations, but effects of the initial graphs on the behavior are not examined enough. Also, we need more analysis to clearly describe the dynamic behavior of clusters. Development processes were depicted mainly as the change of the number of clusters. This reflects only a limited aspect of the dynamics. The process of clustering was not straightforward and affected by the global structure. We need to develop better methods for describing the process including dynamic structural changes. In addition, analysis of structures and dynamics in a cluster are also necessary.

Acknowledgment. This work was supported by JSPS KAKENHI (21500231).

References

1. Greenlaw, R., Petreschi, R.: Cubic graphs. ACM Computing Surveys 27(4), 471–495 (1995)
2. Gross, T., Sayama, H. (eds.): Adaptive Networks: Theory, Models and Applications. Springer (2009)
3. Rozenberg, R. (ed.): Handbook of Graph Grammars and Computing by Graph Transformation, Foundations, vol. 1. World Scientific (1997)
4. Tomita, K., Kurokawa, H., Murata, S.: Graph Automata: Natural Expression of Self-Reproduction. Physica D 171(4), 197–210 (2002)
5. Tomita, K., Murata, S., Kurokawa, H.: Self-description for construction and computation on graph-rewriting automata. Artificial Life 13(4), 383–396 (2007)
6. von Neumann, J.: Theory of Self-Reproducing Automata. Univ. of Illinois Press (1966)
7. Wolfram, S.: A New Kind of Science. Wolfram Media (2002)