Title	Structural and Behavioral Equivalence Relations in Automata Networks
Author(s)	SAITO, Takashi; NISHIO, Hidenosuke
Citation	数理解析研究所講究録 (1984), 522: 225-239
Issue Date	1984-05
URL	http://hdl.handle.net/2433/98464
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録
522巻 1984年 225-239

Structural and Behavioral Equivalence Relations in Automata Networks

Takashi SAITO and Hidenosuke NISHIO
斉藤 隆 西尾 英之助
Department of Biophysics, Faculty of Science, Kyoto University

1. Introduction

The automata network is an information processing system consisting of intercommunicating finite automata. We can interpret it not only as abstraction of biological systems (like a neural network), but also as theoretical framework of artificial information processing systems (like a logical network). From the logical point of view, the automata network is a generalization of cellular automaton and Maculloch-Pitts neuron network. The former is uniform both in structure and element automaton and the latter is a structurally nonuniform system that consists of one kind of elements (i.e. threshold element). The theory of automata networks can be considered to be the structural theory of automata in contrast to the abstract theory of automata.

We formulate the automata network as an edge- and vertex-labeled directed graph. A vertex-label corresponds to the finite automaton which is to be placed at the vertex and an edge-label corresponds to the label of the input terminal of a finite automaton on the vertex.

In this paper, we define, as to the set of vertices, the structural equivalence relation induced by the structure of the graph and the behavioral equivalence relation induced by the behavior (i.e. state transition) of finite automaton on each vertex and investigate the relationships between these relations. We have obtained some results in particular about the element-uniform system. This study of the equivalence

relations on vertices can be considered to be an approach to the problem of minimizing the number of elements of automata networks.

2. Definition of Automata Networks

We define the automaton networks in the following way. At first we need the notion of a graph.

Definition 2.1 Edge-labeled directed graph G

An edge-labeled graph (simply a graph) G is defined by $G = (V', E)$, where V' is a set of finite or infinite number of vertices with conditions $V' = V \cup V_I$, $V \cap V_I = \phi$ and $V \neq \phi$, where V is called the set of inner vertices and V_I the set of input vertices. E is a set of labeled edges, i.e. $E = (E_1, E_2, .., E_k)$ where $E_i \subseteq V \times V'$ and for every vertex $u \in V$ there is at most one vertex $v \in V'$ such that $(u,v) \in E_i$ for every i $(1 \leq i \leq k)$.

We also write $E_i(u) = v$ if $(u,v) \in E_i$, which means that there is an edge labeled with i from v to u.

When we consider the automata network with output, we specify the set of output vertices $V_0 \subseteq V$. If $V_I = \phi$ (i.e. $V' = V$) then the graph G becomes the base of an autonomous automata network.

Definition 2.2 Cell M

An automata network is composed of many cells, each of which is a finite automaton placed at an inner vertex of graph G and defined as follows.

A cell (or element automaton) M is defined by 3-tuple $M = (Q, k, f)$, where Q is a finite set of inner states (Q designates also the set of input and output alphabets), k is the number of input terminals of M and f is the state transition function called the local map defined as $f : Q^{k+1} \rightarrow Q$. If $f(q, x_1, x_2, \ldots, x_k) = q'$, then q' represents the next state provided that

the present state is $q \in Q$ and the input to the i-th terminal is $x_i \in Q$. The output of M is defined to be the same as the present state of M. The set of all kinds of cells is denoted by **M**.

Definition 2.3 Automata network A

Suppose that a graph G is given. Then we can define an automata network $A = (G,\alpha)$ by allocating cells to vertices.

That is, in $A = (G,\alpha)$, G is an edge-labeled directed graph and α is an allocation map of cells, i.e. $\alpha : V \to \mathbf{M}$. If the indegree of $v \in V$ is k, then $\alpha(v)$ is taken to be a k-input cell. It is also assumed that α maps every vertex to a cell having a common state set Q.

$A = (G,\alpha)$ is called the element-uniform automata network if α maps all vertices to one cell. In this case, α is also called uniform. The so called "cellular automaton" is element-uniform automata network with "uniform" unlabeled graph.

Now we define the behavior of automata networks after that of cellular automata.

Definition 2.4 Behavior of automata networks

Assume that $A = (G,\alpha)$ is given. When f is the local map of $\alpha(v)$ (i.e. the cell M placed at $v \in V$), we sometimes represent it by $\alpha(v)$ if there is no confusion.

Let $\Phi = \{\varphi \mid \varphi : V_I \to Q\}$ be the set of all input configurations to automata network A. That is, $\varphi(v) = x$ represents that the input state of the input vertex v of automata network A is x.

$\mathbf{C} = \{c \mid c : V \to Q\}$ is the set of all state configurations of A as in the case of a usual cellular automaton. We define the next state configuration c' as follows: Let c and φ be the present state configuration

and the input configuration to A respectively. For $v \in V$, $c'(v) = \tilde{\alpha}_\varphi(c)(v) = \alpha(v)(c(v), c(E_1(v)), \ldots, c(E_k(v)))$. We consider that if $E_i(v) \in V_I$ then $c(E_i(v)) = \varphi(E_i(v))$. The mapping $\tilde{\alpha}_\varphi: \mathbf{C} \to \mathbf{C}$ thus defined is called the global map as opposed to the local map.

3. Equivalence Relations on V

Definition 3.1 Equivalence relations on V

When $G = (V', E)$ is given, we denote by $\mathbf{R}_G$ the set of all equivalence relations (or partitions) on V. For $R \in \mathbf{R}_G$, when the vertices u and v are equivalent in the sense of R, we denote $(u,v) \in R$. $R_0 \in \mathbf{R}_G$ denotes the equivalence relation such that for every pair of vertices u and v ($u \neq v$) $(u,v) \notin R_0$. That is every pair of vertices are not equivalent. R_U denotes the equivalence relation such that for every u, $v \in V$, $(u,v) \in R_U$. This means that the number of equivalence classes is one. If there is no confusion, we may use $\mathbf{R}$ instead of $\mathbf{R}_G$.

When V is finite, an equivalence relation $R \in \mathbf{R}$ is expressed by specifying every equivalence classes like $R = (\{v_1^1, .., v_{c1}^1\}, \{v_1^2, .., v_{c2}^2\}, \ldots)$. So when $V = \{v_1, .. v_n\}$, $R_0 = (\{v_1\}, \{v_2\}, .., \{v_n\})$ and $R_U = (\{v_1, v_2, .., v_n\})$. Sometimes we use the abbreviated notation like $(\{v_1, v_2\}, v_3, v_4)$ in place of $(\{v_1, v_2\}, \{v_3\}, \{v_4\})$.

For R, $R' \in \mathbf{R}_G$, we denote $R' < R$ if R' is a refinement of R.

Definition 3.2 Structural equivalence relation (SER)

$R \in \mathbf{R}_G$ is called a structural equivalence relation (SER) of automata network $A = (G, \alpha)$, iff the following conditions are satisfied.

If $(u,v) \in R$, then $\alpha(u) = \alpha(v)$

and $(E_i(u), E_i(v)) \in R$ for every i ($1 \leqq i \leqq k$).

If $E_i(u) \in V_I$ or $E_i(v) \in V_I$, we formally define that $(E_i(u),E_i(v)) \in R$ only in the case of $E_i(u) = E_i(v) \in V_I$, and $(E_i(u),E_i(v)) \notin R$ in other cases. If $E_i(u)$ or $E_i(v)$ are undefined, we formally define that $(E_i(u),E_i(v)) \in R$ only in the case that $E_i(u)$ and $E_i(v)$ are both undefined, and $(E_i(u),E_i(v)) \notin R$ in other cases.

S_A denotes the set of all SER's of $A = (G,\alpha)$. S_A is often denoted by S_G if A is element uniform.

Algorithm for construction of S_A

We assume here that V is finite and α is uniform. At first, we define u_x for $u \in V$ and $x = E_{i_1}E_{i_2}....E_{i_m} \in \{E_1,E_2,..,E_k\}^*$ as follows:

$u_x = E_{i_m}(..(E_{i_2}(E_{i_1}(u)))..)$.

For arbitrarily given SER S and vertex pair u and v such that $(u,v) \notin S$, we construct the minimum SER R such that $(u,v) \in R$ and $S < R$ (i.e. $(u,v) \in R$ and $(u,v) \notin R'$ for every refinement R' of R) by the following algorithm.

Algorithm A

1) For every vertex pair p and q, let $(p,q) \in R$ if $(p,q) \in S$.
2) Let $(u,v) \in R$.
3) For every $x \in \{E_1,..E_k\}^*$ such that $1 \leqq |x| \leqq |V|^2$ where $|V|$ is the number of vertices, let $(u_x,v_x) \in R$.

By Algorithm A we say that R is constructed from S and the vertex pair u and v.

Proposition 3.1

For given SER S and vertex pair u and v such that $(u,v) \notin S$, Algorithm A gives the minimum SER R such that $(u,v) \in R$ and $S < R$.

Proof

1) R is a SER.

At first we note that the number of the distinct vertex pairs (p,r) is $|V|^2$. Furthermore for $x \in \{E_1,..,E_k\}^*$ such that $|x| = |V|^2$, there exist substrings x_1, x_2 and x_3 such that $x = x_1x_2x_3$ and $(u_{x_1},v_{x_1}) = (u_{x_1x_2},v_{x_1x_2})$. Thus there exists a proper prefix x' of x which satisfies that $(u_x,v_x) = (u_{x'},v_{x'})$.

Let $p, q \in V$ and $(p,q) \in R$. Then from the construction of R, there exists $x \in \{E_1,..,E_k\}^*$ such that $p = u_x$ and $q = v_x$.

If $|x| < |V|^2$ then $(E_i(p),E_i(q)) = (E_i(u_x),E_i(v_x)) = (u_{xE_i},v_{xE_i}) \in R$ for every i $(1 \leqq i \leqq k)$. This is because $|xE_i| \leqq |V|^2$.

If $|x| = |V|^2$, then by the above remark $(p,q) = (u_x,v_x) = (u_{x'},v_{x'})$ where x' is a proper prefix of x. Then this case is also reduced to the above case.

Therefore if $(p,q) \in R$ then $(E_i(p),E_i(q)) \in R$ for every i $(1 \leqq i \leqq k)$.

2) $(u,v) \in R$.

It is trivial from the construction of R.

3) R is the minimum SER.

Suppose that SER R' is a refinement of R and $(u,v) \in R'$. Since $R' \neq R$, there exists a vertex pair p and q such that $(p,q) \in R$ and $(p,q) \notin R'$. By the construction of R, there exists $x \in \{E_1,..,E_k\}^*$ such that $p = u_x$ and $q = v_x$. Let $x = E_{i_1}E_{i_2}...E_{i_m}$. From the definition of SER, if $(p,q) = (u_x,v_x) \notin R'$ then $(u_{E_{i_1}..E_{i_{m-1}}},v_{E_{i_1}..E_{i_{m-1}}}) \notin R'$. So $(u_{E_{i_1}..E_{i_{m-2}}},v_{E_{i_1}..E_{i_{m-2}}}) \notin R'$. By repeating this reasoning we obtain $(u,v) \notin R'$. This is a contradiction. Therefore R is the minimum SER such that $(u,v) \in R$.

4) $S < R$.

It is trivial from the construction of R. ∎

Let $\mathbf{S}$ be a set of SER's. SER(S) denotes the set of all SER's each of which is constructed by the Algorithm A from an element R of $\mathbf{S}$ and a vertex pair u and v such that $(u,v) \notin R$.

Algorithm B

At first we make the sequence of sets of SER's as follows:

$\mathbf{S}_0 = \{R_0\}, \mathbf{S}_1 = SER(\mathbf{S}_0), \ldots, \mathbf{S}_j = SER(\mathbf{S}_{j-1}), \ldots, \mathbf{S}_{m+1} = SER(\mathbf{S}_m) = \{R_U\}$ where $\mathbf{S}_m$ is the first set of SER's such that $SER(\mathbf{S}_m)$ is $\{R_U\}$.

Let $\mathbf{S}_A = \bigcup_{j=1}^{m+1} \mathbf{S}_j$.

Proposition 3.2

For a given automata network $A = (G,\alpha)$ where G is finite and α is uniform, Algorithm B gives $\mathbf{S}_A$.

Proof

Let $\mathbf{S}$ be the set of SER's constructed by Aalgorithm B. If there exists a SER $R \notin \mathbf{S}$, then there exists a sequence of SER's $R_0 < R_1 < \ldots < R_j < \ldots < R_m = R$ where each R_j is the minimum SER such that $R_{j-1} < R_j$. From each step of the algorithm, since $R_0 \in \mathbf{S}_0$, $R_1 \in \mathbf{S}_1$. So $R_2 \in \mathbf{S}_2$. By repeating this reasoning, $R \in \mathbf{S}$. This is a contradiction. ∎

Definition 3.3 State configurations which are consistent with an equivalence relation

A configuration $c \in \mathbf{C}$ is defined to be consistent with an equivalence relation $R \in \mathbf{R}_G$, iff if $(u,v) \in R$, then $c(u) = c(v)$.

Such a configuration is also called "R-consistent". $\mathbf{C}_R$ denotes the set of all R-consistent configurations.

Definition 3.4 Behavioral equivalence relation (BER)

For $A = (G,\alpha)$, $B \in \mathbf{R}_G$ is said to be a behavioral equivalence relation

(BER), if the following condition is satisfied.

If $(u,v) \in B$, then for every $c \in C_B$ and for every $\varphi \in \Phi$

$\tilde{\mathcal{L}}_\varphi(c)(u) = \tilde{\mathcal{L}}_\varphi(c)(v)$.

B_A is the set of all BER's of $A = (G,\alpha)$. we also write B_α in place of B_A if G is understood.

4. Some Basic Properties of Two Equivalence Relations

Proposition 4.1

Suppose that there are two graphs $G^1 = (V',E^1)$ and $G^2 = (V',E^2)$ with the common set of vertices. $E^1 = (E_1,..,E_k)$ and $E^2 = (E_{k+1},..,E_m)$. Then $S_{G^1 \cup G^2} = S_{G^1} \cap S_{G^2}$.

Remark: The union of two graphs $G^1 \cup G^2$ is defined according to the graph theory, i.e. $G^1 \cup G^2 = (V',E^1 \cup E^2)$ where $E^1 \cup E^2 = (E_1,E_2,..,E_k, E_{k+1},..,E_m)$.

Corollary 4.2 Let $G = (V',(E_1,..E_k))$ and $G_i = (V',E_i)$. Then $S_G = \bigcap_{i=1}^{k} S_{G_i}$.

Definition 4.1 Union of equivalence relations

For $R, R' \in R_G$, we define their union $R \vee R'$ as follows: $(u,v) \in R \vee R'$ iff there exists a sequence of vertices $u = u_1,u_2,...,u_m = v$ such that $(u_i,u_{i+1}) \in R$ or $(u_i,u_{i+1}) \in R'$ for every i $(1 \leq i \leq m-1)$.

It is clear that if R and R' are equivalence relations in R_G then the union $R \vee R'$ is also an equivalent relation in R_G.

Proposition 4.3 Let $A = (G,\alpha)$. Then S_A and B_A are closed under the union of relations.

Proof

1) We prove first if $R, R' \in S_A$ then also $R \vee R' \in S_A$.

Suppose that $(u,v) \in R \vee R'$. Thus there exists a sequence of vertices $u = u_1, u_2, \ldots, u_m = v$ such that $(u_j, u_{j+1}) \in R$ or $(u_j, u_{j+1}) \in R'$ for every j $(1 \leqq j \leqq m-1)$. Then for every j $(1 \leqq j \leqq m-1)$ the next condition holds: $\alpha(u_j) = \alpha(u_{j+1})$ and $(E_i(u_j), E_i(u_{j+1})) \in R$ or $(E_i(u_j), E_i(u_{j+1})) \in R'$ for every i $(1 \leqq i \leqq k)$.

So $\alpha(u) = \alpha(v)$ and $(E_i(u), E_i(v)) \in R \vee R'$ for every i $(1 \leqq i \leqq k)$.

Therefore $R \vee R' \in S_A$.

2) Secondly we prove that if $R, R' \in B_A$ then $R \vee R' \in B_A$.

For this we show that if $(u,v) \in R \vee R'$ then for every $c \in C_{R \vee R'}$ and for every $\varphi \in \Phi$, $\widetilde{\alpha}_\varphi(c)(u) = \widetilde{\alpha}_\varphi(c)(v)$.

From the definition of union, if $(u,v) \in R \vee R'$ then there exists a sequence of vertices $u = u_1, .., u_m = v$ such that $(u_j, u_{j+1}) \in R$ or $(u_j, u_{j+1}) \in R'$ for every j $(1 \leqq j \leqq m-1)$. Since $C_{R \vee R'} \subseteq C_R$ and $C_{R \vee R'} \subseteq C_{R'}$, $C_{R \vee R'} = C_R \cap C_{R'}$. So for every $c \in C_{R \vee R'}$ and for every $\varphi \in \Phi$, $\widetilde{\alpha}_\varphi(c)(u_j) = \widetilde{\alpha}_\varphi(c)(u_{j+1})$ for every j $(1 \leqq j \leqq m-1)$. Thus $\widetilde{\alpha}_\varphi(c)(u) = \widetilde{\alpha}_\varphi(c)(v)$. Therefore $R \vee R' \in B_A$. ∎

Definition 4.2 Intersection of equivalence relations

We define $R \wedge R'$, the intersection of R and R', as follows.

$(u,v) \in R \wedge R'$ iff $(u,v) \in R$ and $(u,v) \in R'$

The intersection of equivalence relations thus defined is clearly an equivalence relation.

Proposition 4.4

S_A and B_A are closed under the intersection of equivalence relations.

Proposition 4.5 For every $A = (G,\alpha)$, $S_A \subseteq B_A \subseteq R_G$.

Proof $B_A \subseteq R_G$ is trivial. So we will show $S_A \subseteq B_A$. Take a pair (u,v) from $R \in S_A$. So since $(u,v) \in R$, $\alpha(u) = \alpha(v)$ and $(E_i(u),E_i(v)) \in R$ for every i $(1 \leq i \leq k)$. So for every $c \in C_R$, $c(u) = c(v)$ and $c(E_i(u)) = c(E_i(v))$ for every i $(1 \leq i \leq k)$ (see Definition 2.4 and Definition 3.2). So for every $\varphi \in \Phi$,

$$\begin{aligned}\tilde{\alpha}_\varphi(c)(u) &= \alpha(u)(c(u),c(E_1(u)),..,c(E_k(u))) \\ &= \alpha(v)(c(v),c(E_1(v)),..,c(E_k(v))) = \tilde{\alpha}_\varphi(c)(v).\end{aligned}$$

Thus $R \in B_A$. ∎

Proposition 4.6 If α is uniform, then for every G, $S_G = \bigcap_\alpha B_\alpha$.

Proof $S_G \subseteq \bigcap_\alpha B_\alpha$ is trivial from Proposition 4.5. So we will show $S_G \supseteq \bigcap_\alpha B_\alpha$. Let $R \in \bigcap_\alpha B_\alpha$. Then from the definition of B_α the next condition holds. For every $c \in C_R$, for every $\varphi \in \Phi$, and for every α, if $(u,v) \in R$ then $\tilde{\alpha}_\varphi(c)(u) = \tilde{\alpha}_\varphi(c)(v)$.

So $\alpha(u)(c(u),c(E_1(u),..,c(E_k(u))) = \alpha(v)(c(v),c(E_1(v),..,c(E_k(v)))$.

If there exists i such that $(E_i(u),E_i(v)) \notin R$ then there exists $c^* \in C_R$, $c^*(E_i(u)) \neq c^*(E_i(v))$

We obtain that there exists α^* such that

$$\begin{aligned}&\alpha^*(u)(c^*(u),c^*(E_1(u)),..,c^*(E_i(u)),..,c^*(E_k(u))) \\ &\quad \neq \alpha^*(v)(c^*(v),c^*(E_1(V)),..,c^*(E_i(v)),..,c^*(E_k(v)))\end{aligned}$$

This is a contradiction. So for every i $(E_i(u),E_i(v)) \in R$. Therefore $R \in S_G$. ∎

5. Relationships between S_A, B_A and R_G

Proposition 5.1

There is an automata network $A = (G,\alpha)$ such that $S_A \subsetneq B_A \subsetneq R_G$.

Proof To prove the proposition, we construct the automata network as follows.

$A = (G,\alpha)$

$G = (\{v_1,v_2,v_3,v_4\},(E_1,E_2))$ $V_I = \phi$

$E_1 = \{(v_1,v_2),(v_2,v_1),(v_3,v_1),(v_4,v_2)\}$

$E_2 = \{(v_1,v_3),(v_2,v_3),(v_3,v_4),(v_4,v_3)\}$

$Q = \{0,1\}$

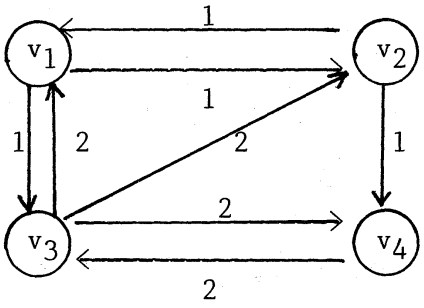

Fig.1

α is uniform and local map f is as follows:

$f = f(q,x_1,x_2)$ where q,x_1 and x_2 are the present state, the first input and the second one, respectively.

$f(0,0,0) = f(0,0,1) = f(0,1,0) = 1$

In the other cases the value of f is defined to be 0.

Thus the graph G is illustrated as in Fig.1.

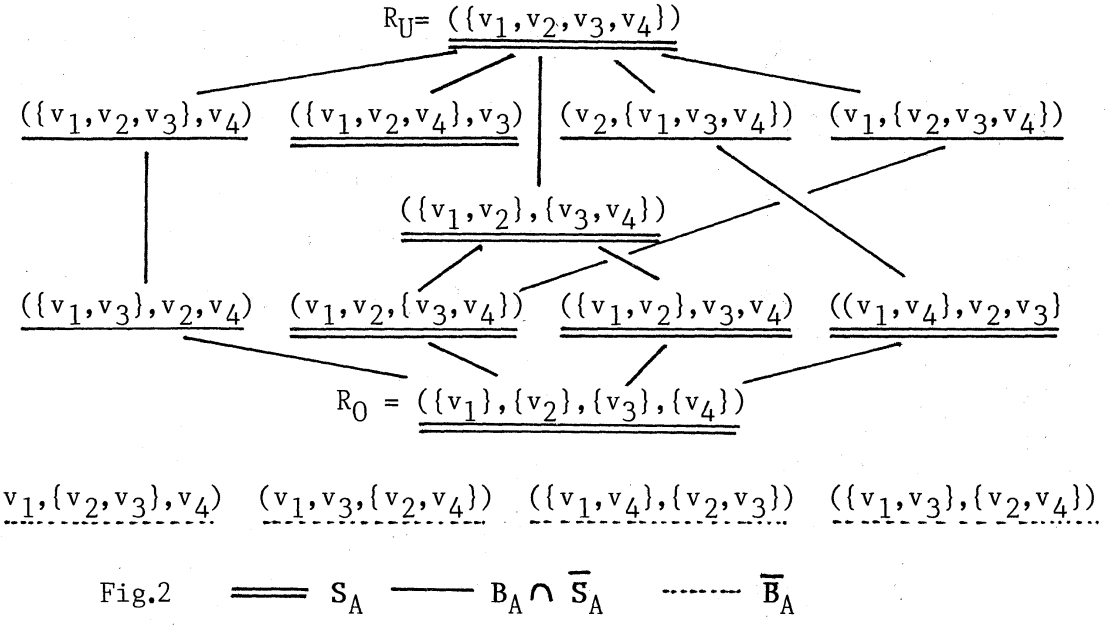

Fig.2 ═══ S_A ——— $B_A \cap \overline{S}_A$ ····· $\overline{B}_A$

We can obtain all equivalence relations in S_A, B_A and R_G as illustrated in Fig.2. S_G is constructed by Algorithm B in section 3. To construct B_A we check all equivalence relations in $\overline{S}_A$ according to the definition of BER. We use the abbreviated notation of equivalence relation

and not all refinement relations are illustrated. ∎

In Proposition 4.3, we showed that $\mathbf{S}_A$ and $\mathbf{B}_A$ are closed under the union of equivalence relation. Therefore if $S \in \mathbf{S}_A$ and $B \in \mathbf{B}_A$, then $S \vee B \in \mathbf{B}_A$. The next proposition shows that there is a case where $S \vee B \in \mathbf{S}_A$.

Proposition 5.2 There exists an automata network $A = (G,\alpha)$ such that for some $S \in \mathbf{S}_A$ and $B \in \mathbf{B}_A \cap \overline{\mathbf{S}}_A$, $S \vee B \in \mathbf{S}_A$ holds.

Proof To prove the proposition, we construct the automata network as follows. Graph G ($V_I = \phi$) is illustrated in Fig.3. α is uniform, $Q = \{0,1\}$ and local map f satisfies the next condition: $f(q,0,x_2) = f(q,1,x_2)$ for every q and $x_2 \in Q$.

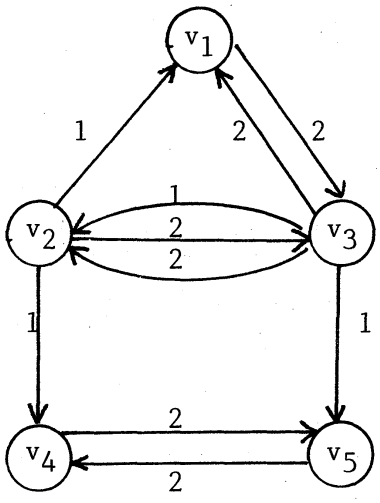

Fig.3

$S_A = \{R_0, R_U, S_1, S_2, S_3\}$ where

$R_0 = (\{v_1\},\{v_2\},\{v_3\},\{v_4\},\{v_5\})$

$R_U = (\{v_1,v_2,v_3,v_4,v_5\})$

$S_1 = (\{v_1,v_3\},v_2,v_4,v_5)$

$S_2 = (\{v_1,v_2,v_3\},v_4,v_5)$

$S_3 = (\{v_1,v_2,v_3\},\{v_4,v_5\})$

$\mathbf{B}_A \cap \overline{\mathbf{S}}_A = \{B_1,B_2,B_3,B_4\}$ where

$B_1 = (\{v_1,v_2\},v_3,v_4,v_5)$ $\quad B_2 = (v_1,v_2,v_3,\{v_4,v_5\})$

$B_3 = (\{v_1,v_3\},v_2,\{v_4,v_5\})$ $\quad B_4 = (\{v_1,v_2\},v_3,\{v_4,v_5\})$

Since $S_1 \vee B_1 = S_2 \in \mathbf{S}_A$ and $S_2 \vee B_2 = S_3 \in \mathbf{S}_A$, the proposition is proved. ∎

Propositions 4.3 and 4.4 show that $\mathbf{S}_A$ and $\mathbf{B}_A$ are closed under union and intersection of equivalence relations, but they do not imply that every refinement of an equivalence relation R in $\mathbf{S}_A$ or $\mathbf{B}_A$ is necessarily a member

of $\mathbf{S}_A$ or $\mathbf{B}_A$ respectively. In fact, in the example used in the proof of Proposition 5.2, B_4 is a refinement of S_3 but it is not a member of $\mathbf{S}_A$.

Definition 5.1 One-pair relation

A relation $R \in \mathbf{R}_G$ is said to be one-pair relation, if there exist v_1, $v_2 \in V$ such that $R = (\{v_1,v_2\},\{v_3\},..,\{v_n\},..)$ where $V = \{v_1,v_2, v_3,..,v_n,..\}$.

Let OP_{uv} denote a one-pair relation which satisfies that $u \neq v$, $(u,v) \in OP_{uv}$. By $\mathbf{R}_1$ we denote the set of all one pair relations, where the set of vertices is understood.

Proposition 5.3 If $\mathbf{R}_1 \subseteq \mathbf{S}_A$ (or $\mathbf{B}_A$), then $\mathbf{S}_A$ (or $\mathbf{B}_A$) = $\mathbf{R}_G$.

In some cases, however, there exists a non trivial equivalence relation R (i.e. $R \neq R_0$, $R \neq R_U$) in $\mathbf{S}_A$, while $\mathbf{S}_A$ contains no one-pair relation. In fact we have the following proposition.

Proposition 5.4 There exists an automata network $A = (G,\alpha)$ such that for any one-pair relation $OP_{uv} \notin \mathbf{S}_A$ and $\mathbf{S}_A \supsetneq \{R_0,R_U\}$.

Proposition 5.5

Let $A = (G,\alpha)$.

$\mathbf{S}_A = \mathbf{R}_G$ iff the following conditions holds:

α is uniform and for every i ($1 \leq i \leq k$):

$E_i(u) = u$ for every $u \in V$ or

there exists a certain vertex $v \in V'$ such that $E_i(u) = v$ for every $u \in V$.

Proof

1) If $\mathbf{S}_A = \mathbf{R}_G$ then a one-pair relation $OP_{uv} \in \mathbf{S}_A$. Therefore OP_{uv} is the SER such that $(u,v) \in OP_{uv}$. Then $\alpha(u) = \alpha(v)$ and $(E_i(u),E_i(v)) \in OP_{uv}$ for every i ($1 \leq i \leq k$). However OP_{uv} is a one-pair relation, $E_i(u)$ and

$E_i(v)$ must be u or v. Then $E_i(u) = u$ and $E_i(v) = v$ or $E_i(u) = v$ and $E_i(v) = u$ or $E_i(u) = u$ and $E_i(v) = u$ or $E_i(u) = v$ and $E_i(v) = v$ for every i $(1 \leqq i \leqq k)$. Since this condition holds for every one-pair relations, α is uniform and for every i $(1 \leqq i \leqq k)$ $E_i(u) = u$ for every $u \in V$ or there exists a certain vertex $v \in V'$ such that $E_i(u) = v$ for every $u \in V$.

2) From the proof of only if part, it is clear that if the condition holds then all one-pair relations belongs to S_A. Then $\mathbf{S}_A = \mathbf{R}_G$ from Proposition 5.3. ∎

Proposition 5.6 If α is uniform and the element automaton is autonomous, then for every G $\mathbf{B}_A = \mathbf{R}_G$

Proposition 5.7 Suppose that α is uniform and not autonomous. If G is a graph such that indegree of every vertex is one, then $\mathbf{B}_A = \mathbf{S}_A$.

Proof If $\mathbf{B}_A = \{R_0, R_U\}$ then the proposition is trivial. So we assume the case where there exists $R \in \mathbf{B}_A$ such that $R \neq R_0$, $R \neq R_U$ and $R \notin S_A$. In other words, there exists a vertex pair u and v such that $(u,v) \in R$ and $(E_1(u), E_1(v)) \notin R$. Then there exists a state configuration $c \in \mathbf{C}_R$ such that $c(E_1(u)) \neq c(E_1(v))$.

Since α is uniform and non autonomous, there must occur the case where for every $\varphi \in \Phi$, $\hat{\alpha}_\varphi(c)(u) = \alpha(c)(c(u), c(E_1(u)))$

$$\neq \alpha(c)(c(v), c(E_1(v))) = \hat{\alpha}_\varphi(c)(v)$$

Therefore $R \notin \mathbf{B}_A$. This contradicts to the assumption that $R \in \mathbf{B}_A$. So $\mathbf{B}_A = \mathbf{S}_A$. ∎

Proposition 5.8

There is an automata network $A = (G, \alpha)$ such that α is uniform, G is a graph where the indegree of every vertex is two, and $S_A = \mathbf{B}_A \neq \mathbf{R}_G$.

Proof

The graph is illustrated in Fig.1. $Q = \{0,1\}$, α is uniform and local map f is as follows:

$f = f(q,x_1,x_2)$ such that $f(0,0,1) = f(0,1,0) = f(1,0,1) = f(1,1,0) = 1$ and in the other cases the value of f is defined to be 0. ∎

6. Concluding Remarks

We have formulated the framework of the automata network and investigated some properties of two kinds of equivalence relation in the automata network.

From the Propositions 4.5, 5.1, 5.6, 5.7, and 5.8, we have the following conjecture: for every graph G, with some exceptions, where the indegree of every vertex is more than one, there exists a nonautonomous α such that $S_A \subsetneq B_A$. We are interested in this conjecture because it implies that we can choose some suitable α for every graph G and the automata network $A = (G,\alpha)$ can behave more "freely" than constrained by the interconnection relations (S_A).

We are also interested in the problem to get the conditions for that $S_A = \{R_O,R_U\}$, but we have not succeeded in solving it.