

## On Weakly Invertible Semi-input-memory Finite Automata with Delay 2

Renji Tao  
*Institute of Software  
Chinese Academy of Sciences  
Beijing 100080, China  
trj@ios.ac.cn*

*Abstract:* Semi-input-memory finite automata are a generalization of input-memory finite automata by appending an autonomous finite automaton component. This paper gives a decision criterion of weakly invertible semi-input-memory finite automata with delay 2 of which the minimal output weight of length 2 and the sizes of input and output alphabets are identical. The results are used to generate a kind of weakly invertible semi-input-memory finite automata with delay 2 and to give other proofs of results in binary case. In addition, a frame of binary ciphers with delay 2 and bounded propagation is presented.

**Keywords** finite automata, semi-input-memory, invertibility

### 1. Introduction

Finite automata are regarded as a natural mathematical model of cryptographic systems. This stimulates the investigation of invertibility of finite automata, which has been received extensive attentions since 1950s (see the references of [5]). Among others, in [3] we introduce the concept of semi-input-memory finite automata for characterizing the boundness of the error propagation in decoding. Such a finite automaton is called a feedforward inverse if it is a weak inverse. (cf. [5 page 13 and §1.5]) Due to the equivalence between boundness of the error propagation in decoding and feedforward invertibility [3], the simplicity of semi-input-memory finite automata relative to general finite automata causes investigating the structure of feedforward inverses for the investigation of the error propagation. But the problem of the structure of feedforward inverses is not trivial. There are systematic results on this topic only in the case of small delay.

In [4], a decision criterion of feedforward inverse finite automata is presented which is used to characterize the structure of feedforward inverse finite automata with delay 0 and 1 in [4, 1, 2, 10] and binary ones with delay 2 in [11].

In [6], a result on mutual invertibility is given: for any stronger connected finite automaton with the same sizes of the input and output alphabets, it is a feedforward inverse if and only if it is weakly invertible. (cf. [5 page 56, Theorem 2.2.2]) Based on mutual invertibility, [7, 8] give another characterization of the structure of automata mentioned in previous paragraph, and [9] studies the structure of binary feedforward

inverse finite automata with delay 3 in some case of the minimal output weight.

This paper deals with the general case of sizes of the input and output alphabets. We prove in Section 2 a new decision criterion of weakly invertible semi-input-memory finite automata with delay 2 of which the minimal output weight  $w_{2,M}$  and sizes of the input and output alphabets are identical. As an application of the decision criterion, Section 3 presents a method to generate a kinds of weakly invertible semi-input-memory finite automata with delay 2. In Section 4, results for general case in Section 2 are also used to give other proofs of results in binary case. And in Section 5, several open problems are presented. In addition, a frame of binary ciphers with delay 2 and bounded propagation is given in Appendix.

For terminology and notation in automata theory not explained in this paper, readers are referred to [5].

## 2. A decision criterion

Recall some definitions. A finite automaton is a quintuple  $\langle X, Y, S, \delta, \lambda \rangle$ , where  $X$ ,  $Y$  and  $S$  are nonempty finite sets,  $\delta$  and  $\lambda$  are single-valued mappings from  $S \times X$  to  $S$  and to  $Y$ , respectively. The domains of  $\delta$  and  $\lambda$  are expanded to  $S \times X^*$  by  $\delta(s, \varepsilon) = s$ ,  $\delta(s, \alpha x) = \delta(\delta(s, \alpha), x)$ ,  $\lambda(s, \varepsilon) = \varepsilon$ ,  $\lambda(s, \alpha x) = \lambda(s, \alpha)\lambda(\delta(s, \alpha), x)$ ,  $s \in S$ ,  $\alpha \in X^*$ ,  $x \in X$ , where  $X^*$  stands for the set of all words over  $X$  including the empty word  $\varepsilon$ . If for any  $s$  in  $S$ ,  $\delta(s, x)$  and  $\lambda(s, x)$  do not depend on  $x$ ,  $M$  is said to be autonomous and abbreviated to  $\langle Y, S, \delta, \lambda \rangle$ , where  $\delta$  and  $\lambda$  are single-valued mappings from  $S$  to  $S$  and to  $Y$ , respectively. Let  $M_a = \langle Y_a, S_a, \delta_a, \lambda_a \rangle$  be an autonomous finite automaton,  $f$  a single-valued mapping from  $X^{c+1} \times \lambda_a(S_a)$  to  $Y$ . We use  $\mathcal{STM}(M_a, f)$  to denote a finite automaton  $\langle X, Y, X^c \times S_a, \delta, \lambda \rangle$ , where

$$\begin{aligned} \delta(\langle x_{-1}, \dots, x_{-c}, s_a \rangle, x_0) &= \langle x_0, x_{-1}, \dots, x_{-c+1}, \delta_a(s_a) \rangle, \\ \lambda(\langle x_{-1}, \dots, x_{-c}, s_a \rangle, x_0) &= f(x_0, x_{-1}, \dots, x_{-c}, \lambda_a(s_a)), \\ x_0, x_{-1}, \dots, x_{-c} &\in X, \quad s_a \in S_a. \end{aligned}$$

$\mathcal{STM}(M_a, f)$  is referred to as a  $c$ -order semi-input-memory finite automaton determined by  $M_a$  and  $f$ .

Let  $M = \langle X, Y, S, \delta, \lambda \rangle$  be a finite automaton. Define  $W_{l,s}^M = \lambda(s, X^l) = \{\lambda(s, x_0 \dots x_{l-1}) \mid s \in S, x_0, \dots, x_{l-1} \in X\}$ ,  $w_{l,M} = \min_{s \in S} |W_{l,s}^M|$  and  $I_{\beta,s}^M = \{x_0 \dots x_{|\beta|-1} \mid \lambda(s, x_0 \dots x_{|\beta|-1}) = \beta, x_0, \dots, x_{|\beta|-1} \in X\}$ .

It is easy to see that  $M$  is weakly invertible with delay 2 if and only if for any  $x_0, x_1, x'_0, x'_1 \in X$  with  $x_0 \neq x'_0$ ,  $\lambda(s, x_0 x_1) = \lambda(s, x'_0 x'_1)$  derives  $W_{1,\delta(s, x_0 x_1)}^M \cap W_{1,\delta(s, x'_0 x'_1)}^M = \emptyset$ .

**Lemma 2.1.** *Assume that  $M$  is stronger connected and weakly invertible with delay 2. Let  $w_{2,M} = |Y| = |X|$ . If  $\lambda(s, x_1) = \dots = \lambda(s, x_r)$  holds for different  $x_i$ ,  $i = 1, \dots, r$  in  $X$ , then  $|\bigcup_{i=1}^r W_{1,\delta(s, x_i)}^M| \geq r$ .*

*Proof.* Suppose that  $\lambda(s, x_1) = \dots = \lambda(s, x_r)$  holds for different  $x_i$ ,  $i = 1, \dots, r$  in  $X$ . We prove  $|\bigcup_{i=1}^r W_{1,\delta(s, x_i)}^M| \geq r$  by reduction to absurdity. Suppose to the

contrary that  $|\bigcup_{i=1}^r W_{1,\delta(s,x_i)}^M| < r$ . Let  $\bigcup_{i=1}^r W_{1,\delta(s,x_i)}^M = \{y_1, \dots, y_t\}$  for some  $t < r$ . For any  $i, j$ ,  $1 \leq i \leq r, 1 \leq j \leq t$ , let  $|\bigcup_{x \in I_{y_j, \delta(s,x_i)}^M} W_{1,\delta(s,x_i)}^M| = c_{i,j}$ . Since  $M$  is stronger connected,  $w_{2,M} = |Y|$  derives  $|W_{2,\delta(s,x_i)}^M| = |Y|$ ,  $i = 1, \dots, r$ . Then we have  $\sum_{j=1}^t c_{i,j} = |Y|$ ,  $i = 1, \dots, r$ . Since  $M$  is weakly invertible with delay 2, for any  $j$ ,  $1 \leq j \leq t$ ,  $\bigcup_{x \in I_{y_j, \delta(s,x_i)}^M} W_{1,\delta(s,x_i)}^M$ ,  $i = 1, \dots, r$  are disjoint with each other. Thus  $\sum_{i=1}^r c_{i,j} = \sum_{i=1}^r |\bigcup_{x \in I_{y_j, \delta(s,x_i)}^M} W_{1,\delta(s,x_i)}^M| \leq |Y|$ ,  $j = 1, \dots, t$ . We then obtain  $r|Y| = \sum_{i=1}^r \sum_{j=1}^t c_{i,j} = \sum_{j=1}^t \sum_{i=1}^r c_{i,j} \leq t|Y|$ . It follows that  $r \leq t$ . This contradicts  $t < r$ . We conclude that  $|\bigcup_{i=1}^r W_{1,\delta(s,x_i)}^M| \geq r$ .  $\square$

**Lemma 2.2.** *Assume that  $M$  is stronger connected and weakly invertible with delay 2. Let  $w_{2,M} = |Y| = |X|$ . Then for any  $y \in W_{1,s}^M$ , we have  $|\bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M| = |I_{y,s}^M|$ .*

*Proof.* Since  $M$  is stronger connected,  $w_{2,M} = |Y|$  derives  $|W_{2,s}^M| = |Y| = |X|$ . Obviously,  $|W_{2,s}^M| = \sum_{y \in W_{1,s}^M} |\bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M|$ . From Lemma 2.1, for any  $y \in W_{1,s}^M$ , we have  $|\bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M| \geq |I_{y,s}^M|$ . Thus  $|X| = \sum_{y \in W_{1,s}^M} |I_{y,s}^M| \leq \sum_{y \in W_{1,s}^M} |\bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M| = |W_{2,s}^M| = |X|$ . It follows that  $\sum_{y \in W_{1,s}^M} |I_{y,s}^M| = \sum_{y \in W_{1,s}^M} |\bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M|$ . From  $|\bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M| \geq |I_{y,s}^M|$  for any  $y \in W_{1,s}^M$ , this derives  $|\bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M| = |I_{y,s}^M|$  for any  $y \in W_{1,s}^M$ .  $\square$

**Lemma 2.3.** *(equilibration) Assume that  $M$  is stronger connected and weakly invertible with delay 2. Let  $w_{2,M} = |Y| = |X|$ . Then for any  $y \in W_{1,s}^M$  and any  $y' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M$ ,  $\sum_{x \in I_{y',s}^M} |I_{y',\delta(s,x)}^M| = |X|$ .*

*Proof.* Let  $y \in W_{1,s}^M$  and  $I_{y,s}^M = \{x_1, \dots, x_r\}$ , where  $r = |I_{y,s}^M|$ . Let  $s_i = \delta(s, x_i)$ ,  $i = 1, \dots, r$ . Obviously,  $\bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M = \bigcup_{i=1}^r W_{1,s_i}^M$ , and  $\sum_{x \in I_{y,s}^M} |I_{y',\delta(s,x)}^M| = \sum_{i=1}^r |I_{y',s_i}^M|$ . From Lemma 2.2, we have  $|\bigcup_{i=1}^r W_{1,s_i}^M| = |\bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M| = |I_{y,s}^M| = r$ . Let  $\bigcup_{i=1}^r W_{1,s_i}^M = \{y_1, \dots, y_r\}$ .

We prove  $\sum_{i=1}^r |I_{y_j, s_i}^M| = |X|$ ,  $j = 1, \dots, r$ . Let  $d_{i,j} = |I_{y_j, s_i}^M|$ ,  $i, j = 1, \dots, r$ . We prove  $\sum_{i=1}^r d_{i,j} \leq |X|$ ,  $j = 1, \dots, r$  by reduction to absurdity. Suppose to the contrary that  $\sum_{i=1}^r d_{i,j} > |X|$  for some  $j$ ,  $1 \leq j \leq r$ . From Lemma 2.2, we have  $|\bigcup_{x \in I_{y_j, s_i}^M} W_{1,\delta(s_i,x)}^M| = |I_{y_j, s_i}^M| = d_{i,j}$ ,  $i, j = 1, \dots, r$ . Thus  $\sum_{i=1}^r |\bigcup_{x \in I_{y_j, s_i}^M} W_{1,\delta(s_i,x)}^M| = \sum_{i=1}^r d_{i,j} > |X|$ . This derives that  $\bigcup_{x \in I_{y_j, s_i}^M} W_{1,\delta(s_i,x)}^M$ ,  $i = 1, \dots, r$  are not disjoint with each other. Thus  $M$  is not weakly invertible with delay 2. This is a contradiction. We conclude  $\sum_{i=1}^r d_{i,j} \leq |X|$ ,  $j = 1, \dots, r$ .

We prove  $\sum_{i=1}^r d_{i,j} = |X|$ ,  $j = 1, \dots, r$  by reduction to absurdity. Suppose to the contrary that  $\sum_{i=1}^r d_{i,j'} \neq |X|$  for some  $j'$ ,  $1 \leq j' \leq r$ . It is easy to see that  $\sum_{j=1}^r d_{i,j} = |X|$ ,  $i = 1, \dots, r$ . Since  $\sum_{i=1}^r d_{i,j} \leq |X|$ ,  $j = 1, \dots, r$ , we have  $r|X| < \sum_{j=1}^r \sum_{i=1}^r d_{i,j} = \sum_{i=1}^r \sum_{j=1}^r d_{i,j} = \sum_{i=1}^r |X| = r|X|$ . It follows that  $r|X| < r|X|$ . This is a contradiction. We conclude  $\sum_{i=1}^r d_{i,j} = |X|$ ,  $j = 1, \dots, r$ .  $\square$

**Lemma 2.4.** (*compatibility and equilibration*) Assume that  $M$  is stronger connected and weakly invertible with delay 2. Let  $w_{2,M} = |Y| = |X|$ . Then (a) for any  $y \in W_{1,s}^M$  and any  $y' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M$ , the sets  $\bigcup_{x' \in I_{y',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M$ ,  $x \in I_{y,s}^M$  are disjoint with each other, and  $\sum_{x \in I_{y,s}^M} |\bigcup_{x' \in I_{y',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M| = |Y|$ , and (b) for any  $y \in W_{1,s}^M$  and any  $y'' \in Y$ , we have  $\sum_{x \in I_{y,s}^M, x' \in X} |I_{y'',\delta(s,xx')}^M| = |I_{y,s}^M||X|$ .

*Proof.* (a) Let  $y \in W_{1,s}^M$  and  $y' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M$ . Since  $M$  is weakly invertible with delay 2,  $\bigcup_{x' \in I_{y',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M$ ,  $x \in I_{y,s}^M$  are disjoint with each other. Thus  $\sum_{x \in I_{y,s}^M} |\bigcup_{x' \in I_{y',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M| \leq |Y|$ .

We prove  $\sum_{x \in I_{y,s}^M} |\bigcup_{x' \in I_{y',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M| = |Y|$  for any  $y' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M$  by reduction to absurdity. Suppose to the contrary that

$$\sum_{x \in I_{y,s}^M} \left| \bigcup_{x' \in I_{y',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M \right| \neq |Y|$$

holds for some  $\bar{y}' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M$ . Since  $\sum_{x \in I_{y,s}^M} |\bigcup_{x' \in I_{\bar{y}',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M| \leq |Y|$ , we have  $\sum_{x \in I_{y,s}^M} |\bigcup_{x' \in I_{\bar{y}',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M| < |Y|$ . Using Lemma 2.2, it follows that

$$\begin{aligned} & \sum_{y' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M} \sum_{x \in I_{y,s}^M} \left| \bigcup_{x' \in I_{y',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M \right| < \sum_{y' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M} |Y| \\ & = \left| \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M \right| |Y| = |I_{y,s}^M| |X|. \end{aligned} \quad (1)$$

On the other hand, using Lemma 2.2, we have

$$\begin{aligned} & \sum_{y' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M} \sum_{x \in I_{y,s}^M} \left| \bigcup_{x' \in I_{y',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M \right| \\ & = \sum_{x \in I_{y,s}^M} \sum_{y' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M} \left| \bigcup_{x' \in I_{y',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M \right| \\ & = \sum_{x \in I_{y,s}^M} \sum_{y' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M} |I_{y',\delta(s,x)}^M| = \sum_{x \in I_{y,s}^M} |X| = |I_{y,s}^M| |X|. \end{aligned} \quad (2)$$

From (1) and (2),  $|I_{y,s}^M| |X| < |I_{y,s}^M| |X|$  holds. This is a contradiction. We conclude that  $\sum_{x \in I_{y,s}^M} |\bigcup_{x' \in I_{y',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M| = |Y|$  holds for any  $y' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M$ .

(b) Let  $y \in W_{1,s}^M$  and  $x \in I_{y,s}^M$ . Using Lemma 2.3, for any  $y' \in W_{1,\delta(s,x)}^M$  and any  $y'' \in \bigcup_{x' \in I_{y',\delta(s,x)}^M} W_{1,\delta(s,xx')}^M$ ,  $\sum_{x' \in I_{y',\delta(s,x)}^M} |I_{y'',\delta(s,xx')}^M| = |X|$ . From (a), it follows that for any  $y' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M$  and any  $y'' \in Y$ ,  $\sum_{x \in I_{y,s}^M} \sum_{x' \in I_{y',\delta(s,x)}^M} |I_{y'',\delta(s,xx')}^M| = |X|$ . Thus, using Lemma 2.2, for any  $y'' \in Y$ ,  $\sum_{y' \in \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M} \sum_{x \in I_{y,s}^M} \sum_{x' \in I_{y',\delta(s,x)}^M} |I_{y'',\delta(s,xx')}^M| = \left| \bigcup_{x \in I_{y,s}^M} W_{1,\delta(s,x)}^M \right| |X| =$

$|I_{y,s}^M||X|$ . It follows that for any  $y'' \in Y$ ,

$$\begin{aligned}
\sum_{x \in I_{y,s}^M, x' \in X} |I_{y'', \delta(s, xx')}^M| &= \sum_{x \in I_{y,s}^M} \sum_{x' \in X} |I_{y'', \delta(s, xx')}^M| \\
&= \sum_{x \in I_{y,s}^M} \sum_{y' \in \bigcup_{x \in I_{y,s}^M} W_{1, \delta(s, x)}^M} \sum_{x' \in I_{y', \delta(s, x)}^M} |I_{y'', \delta(s, xx')}^M| \\
&= \sum_{y' \in \bigcup_{x \in I_{y,s}^M} W_{1, \delta(s, x)}^M} \sum_{x \in I_{y,s}^M} \sum_{x' \in I_{y', \delta(s, x)}^M} |I_{y'', \delta(s, xx')}^M| \\
&= |I_{y,s}^M||X|.
\end{aligned}$$

□

Below, let  $M = \langle X, Y, S, \delta, \lambda \rangle$  be a  $c$ -order semi-input-memory finite automaton  $\mathcal{SIM}(M_a, f)$ , where  $S = X^c \times S_a$ ,  $M_a = \langle Y_a, S_a, \delta_a, \lambda_a \rangle$  is an autonomous finite automaton, and  $f$  is a single-valued mapping from  $X^{c+1} \times \lambda_a(S_a)$  to  $Y$ .

For any state  $s$  of  $M$ , the input-partition of  $s$  means a partition of  $X$  so that  $x, x'$  belong to the same block if and only if  $\lambda(s, x) = \lambda(s, x')$ .

For any  $T \subseteq S$ , a output-partition of  $T$  means a partition of  $T$  so that  $|B| = |\lambda(B, X)|$  holds for each block  $B$ .

Recall that a partition  $\pi'$  of a set  $T$  is called a refinement of a partition  $\pi$  of  $T$ , if  $\pi'(t, t') \rightarrow \pi(t, t')$  is true for any  $t, t' \in T$ , that is, any block of  $\pi'$  is a subset of some block of  $\pi$ . Thus, any block of  $\pi$  is the union of some blocks of  $\pi'$ .

A output-partition  $\pi$  of  $T$  is called primitive, if any refinement of  $\pi$  other than  $\pi$  is not a output-partition.

Let  $B_1, \dots, B_r$  be all blocks of a partition.  $(|B_{i_1}|, |B_{i_2}|, \dots, |B_{i_r}|)$  is called the type of the partition, if  $|B_{i_1}| \leq |B_{i_2}| \leq \dots \leq |B_{i_r}|$ , for some permutation  $i_1, \dots, i_r$  of  $1, \dots, r$ . If a partition  $\pi'$  is a refinement of a partition  $\pi$ , the type of  $\pi'$  is called a refinement of the type of  $\pi$ .

For any state  $s$  of  $M$ , a join between  $s$  and the  $n$  states  $\delta(s, X)$  means a one-to-one mapping  $\varphi_s$  from the blocks of the input-partition of  $s$  onto the blocks of a output-partition of  $\delta(s, X)$  so that  $|\varphi_s(B)| = |B|$  holds for any block  $B$  of the input-partition of  $s$ . Clearly, types of the input-partition of  $s$  and the output-partition of  $\delta(s, X)$  are the same.

For any different states  $s_1, \dots, s_r$  of  $M$ , a join between the  $r$  states  $s_1, \dots, s_r$  and the  $r|X|$  states  $\delta(s_i, X)$ ,  $i = 1, \dots, r$  means  $r$  mappings  $\varphi_{s_i}$ ,  $i = 1, \dots, r$ , where  $\varphi_{s_i}$  is a join between  $s_i$  and  $\delta(s_i, X)$ ,  $i = 1, \dots, r$ . For any  $B \subseteq \{s_1, \dots, s_r\}$ , the  $|B|$  sets  $\delta(s', X)$ ,  $s' \in B$  are called compatible under the join, if for any  $y' \in \lambda(B, X)$ ,  $\lambda(\varphi_{s'}(I_{y', s'}^M), X)$ ,  $s' \in B$  with  $I_{y', s'}^M \neq \emptyset$  are a partition of  $Y$ .

$\delta$  is called a refinement of a join  $\varphi_s$ , if  $\delta(s, I_{y,s}^M) = \varphi_s(I_{y,s}^M)$  holds for any  $y \in \lambda(s, X)$ .

**Theorem 2.1.** Let  $M = \langle X, Y, S, \delta, \lambda \rangle$  be a  $c$ -order semi-input-memory finite automaton  $= \mathcal{SIM}(M_a, f)$  with  $|X| = |Y| = n$ . Assume that  $w_{2,M} = n$  and  $M_a$  is stronger cyclic. Then  $M$  is weakly invertible with delay 2 if and only if for any  $n$

states  $\delta(\bar{s}, X)$ , there is a set  $P_{\delta(\bar{s}, X)}$  of primitive output-partitions of  $\delta(\bar{s}, X)$  such that for any state  $s$  of  $M$  we can construct a join  $\varphi_s$  between  $s$  and  $\delta(s, X)$  satisfying the following conditions: (1) for any  $s \in S$ ,  $P_{\delta(s, X)}$  contains a refinement of the output-partition  $\varphi_s(I_{y, s}^M)$ ,  $y \in \lambda(s, X)$ , of  $\delta(s, X)$ ; (2) for any  $\bar{s} \in S$ , any output-partition  $\pi$  in  $P_{\delta(\bar{s}, X)}$  and any block  $B$  of  $\pi$ ,  $\delta(s', X)$ ,  $s' \in B$  are compatible under the join; (3)  $\delta$  is a refinement of the join  $\varphi_s$  for any  $s \in S$ .

*Proof. only if.* Suppose that  $M$  is weakly invertible with delay 2. For any  $n$  states  $\delta(\bar{s}, X)$ ,  $\bar{s}$  being a state of  $M$ , let  $s$  be a precursor of  $\delta(\bar{s}, X)$ . Then  $\delta(s, X) = \delta(\bar{s}, X)$ . It is easy to see that  $\delta(s, I_{y, s}^M)$ ,  $y \in \lambda(s, X)$  are a partition of  $\delta(\bar{s}, X)$ . Denote the partition by  $\pi_s$ . Since  $M$  is stronger connected, from Lemma 2.2, for any  $y \in W_{1, s}^M$ , we have  $|\lambda(\delta(s, I_{y, s}^M), X)| = |\bigcup_{x \in I_{y, s}^M} W_{1, \delta(s, x)}^M| = |I_{y, s}^M| = |\delta(s, I_{y, s}^M)|$ . Thus  $\pi_s$  is a output-partition. Let  $P'_{\delta(\bar{s}, X)} = \{\pi_s \mid s \text{ is a precursor of } \delta(\bar{s}, X)\}$ . Take  $P_{\delta(\bar{s}, X)}$  to be a least set of some primitive output-partitions of  $\delta(\bar{s}, X)$  which contains a refinement partition of each partition in  $P'_{\delta(\bar{s}, X)}$ .

For any state  $s$  of  $M$ , let  $\varphi_s$  be a mapping from the input-partition of  $s$  to the output-partition  $\pi_s$  of  $\delta(s, X)$  so that  $\varphi_s(I_{y, s}^M) = \delta(s, I_{y, s}^M)$  for any  $y \in \lambda(s, X)$ . Clearly,  $\varphi_s$  is one-to-one and onto, and  $|\varphi_s(I_{y, s}^M)| = |\delta(s, I_{y, s}^M)| = |I_{y, s}^M|$ . Thus  $\varphi_s$  is a join between  $s$  and  $\delta(s, X)$ . Since  $\pi_s \in P'_{\delta(s, X)}$ , from the construction of  $P_{\delta(s, X)}$ ,  $P_{\delta(s, X)}$  contains a refinement partition of  $\pi_s$ . Thus the condition (1) holds. From the definition,  $\delta$  is a refinement of the join. That is, the condition (3) holds. To prove the condition (2), for any output-partition  $\pi$  in  $P_{\delta(\bar{s}, X)}$ , from the construction of  $P_{\delta(\bar{s}, X)}$ , there is a precursor  $s$  of  $\delta(\bar{s}, X)$  such that  $\pi$  is a refinement of  $\pi_s$ . From Lemma 2.4 (a), for any  $y \in W_{1, s}^M$  and any  $y' \in \bigcup_{x \in I_{y, s}^M} W_{1, \delta(s, x)}^M$ , the sets  $\bigcup_{x' \in I_{y', \delta(s, x)}^M} W_{1, \delta(s, xx')}^M$ ,  $x \in I_{y, s}^M$  are disjoint with each other, and  $\sum_{x \in I_{y, s}^M} |\bigcup_{x' \in I_{y', \delta(s, x)}^M} W_{1, \delta(s, xx')}^M| = |Y|$ , that is, the sets  $\bigcup_{x' \in I_{y', \delta(s, x)}^M} W_{1, \delta(s, xx')}^M$ ,  $x \in I_{y, s}^M$  with  $I_{y', \delta(s, x)}^M \neq \emptyset$  are a partition of  $Y$ . Let  $B$  be a block of  $\pi_s$ . Then  $B = \delta(s, I_{y, s}^M)$  for some  $y \in W_{1, s}^M$ . Thus for any  $y' \in \bigcup_{x \in I_{y, s}^M} W_{1, \delta(s, x)}^M = \lambda(B, X)$ , the sets  $\bigcup_{x' \in I_{y', \delta(s, x)}^M} W_{1, \delta(s, xx')}^M$ ,  $x \in I_{y, s}^M$  with  $I_{y', \delta(s, x)}^M \neq \emptyset$ , i.e.,  $\lambda(\varphi_{s'}(I_{y', s'}^M), X)$ ,  $s' \in B$  with  $I_{y', s'}^M \neq \emptyset$  are a partition of  $Y$ . Assume that  $B_1, \dots, B_r$  are different blocks of  $\pi$  and  $B_1, \dots, B_r$  are a partition of  $B$ . Then  $|B| = |B_1| + \dots + |B_r|$ . Since  $\pi$  and  $\pi_s$  are output-partitions, we have  $|\lambda(B, X)| = |\lambda(B_1, X)| + \dots + |\lambda(B_r, X)|$ . It follows that  $\lambda(B_i, X), i = 1, \dots, r$  are a partition of  $\lambda(B, X)$ . Thus for any  $y' \in \lambda(B_i, X)$ , the sets  $\lambda(\varphi_{s'}(I_{y', s'}^M), X)$ ,  $s' \in B$  with  $I_{y', s'}^M \neq \emptyset$  are a partition of  $Y$ . Since  $I_{y', s'}^M = \emptyset$  for any  $y' \in \lambda(B_i, X)$  and any  $s' \in B \setminus B_i$ , we have that for any  $y' \in \lambda(B_i, X)$ , the sets  $\lambda(\varphi_{s'}(I_{y', s'}^M), X)$ ,  $s' \in B_i$  with  $I_{y', s'}^M \neq \emptyset$  are a partition of  $Y$ . Thus  $\delta(s', X)$ ,  $s' \in B_i$  are compatible under the join. Since any block of  $\pi$  is a subset of some block of  $\pi_s$ , for any block  $B'$  of  $\pi$ ,  $\delta(s', X)$ ,  $s' \in B'$  are compatible under the join. Therefore, the condition (2) holds.

*if.* Suppose that for any  $n$  states  $\delta(\bar{s}, X)$ , there is a set  $P_{\delta(\bar{s}, X)}$  of output-partitions of  $\delta(\bar{s}, X)$  such that for any state  $s$  of  $M$  we can construct a join  $\varphi_s$  between  $s$  and  $\delta(s, X)$  satisfying the conditions (1),(2) and (3). We prove by reduction to absurdity that  $M$  is weakly invertible with delay 2. Suppose to the

contrary that  $M$  is not weakly invertible with delay 2. Then there are  $s \in S$  and  $x_i, x'_i \in X$ ,  $i = 0, 1, 2$  such that  $x_0 \neq x'_0$  and  $\lambda(s, x_0 x_1 x_2) = \lambda(s, x'_0 x'_1 x'_2) = yy'y''$ , for some  $y, y', y''$  in  $Y$ . Let  $s_i = \delta(s, x_0 \dots x_{i-1})$  and  $s'_i = \delta(s, x'_0 \dots x'_{i-1})$ ,  $i = 1, 2$ . Since the condition (3) holds, we have  $\delta(s, I_{y,s}^M) = \varphi_s(I_{y,s}^M)$ , for any  $y \in \lambda(s, X)$ . From the condition (1), there is  $\pi$  in  $P_{\delta(s, X)}$  such that  $\pi$  is a refinement of  $\pi_s$ . Since the condition (2) holds, for the output-partition  $\pi$  in  $P_{\delta(s, X)}$  and any block  $B'$  of  $\pi$ ,  $\delta(s', X)$ ,  $s' \in B'$  are compatible under the join. Since  $x_0 \neq x'_0$ , we have  $s_1 \neq s'_1$ . From  $\lambda(s, x_0) = \lambda(s, x'_0) = y$ ,  $s_1$  and  $s'_1$  belong to the same block of  $\pi_s$ , say  $B$ . Since  $\pi$  is a refinement of  $\pi_s$ , there are blocks  $B_1, \dots, B_r$  of  $\pi$  such that  $B_1, \dots, B_r$  are a partition of  $B$ . As proven above,  $\lambda(B_i, X)$ ,  $i = 1, \dots, r$  are a partition of  $\lambda(B, X)$ . From  $s_1, s'_1 \in B$ , using  $\lambda(s_1, x_1) = y' = \lambda(s'_1, x'_1)$ , this derives  $s_1, s'_1 \in B_i$  for some  $i$ . Since  $\delta(s', X)$ ,  $s' \in B_i$  are compatible under the join and  $s_1, s'_1 \in B_i$ ,  $\lambda(\varphi_{s_1}(I_{y',s_1}^M), X)$  and  $\lambda(\varphi_{s'_1}(I_{y',s'_1}^M), X)$  are disjoint. Since the condition (3) holds, we have  $\delta(s_1, I_{y',s_1}^M) = \varphi_{s_1}(I_{y',s_1}^M)$  and  $\delta(s'_1, I_{y',s'_1}^M) = \varphi_{s'_1}(I_{y',s'_1}^M)$ . It follows that  $\lambda(\delta(s_1, I_{y',s_1}^M), X)$  and  $\lambda(\delta(s'_1, I_{y',s'_1}^M), X)$  are disjoint. Since  $s_2 \in \delta(s_1, I_{y',s_1}^M)$  and  $s'_2 \in \delta(s'_1, I_{y',s'_1}^M)$ ,  $\lambda(s_2, X)$  and  $\lambda(s'_2, X)$  are disjoint. Then we have  $\lambda(s_2, x_2) \neq \lambda(s'_2, x'_2)$ . This contradicts  $\lambda(s_2, x_2) = \lambda(s'_2, x'_2) = y''$ . We conclude that  $M$  is weakly invertible with delay 2.  $\square$

### 3. Application

**Lemma 3.1.** *For any positive  $m$  and  $n$  with  $m \leq n$  and any nonnegative integers  $c_{ij}$ ,  $i, j = 1, \dots, m$  satisfying  $\sum_{i=1}^m c_{ij} = \sum_{i=1}^m c_{ji} = n$ ,  $j = 1, \dots, m$ , there exists an  $m \times n$  matrix  $A$  over  $\{1, \dots, m\}$  such that each column of  $A$  is a permutation of  $1, \dots, m$  and the number of occurrences of  $j$  in row  $i$  of  $A$  is  $c_{ij}$ ,  $i, j = 1, \dots, m$ .*

*Proof.* We construct  $A$  recurrently. For any  $k$ ,  $1 \leq k \leq m$ , we use  $A_k$  to denote an  $m \times n$  matrix over  $\{0, 1, \dots, k\}$  satisfying the condition  $P(k)$ : for any  $h$ ,  $1 \leq h \leq k$ , each column of  $A_k$  has unique occurrence of  $h$  and any row  $i$  of  $A_k$  has  $c_{ih}$  occurrences of  $h$ . It is easy to see that such a  $A_1$  is existent. For example, in row  $i$  of  $A_1$ , elements in column  $c_{i1} + \dots + c_{i-1,1} + 1$  to column  $c_{i1} + \dots + c_{i1}$  are 1,  $i = 1, \dots, m$ , and elements are 0 elsewhere. From  $\sum_{i=1}^m c_{i1} = n$ , this is possible. Clearly,  $A_1$  satisfies  $P(1)$ .

Suppose that we have constructed  $A_k$ ,  $k < m$ . We construct  $A_{k+1}$  from  $A_k$  as follows. For each  $i$ ,  $1 \leq i \leq m$ , using  $\sum_{j=1}^m c_{ij} = n$ , we have  $c_{i,k+1} \leq n - c_{i1} - \dots - c_{ik}$ . Since the number of element 0 in row  $i$  of  $A_k$  is  $n - c_{i1} - \dots - c_{ik}$ , we can replace  $c_{i,k+1}$  elements 0 in row  $i$  by  $k + 1$ . From  $\sum_{i=1}^m c_{i,k+1} = n$ ,  $n$  elements 0 in  $A_k$  are replaced by  $k + 1$ . Denote the result matrix by  $B_0$ . Start off with  $B_0$ , we recurrently construct  $B_0, B_1, \dots$ . Suppose that we have constructed an  $m \times n$  matrix  $B_u$  over  $\{0, 1, \dots, k + 1\}$  satisfying the condition  $P'(k)$ : for any  $h$ ,  $1 \leq h \leq k$ , each column of  $B_u$  has unique occurrence of  $h$ , and for any  $h$ ,  $1 \leq h \leq k + 1$ , any row  $i$  of  $B_u$  has  $c_{ih}$  occurrences of  $h$ . Since  $A_k$  satisfies  $P(k)$ , from the construction of  $B_0$ ,  $B_0$  satisfies  $P'(k)$ . If each column of  $B_u$  has unique  $k + 1$ , then take  $A_{k+1} = B_u$ . Since

$B_u$  satisfies  $P'(k)$  and each column of  $B_u$  has unique  $k+1$ ,  $A_{k+1}(=B_u)$  satisfies  $P(k+1)$ . Otherwise, we construct an  $m \times n$  matrix  $B_{u+1}$  over  $\{0, 1, \dots, k+1\}$  such that  $B_{u+1}$  satisfies  $P'(k)$  and the number of columns without the element  $k+1$  of  $B_{u+1}$  equals the number of columns without the element  $k+1$  of  $B_u$  minus 1. Since the number of columns without the element  $k+1$  of  $B_0, B_1, \dots$  are strictly decreased, we arrive at some nonnegative integer  $u'$  so that the number of columns without the element  $k+1$  of  $B_{u'}$  equals 0. It follows that each column of  $B_{u'}$  has unique  $k+1$ . We then take  $A_{k+1} = B_{u'}$ .

We give a method to construct  $B_{u+1}$  from  $B_u$ . Suppose that one column of  $B_u$ , say column  $j$ , has not the element  $k+1$ . Since the number of occurrences of  $k+1$  in  $B_u$  is  $n$ , at least one column of  $B_u$ , say column  $j'$ , has at least two occurrences of  $k+1$ . Let the element at row  $r_0$  and column  $j'$  is  $k+1$ . Check the element at row  $r_0$  and column  $j$ . If it is 0, then stop. If it is  $a_1$  other than 0, then  $1 \leq a_1 \leq k$  and  $a_1$  occurs at row  $r_1$  and column  $j'$  for some  $r_1$  other than  $r_0$ . Analogously, check the element at row  $r_1$  and column  $j$ . If it is 0, then stop. If it is  $a_2$  other than 0, then  $1 \leq a_2 \leq k$  and  $a_2$  is different from  $a_1$  and occurs at row  $r_2$  and column  $j'$  for some  $r_2$  other than  $r_0, r_1$ . Continue this process, until stop. (It is easy to see that the process must stop.) Thus, we can find different rows  $r_0, \dots, r_t$  such that the element at row  $r_0$  and column  $j'$  is  $k+1$ , the element at row  $r_t$  and column  $j$  is 0, and for any  $p, 1 \leq p \leq t$ , the element at row  $r_p$  and column  $j'$  is identical with the element at row  $r_{p-1}$  and column  $j$ . Interchange the element at row  $r_p$  and column  $j'$  and the element at row  $r_p$  and column  $j$ ,  $p = 0, 1, \dots, t$ . Then the number of 0 decreases by one and the number of  $k+1$  increases by one in column  $j$ , the number of 0 increases by one and the number of  $k+1$  decreases by one in column  $j'$ , and column  $j'$  and column  $j$  keep the number of  $h$  unchanged for any  $h, 1 \leq h \leq k$ . We denote the result matrix by  $B_{u+1}$ . Clearly, column  $j$  of  $B_{u+1}$  has unique  $k+1$  and column  $j'$  of  $B_{u+1}$  has  $k+1$  still. Thus the number of columns without the element  $k+1$  of  $B_{u+1}$  equals the number of columns without the element  $k+1$  of  $B_u$  minus 1. It is easy to see that  $B_{u+1}$  also satisfies  $P'(k)$ .

Since  $A_m$  satisfies  $P(m)$ ,  $A_m$  is a matrix over  $\{1, \dots, m\}$ . Take  $A = A_m$ . Then each column of  $A$  is a permutation of  $1, \dots, m$  and the number of occurrences of  $j$  in row  $i$  of  $A$  is  $c_{ij}$ ,  $i, j = 1, \dots, m$ .  $\square$

We use some notations. The  $n$ -output of a state  $s$  means a sequence of length  $n$  consisting of  $\lambda(s, x)$ ,  $x \in X$ , that is,  $\lambda(s, x_1)\lambda(s, x_2) \dots \lambda(s, x_n)$ , where  $x_1, \dots, x_n$  are different elements of  $X$ . For example,  $y_1y_1y_1y_2y_2y_6$ , or  $y_1^3y_2^2y_6$  in short, is the  $n$ -output of  $s$  in case where  $n = 6, |I_{y_1, s}^M| = 3, |I_{y_2, s}^M| = 2$ , and  $|I_{y_6, s}^M| = 1$ . (Sometimes, the  $n$ -output  $y_1^3y_2^2y_6$  is denoted by  $(3y_1, 2y_2, 1y_6)$ .)

The  $n$ -outputs of the states  $s_{x_{-1}} = \langle x_{-1}, \dots, x_{-c}, s_a \rangle$ ,  $x_{-1} = 0, 1, \dots, n-1$  are called equilibrated with respect to a partition  $\pi$ , if for any block  $B$  of  $\pi$ ,  $\sum_{s \in B} |I_{y, s}^M| = n$  holds for any  $y$  in  $\lambda(B, X)$ .

A straightforward method to generate a part of weakly invertible semi-input-memory finite automata with delay 2:

1. Given alphabets  $X = \{0, 1, \dots, n-1\}$  and  $Y$  with  $|Y| = n$ , choose arbitrarily a stronger cyclic autonomous finite automaton  $M_a = \langle Y_a, S_a, \delta_a, \lambda_a \rangle$  with  $|S_a| > 2$ . Given a positive integer  $c \geq 2$ , take

$$S = \{\langle x_{-1}, \dots, x_{-c}, s_a \rangle \mid x_i \in X, i = -1, \dots, -c, s_a \in S_a\}.$$

2. (*initial choice*) Fix a state in  $S_a$ , say  $s_{a0}$ . For any  $n$  states  $s_{x_{-1}} = \langle x_{-1}, \dots, x_{-c}, s_{a0} \rangle, x_{-1} = 0, 1, \dots, n-1$ , take  $(1, 1, \dots, 1)$  as the type of a output-partition  $\pi_{s_0, \dots, s_{n-1}}$ , and take  $y_{s_i}^n$  as the  $n$ -output of  $s_i, i = 0, 1, \dots, n-1$ , where  $y_{s_i}, i = 0, 1, \dots, n-1$  are  $n$  different elements of  $Y$ .

3. (*recursion*) Suppose that for any  $x_{-1}, \dots, x_{-c}$  in  $X$ , the  $n$ -output of the state  $\langle x_{-1}, \dots, x_{-c}, \delta_a^k(s_{a0}) \rangle$  has been defined,  $k < |S_a| - 1$ , and that for any  $x_{-2}, \dots, x_{-c}$  in  $X$ , a output-partition  $\pi_{x_{-2}, \dots, x_{-c}}^k$  of  $s_{x_{-1}, \dots, x_{-c}}^k = \langle x_{-1}, \dots, x_{-c}, \delta_a^k(s_{a0}) \rangle, x_{-1} = 0, 1, \dots, n-1$  has been chosen, and the  $n$ -outputs of the states  $s_{x_{-1}, \dots, x_{-c}}^k, x_{-1} = 0, 1, \dots, n-1$  are equilibrated with respect to  $\pi_{x_{-2}, \dots, x_{-c}}^k$ .

For any  $x_{-3}, \dots, x_{-c}$  in  $X$ , consider the  $n^2$  states  $s_{x_{-1}, x_{-2}} = \langle x_{-1}, \dots, x_{-c}, \delta_a^{k+1}(s_{a0}) \rangle, x_{-1}, x_{-2} = 0, 1, \dots, n-1$ . For any  $x_{-2}$  in  $X$ , let  $s'_{x_{-2}, x_{-c-1}} = \langle x_{-2}, \dots, x_{-c-1}, \delta_a^k(s_{a0}) \rangle, x_{-c-1} = 0, 1, \dots, n-1$ . Choose a type, say  $t_{x_{-2}}$ , of a partition which is a refinement of the type of the input partition of  $s'_{x_{-2}, x_{-c-1}}$  for  $x_{-c-1} = 0, 1, \dots, n-1$ . Take arbitrarily a output-partition  $\pi_{x_{-2}, \dots, x_{-c}}^{k+1}$  of  $s_{x_{-1}, x_{-2}}, x_{-1} = 0, 1, \dots, n-1$  of which the type is  $t_{x_{-2}}$ . We use  $B_{x_{-2}, 1}, \dots, B_{x_{-2}, b_{x_{-2}}}$  to denote the blocks of  $\pi_{x_{-2}, \dots, x_{-c}}^{k+1}$ . For each  $x_{-c-1}$  in  $X$ , take a join between the state  $s'_{x_{-2}, x_{-c-1}}$  and the  $n$  states  $s_{x_{-1}, x_{-2}}, x_{-1} = 0, 1, \dots, n-1$ . That is, define a one-to-one mapping  $\varphi_{s'_{x_{-2}, x_{-c-1}}}$  from the blocks of the input partition of  $s'_{x_{-2}, x_{-c-1}}$  to the blocks of some partition  $\pi'_{x_{-2}}$  which keeps the size of any block unchanged, where  $\pi_{x_{-2}, \dots, x_{-c}}^{k+1}$  is a refinement of  $\pi'_{x_{-2}}$ . Choose the  $n$ -output of each state in each block so that the following three conditions are satisfied:

(1) (*output-partition and equilibration condition*) For any  $x_{-2}$  in  $X$  and any block  $B_{x_{-2}, i}, 1 \leq i \leq b_{x_{-2}}, |\lambda(B_{x_{-2}, i}, X)| = |B_{x_{-2}, i}|$  and  $\sum_{s \in B_{x_{-2}, i}} |I_{y, s}^M| = n$  for any  $y \in \lambda(B_{x_{-2}, i}, X)$ .

(2) (*refinement condition*) For any different blocks  $B_{x_{-2}, i}$  and  $B_{x_{-2}, j}$  which include in the same  $\varphi_{s'_{x_{-2}, x_{-c-1}}}(I)$  for some  $x_{-c-1}$  in  $X$  and some block  $I$  of the input partition of  $s'_{x_{-2}, x_{-c-1}}, \lambda(B_{x_{-2}, i}, X) \cap \lambda(B_{x_{-2}, j}, X) = \emptyset$  is true.

(3) (*compatibility condition*) For any  $x_{-c-1}$  in  $X$ , any block  $B$  of the output-partition  $\pi_{x_{-3}, \dots, x_{-c-1}}^k$  of the  $n$  states  $s'_{x_{-2}, x_{-c-1}}, x_{-2} = 0, 1, \dots, n-1$ , any  $y \in \lambda(B, X)$ , the set  $\lambda(\varphi_s(I_{y, s}^M), X), s \in B$  with nonempty  $I_{y, s}^M$  are a partition of  $Y$ .

4. (*closed*) Let  $x_{-3}, \dots, x_{-c}$  be in  $X$ . For any  $x_{-2}$  and  $x_{-c-1}$  in  $X$ , take a join between the state  $s'_{x_{-2}, x_{-c-1}} = \langle x_{-2}, \dots, x_{-c-1}, \delta_a^{|S_a|-1}(s_{a0}) \rangle$  and the  $n$  states  $\langle x_{-1}, \dots, x_{-c}, s_{a0} \rangle, x_{-1} = 0, 1, \dots, n-1$  so that the condition (3) mentioned above is satisfied. (From Lemma 3.1, such joins are existent.)

The following is an exemplification of step 3 for  $n = 5$ : Given  $x_{-3}, \dots, x_{-c}$  in  $X$ , suppose that for any  $x_{-2}$  and  $x_{-c-1}$  in  $X$ , the  $n$ -output of the state  $s'_{x_{-2}, x_{-c-1}} = \langle x_{-2}, \dots, x_{-c-1}, s_a \rangle$  has been defined as shown in Table 1, where  $s_a = \delta_a^k(s_{a0})$ ,

$k < |S_a| - 1$ . For any  $x_{-c-1}$  in  $X$ , an output-partition  $\pi_{x_{-3}, \dots, x_{-c-1}}^k$  of  $s'_{x_{-2}, x_{-c-1}}$ ,  $x_{-2} = 0, 1, \dots, 4$  has been chosen shown by underline or overline in Table 2. It is easy to verify that for any  $x_{-c-1}$ , the  $n$ -outputs of the states  $s'_{x_{-2}, x_{-c-1}}$ ,  $x_{-2} = 0, 1, \dots, 4$  are equilibrated with respect to  $\pi_{x_{-3}, \dots, x_{-c-1}}^k$ . Table 2 gives outputs of blocks of the output-partition  $\pi_{x_{-2}, \dots, x_{-c}}^{k+1}$  of 5 states  $s_{x_{-1}, x_{-2}} = \langle x_{-1}, \dots, x_{-c}, \delta_a(s_a) \rangle$ ,  $x_{-1} = 0, 1, \dots, 4$ . Using the result, we can choose  $n$ -outputs of  $s_{x_{-1}, x_{-2}}$ ,  $x_{-1} = 0, 1, \dots, 4$  to satisfy equilibration. For example, let  $\{s_{00}\}$ ,  $\{s_{10}\}$ ,  $\{s_{20}, s_{30}\}$ , and  $\{s_{40}\}$  be blocks of  $\pi_{0, x_{-3}, \dots, x_{-c}}^{k+1}$ . From row 1.4 column 1 of Table 2, the  $n$ -output of  $s_{00}$  is  $a^5$ , the  $n$ -output of  $s_{10}$  is  $b^5$ , the  $n$ -output of  $s_{40}$  is  $c^5$ , and take the  $n$ -output of  $s_{20}$  as  $d^r e^{5-r}$  and the  $n$ -output of  $s_{30}$  as  $d^{5-r} e^r$ , for some  $r$ ,  $0 \leq r \leq 5$ .

Table 2 gives some intermediate results also. Types in row 1.3 are chosen based on types of input-partitions, for example, in column 1, the 5 types of input-partition are (2,3), (2,3), (2,3), (1,1,3) and (1,2,2) in Table 1, of which type (1,1,1,2) in Table 2 is a refinement. A join  $\varphi_{s'_{x_{-2}, x_{-c-1}}}$  is given by listing  $\varphi_{s'_I}$  ( $I$ ) for each block  $I$  of the input-partition of  $s'_{x_{-2}, x_{-c-1}}$  which is a union set of some blocks of  $\pi_{x_{-2}, \dots, x_{-c}}^{k+1}$ , for example, in column 1,  $(1+1b, 1+2a)$  at row 2.0.1 means  $\varphi_{s'_{00}}(I_{b, s'_{00}}^M)$  is the union of the first two blocks with size 1 of  $\pi_{0, x_{-3}, \dots, x_{-c}}^{k+1}$ , and  $\varphi_{s'_{00}}(I_{a, s'_{00}}^M)$  is the union of other two blocks, with size 1 and 2 respectively, of  $\pi_{0, x_{-3}, \dots, x_{-c}}^{k+1}$ . The refinement condition and the compatibility condition are marked in row 2.h.2,  $h = 0, 1, \dots, 4$  (the same capital letter corresponds a partition of  $Y$ , subscripts are sizes of blocks); the result in row 1.4 satisfies such conditions.

Table 1:  $\{\langle i, j, *, \delta_a(s_a) \rangle, i = 0, 1, \dots, 4, j = 0, 1, \dots, 4$  and precursors

	1	2	3	4	5
1.1	$\langle i, 0, *, \delta_a(s_a) \rangle$ $i = 0, 1, 2, 3, 4$	$\langle i, 1, *, \delta_a(s_a) \rangle$ $i = 0, 1, 2, 3, 4$	$\langle i, 2, *, \delta_a(s_a) \rangle$ $i = 0, 1, 2, 3, 4$	$\langle i, 3, *, \delta_a(s_a) \rangle$ $i = 0, 1, 2, 3, 4$	$\langle i, 4, *, \delta_a(s_a) \rangle$ $i = 0, 1, 2, 3, 4$
1.2	$\diamond \diamond \diamond \diamond \diamond$	$\diamond \diamond \diamond \diamond \diamond$	$\diamond \diamond \diamond \diamond \diamond$	$\diamond \diamond \diamond \diamond \diamond$	$\diamond \diamond \diamond \diamond \diamond$
2.0.1	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\diamond$
2.0.2	$(2b, 3a)$	$(3b, 2a)$	$(1b, 4c)$	$(1c, 4d)$	$(1d, 4b)$
2.0.3	$\langle 0, *, 0, s_a \rangle$	$\langle 1, *, 0, s_a \rangle$	$\langle 2, *, 0, s_a \rangle$	$\langle 3, *, 0, s_a \rangle$	$\langle 4, *, 0, s_a \rangle$
2.1.1	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\diamond$
2.1.2	$(2b, 3a)$	$(3b, 2a)$	$(1b, 4c)$	$(1c, 4d)$	$(1d, 4b)$
2.1.3	$\langle 0, *, 1, s_a \rangle$	$\langle 1, *, 1, s_a \rangle$	$\langle 2, *, 1, s_a \rangle$	$\langle 3, *, 1, s_a \rangle$	$\langle 4, *, 1, s_a \rangle$
2.2.1	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\diamond$
2.2.2	$(2b, 3e)$	$(1b, 4c)$	$(3b, 2e)$	$(1c, 4d)$	$(1d, 4b)$
2.2.3	$\langle 0, *, 2, s_a \rangle$	$\langle 1, *, 2, s_a \rangle$	$\langle 2, *, 2, s_a \rangle$	$\langle 3, *, 2, s_a \rangle$	$\langle 4, *, 2, s_a \rangle$
2.3.1	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\diamond$
2.3.2	$(1a, 3b, 1c)$	$(1b, 3c, 1a)$	$(1c, 3a, 1b)$	$(1c, 4d)$	$(1d, 4c)$
2.3.3	$\langle 0, *, 3, s_a \rangle$	$\langle 1, *, 3, s_a \rangle$	$\langle 2, *, 3, s_a \rangle$	$\langle 3, *, 3, s_a \rangle$	$\langle 4, *, 3, s_a \rangle$
2.4.1	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\diamond$
2.4.2	$(2a, 2b, 1e)$	$(2a, 1b, 2e)$	$(1a, 2b, 2e)$	$(1c, 4d)$	$(1d, 4b)$
2.4.3	$\langle 0, *, 4, s_a \rangle$	$\langle 1, *, 4, s_a \rangle$	$\langle 2, *, 4, s_a \rangle$	$\langle 3, *, 4, s_a \rangle$	$\langle 4, *, 4, s_a \rangle$

$n = 5$ . Row 1.1 column  $j$ : states  $\{\langle i, j-1, *, \delta_a(s_a) \rangle, i = 0, 1, \dots, 4\}$ ,  $j = 1, \dots, 5$ ,  $*$  standing for the string  $x_{-3}, \dots, x_{-c}$ . Row 1.2 column  $j$ : 5 states,  $j = 1, \dots, 5$ . For  $h, 0 \leq h \leq 4$ , row 2.h.1 column  $j$ : 1 state,  $j = 1, \dots, 5$ . Row 2.h.2 column  $j$ :  $n$ -output of  $\langle j-1, *, h, s_a \rangle$ ,  $j = 1, \dots, 5$ . Row 2.h.3 column  $j$ : state  $\langle j-1, *, h, s_a \rangle$ ,  $j = 1, \dots, 5$ .

Table 2: recursion step

	1	2	3	4	5
1.1	$\langle i, 0, *, \delta_a(s_a) \rangle$ $i = 0, 1, 2, 3, 4$	$\langle i, 1, *, \delta_a(s_a) \rangle$ $i = 0, 1, 2, 3, 4$	$\langle i, 2, *, \delta_a(s_a) \rangle$ $i = 0, 1, 2, 3, 4$	$\langle i, 3, *, \delta_a(s_a) \rangle$ $i = 0, 1, 2, 3, 4$	$\langle i, 4, *, \delta_a(s_a) \rangle$ $i = 0, 1, 2, 3, 4$
1.2	$\diamond \diamond \diamond \diamond \diamond$	$\diamond \diamond \diamond \diamond \diamond$	$\diamond \diamond \diamond \diamond \diamond$	$\diamond \diamond \diamond \diamond \diamond$	$\diamond \diamond \diamond \diamond \diamond$
1.3	(1, 1, 1, 2)	(1, 1, 1, 2)	(1, 1, 1, 2)	(1, 4)	(1, 4)
1.4	$(a, b, c', d' e')$	$(a, b, c, de)$	$(a, b, c, de)$	$(a, bcde)$	$(a, bcde)$
2	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\diamond$
2.0.1	$(1 + 1b, 1 + 2a)$	$(1 + 1a, 1 + 2b)$	$(1b, 1 + 1 + 2c)$	$(1c, 4d)$	$(1d, 4b)$
2.0.2	$(A1, A1, B1, B2)$	$(B1, B1, A1, A2)$	$(C1, D1, D1, D2)$	$(D1, E4)$	$(E1, C4)$
2.1.1	$(1 + 1b, 1 + 2a)$	$(1 + 1a, 1 + 2b)$	$(1b, 1 + 1 + 2c)$	$(1c, 4d)$	$(1d, 4b)$
2.2.1	$(1 + 1b, 1 + 2e)$	$(1b, 1 + 1 + 2c)$	$(1 + 1e, 1 + 2b)$	$(1c, 4d)$	$(1d, 4b)$
2.2.2	$(A1, A1, B1, B2)$	$(C1, D1, D1, D2)$	$(B1, B1, A1, A2)$	$(D1, E4)$	$(E1, C4)$
2.3.1	$(1a, 1c, 1 + 2b)$	$(1b, 1a, 1 + 2c)$	$(1c, 1b, 1 + 2a)$	$(1c, 4d)$	$(1d, 4c)$
2.3.2	$(A1, B1, C1, C2)$	$(C1, A1, B1, B2)$	$(B1, C1, A1, A2)$	$(D1, E4)$	$(E1, D4)$
2.4.1	$(1 + 1b, 1e, 2a)$	$(1 + 1a, 1b, 2e)$	$(1a, 1 + 1b, 2e)$	$(1c, 4d)$	$(1d, 4c)$
2.4.2	$(A1, A1, B1, C2)$	$(C1, C1, A1, B2)$	$(C1, A1, A1, B2)$	$(D1, E4)$	$(E1, D4)$

Row 1.1 column  $j$ : states  $\{s_{i,j-1} = \langle i, j-1, *, \delta_a(s_a) \rangle, i = 0, 1, \dots, 4\}$ ,  $j = 1, \dots, 5$ . Row 1.2 column  $j$ : 5 states,  $j = 1, \dots, 5$ . Row 1.3 column  $j$ : type  $t_{j-1}$  of  $\pi_{j-1, x_{-3}, \dots, x_{-c}}^{k+1}$ ,  $j = 1, \dots, 5$ . Row 1.4 column  $j$ : outputs of blocks of the output-partition  $\pi_{j-1, x_{-3}, \dots, x_{-c}}^{k+1}$  of 5 states  $s_{i,j-1}$ ,  $i = 0, 1, \dots, 4$ ;  $j = 1, \dots, 5$ . ( $\{a, b, c, d, e\} = Y$ ,  $\{c, d, e\} = \{c', d', e'\}$ .) Row 2 column  $j$ :  $\diamond$  represents state  $\langle j-1, *, h, s_a \rangle$ ,  $j = 1, \dots, 5$ . For  $h, 0 \leq h \leq 4$ , row 2.h.1: + for merging blocks of the refinement partitions in row 1.3;  $a, b$ , etc. for 5 inputs with the same output; underline etc. for partition  $\pi_{x_{-3}, \dots, x_{-c}, h}^k$ . Row 2.h.2:  $A, B$ , etc. for outputs of blocks of the partitions in row 1.3, outputs with the same capital letter are a partition of  $Y$  (see row 1.4).

#### 4. Comment on binary case

In Section 5.4 in [5], an explicit expression for binary weakly invertible semi-input-memory finite automata with delay 2 had been presented. It may be deduced based on results in Section 2 of this paper. In this section,  $\bar{a}$  stands for  $a \oplus 1$ .

For a  $c$ -order semi-input-memory finite automaton  $M = \langle X, Y, S, \delta, \lambda \rangle = SIM(M_a, f)$  with stronger cyclic  $M_a$ , let  $n = 2$  and  $X = Y = \{0, 1\}$ . Assume that  $M$  is weakly invertible with delay 2 and  $w_{2,M} = n$ . Let  $s_{x_{-1}} = \langle x_{-1}, \dots, x_{-c}, s_a \rangle$ ,  $x_{-1} = 0, 1$ . Then the  $n$ -output of  $s_0$  is 01 if and only if the  $n$ -output of  $s_1$  is 01. (Suppose to the contrary that the  $n$ -outputs of  $s_0$  and  $s_1$  are  $\bar{a}b$  and  $ab$ . From Lemma 2.2,

the  $n$ -output of a precursor of  $s_0$  and  $s_1$  is  $cc$ . This contradicts Lemma 2.3.) Thus only one of two primitive output-partitions of states  $s_0$  and  $s_1$  can be taken, with type (2) or type (1,1), and the  $n$ -outputs of  $s_0$  and  $s_1$  are in form  $a\bar{a}$  and  $b\bar{b}$  in case of type (2) or  $aa$  and  $bb$  in case of type (1,1). Let  $s'_{x_{-1}} = \langle x_{-1}, \bar{x}_{-2}, x_{-3}, \dots, x_{-c}, s_a \rangle$ ,  $x_{-1} = 0, 1$ . Then the  $n$ -outputs of  $s_0$  and  $s_1$  are  $yy$  and  $\bar{y}\bar{y}$  if and only if the  $n$ -outputs of  $s'_0$  and  $s'_1$  are  $\bar{y}\bar{y}$  and  $y\bar{y}$ . (Suppose that the  $n$ -outputs of  $s_0$  and  $s_1$  are  $yy$  and  $\bar{y}\bar{y}$ . Let  $s$  and  $s'$  be two successors of a state  $t$  so that  $s_0$  and  $s_1$  are two successors of  $s$  and  $s'_0$  and  $s'_1$  are two successors of  $s'$ . From Lemma 2.2, the  $n$ -output of  $s$  is  $a\bar{a}$ , therefore, the  $n$ -output of  $s'$  is  $cc$  and the  $n$ -output of  $t$  is  $cc$ . Using Lemma 2.4 (b), the  $n$ -outputs of  $s'_0$  and  $s'_1$  are  $\bar{y}\bar{y}$  and  $y\bar{y}$ .) Letting  $s_{x_0, x_{-1}} = \langle x_0, x_{-1}, \dots, x_{-c+1}, \delta_a(s_a) \rangle$ , from Lemma 2.2, that the type of primitive output-partition of  $s_0$  and  $s_1$  is (2) derives that the type of primitive output-partition of  $s_{0i}$  and  $s_{1i}$  is (1,1),  $i = 0, 1$ . Therefore, the types of primitive output-partitions and  $n$ -outputs of  $s_0, s_1$  and  $s_{00}, s_{10}$  and  $s_{01}, s_{11}$  fall under one of the four forms described in columns 2 and 3 of Table 3. Thus, using Lemma 2.4 (a) for the last row,  $f$  satisfies relations described in column 4 of Table 3. Define  $h_0$  and  $h_1$  by that  $h_0(x_{-2}, \dots, x_{-c}, s_a) = 1 \Leftrightarrow$  the type of primitive output-partition of  $s_0$  and  $s_1$  is (1,1), and  $h_1(x_{-3}, \dots, x_{-c}, s_a) = 1 \Leftrightarrow f(x_0, x_{-1}, 0, x_{-3}, \dots, x_{-c}, s_a)$  does not depend on  $x_0$  and  $x_{-1}$ . Notice that  $f(x_0, x_{-1}, 0, x_{-3}, \dots, x_{-c}, s_a)$  does not depend on  $x_0$  and  $x_{-1}$  if and only if  $f(x_0, x_{-1}, 1, x_{-3}, \dots, x_{-c}, s_a)$  does not depend on  $x_0$  and  $x_{-1}$ . From Lemma 2.2, we have  $h_0(x_{-2}, \dots, x_{-c}, s_a) = 0 \rightarrow h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) = 1$  and  $h_1(x_{-3}, \dots, x_{-c}, s_a) = 1 \rightarrow h_0(x_{-3}, \dots, x_{-c-1}, \delta_a^{-1}(s_a)) = 0$ . This concludes a proof of Lemma 5.4.5 in [5 page 169], a key result there.

Table 3: deducing explicit expression of  $f$ 

1	2	3	4
states	type(s) of	$n$ -outputs of	$f(x_0, x_{-1}, \dots, x_{-c}, s_a) =$
$s_0, s_1$ $(s'_0, s'_1)$	(1, 1)	$yy, \bar{y}\bar{y}$ $\bar{y}\bar{y}, y\bar{y}$	$f(0, 0, 0, x_{-3}, \dots, x_{-c}, s_a) \oplus x_{-2}$
$s_0, s_1$	(1, 1)	$yy, \bar{y}\bar{y}$	$f(0, 0, x_{-2}, \dots, x_{-c}, s_a) \oplus x_{-1}$
$s_{00}, s_{10}; s_{01}, s_{11}$ $s_0 ; s_1$	(1, 1); (1, 1) (2)	$yy, \bar{y}\bar{y}; \bar{y}\bar{y}, y\bar{y}$ $a\bar{a} ; b\bar{b}$	$f(0, x_{-1}, \dots, x_{-c}, s_a) \oplus x_0$
$s_{00}, s_{10}; s_{01}, s_{11}$ $s_0 ; s_1$	(1, 1); (1, 1) (2)	$yy, \bar{y}\bar{y}; zz, \bar{z}\bar{z}$ $a\bar{a} ; b\bar{b}$	$f(0, 0, x_{-2}, \dots, x_{-c}, s_a) \oplus x_0 \oplus x_{-1} \oplus x_{-1}h_2(\cdot),$ $h_2(\cdot) = \sum_{x_{-1}=0}^1 f(0, 0, x_{-1}, \dots, x_{-c+1}, \delta_a(s_a))$

The above proof only using 3 lemmas. It is easy to modify the proof by only using Theorem 2.1.

## 5. Conclusion

We have given a decision criterion of weakly invertible semi-input-memory finite automata with delay 2 of which the minimal output weight of length 2 and the sizes of input and output alphabets are identical.

The decision criterion is based on leaves of state trees with level 2 instead of the trees themselves. Using the decision criterion, a straightforward method to generate a kind of weakly invertible semi-input-memory finite automata with delay 2 is presented. But it is still a unsolved problem to generate all weakly invertible semi-input-memory finite automata with delay 2, of which the minimal output weight of

length 2 and the sizes of input and output alphabets are identical. The bottleneck is a combinatorial problem:

Let  $n$ ,  $c$  and  $e$  be integers greater than 1. Put  $n$  balls of  $n^{c+1}e$  colored balls,  $n^c e$  balls with color  $y$  for  $y = 0, 1, \dots, n-1$ , to each box of  $n^c e$  boxes labelled by elements in  $\{0, 1, \dots, n-1\}^c \times \{0, 1, \dots, e-1\}$  satisfying the following conditions: 1. for any color  $y$ , the number of balls with color  $y$  in the  $n$  boxes labelled by  $\langle j, x_{-2}, \dots, x_{-c}, i \rangle$ ,  $j = 0, 1, \dots, n-1$  is a multiple of  $n$ ; 2. for any color  $y$ , the number of balls with color  $y$  in the  $n^2$  boxes labelled by  $\langle k, j, x_{-3}, \dots, x_{-c}, i \rangle$ ,  $j, k = 0, 1, \dots, n-1$  equals  $n^2$ ; 3. for any  $n$  boxes labelled by  $\langle j, x_{-2}, \dots, x_{-c}, i \rangle$ ,  $j = 0, 1, \dots, n-1$ , there exists a nonempty set  $P_{x_{-2}, \dots, x_{-c}, i}$  of primitive color-partitions of them, (a color-partition is a partition so that the number of different colors of balls in boxes in each block equals the number of boxes in the block, it is called primitive if any proper refinement partition is not a color-partition,)  $x_{-2}, \dots, x_{-c} \in \{0, 1, \dots, n-1\}$  and  $i \in \{0, 1, \dots, e-1\}$ , such that for any  $x_{-1}, \dots, x_{-c} \in \{0, 1, \dots, n-1\}$  and  $i \in \{0, 1, \dots, e-1\}$ , there exists a one-to-one mapping  $\varphi_{x_{-1}, \dots, x_{-c}, i}$  from balls in box labelled by  $\langle x_{-1}, \dots, x_{-c}, i \rangle$  to  $n$  boxes labelled by  $\langle j, x_{-1}, \dots, x_{-c+1}, i+1 \bmod e \rangle$ ,  $j = 0, 1, \dots, n-1$  satisfying the following two conditions: 3.1. for any  $x_{-1}, \dots, x_{-c} \in \{0, 1, \dots, n-1\}$  and  $i \in \{0, 1, \dots, e-1\}$ , the partition  $\varphi_{x_{-1}, \dots, x_{-c}, i}(B_y), y \in \{0, 1, \dots, n-1\}$  with  $B_y \neq \emptyset$  has a refinement partition in  $P_{x_{-1}, \dots, x_{-c+1}, i+1 \bmod e}$ , where  $B_y$  stands for the set of all balls with color  $y$  in the box labelled by  $\langle x_{-1}, \dots, x_{-c}, i \rangle$ , and 3.2. for any  $x_{-2}, \dots, x_{-c} \in \{0, 1, \dots, n-1\}, i \in \{0, 1, \dots, e-1\}$  and any  $\pi \in P_{x_{-2}, \dots, x_{-c}, i}$ , for each block  $B$  of  $\pi$  and each color  $y$ , we have  $C_b \cap C_{b'} = \emptyset$  for different  $b$  and  $b'$  in  $B$  and  $\cup_{b \in B} C_b = \{0, 1, \dots, n-1\}$ , where  $C_b$  stands for the set of all colors of balls in boxes mapped from balls with color  $y$  in box  $b$  by  $\varphi_{j, x_{-2}, \dots, x_{-c}, i}$ , the box  $b$  is labelled by  $\langle j, x_{-2}, \dots, x_{-c}, i \rangle$ .

Notice that the conditions 1 and 2 can be derived by the condition 3. In the case of  $n \leq 5$ , any  $P_{x_{-2}, \dots, x_{-c}, i}$  can be confined to a singleton set as color distributions of blocks of different primitive color-partitions are the same.

Since  $w_{2,M}$  is a divisor of  $|X|^2$  and the problem is trivial in the cases of  $w_{2,M} = |X|^2$  or 1, for a prime  $|X|$  a decision criterion in the case of any minimal output weight of length 2 is solved. But for a composite  $|X|$ , for example,  $|X| = 16$  or 258, no analogous decision criterion in the case of any minimal output weight of length 2 is known so far.

From the viewpoint of application to cryptography, another important unsolved problem is to find out an algebraic-logical expression for automata mentioned above and for their weak inverses. (Appendix gives a binary cipher from algebraic-logical expressions for automata concerned.)

## References

1. F. Bao, *On the Structure of  $n$ -ary Feedforward Inverses with Delay 1*, MA Thesis, Institute of Software, Chinese Academy of Sciences, Beijing, 1986. (in Chinese)
2. F. Bao, Limited error-propagation, self-synchronization and finite input-memory FSMs as weak inverses, in *Advances in Chinese Computer Science*, Vol. 3, World Scientific, Singapore, 1991, 1–24.
3. R. J. Tao, Relationship between bounded error propagation and feedforward invertibility, *Kexue Tongbao*, **27**(1982), 680–682.
4. R. J. Tao, Some results on the structure of feedforward inverses, *Scientia Sinica*, Ser. A, **27**(1984), 157–162.
5. R. J. Tao, *Finite automata and application to cryptography*, TUP, Beijing & Springer, Berlin, 2008.
6. R. J. Tao and S. H. Chen, Input-trees of finite automata and application to cryptanalysis, *Journal of Computer Science and Technology*, **15**(2000), 305–325.
7. R. J. Tao and S. H. Chen, Structure of weakly invertible semi-input-memory finite automata with delay 1, *Journal of Computer Science and Technology*, **17**(2002), 369–376.
8. R. J. Tao and S. H. Chen, Structure of weakly invertible semi-input-memory finite automata with delay 2, *Journal of Computer Science and Technology*, **17**(2002), 682–688.
9. H. J. Wang, *The Structures and Decomposition of the Finite Automata with Invertibility*, Ph. D. Thesis, Institute of Software, Chinese Academy of Sciences, Beijing, 2005. (in Chinese)
10. G. Yao, Two results on structure of feedforward inverse finite automata, *Journal of Software*, **13**(2002), supplement, 252–258. (in Chinese)
11. X. J. Zhu, On the structure of binary feedforward inverses with delay 2, *Journal of Computer Science and Technology*, **4**(1989), 163–171.

## Appendix

### A frame of binary ciphers with delay 2 and bounded error propagation

We can give another proof of the reverse of Lemma 5.4.5 in [5 page 169] by deduce an explicit expression of a weak inverse of  $M$ . The reverse of Lemma 5.4.5 is the following:

**Lemma 5.1.** *Let  $M = \langle X, Y, S, \delta, \lambda \rangle$  be a  $c$ -order semi-input-memory finite automaton  $SIM(M_a, f)$ , and  $M_a = \langle Y_a, S_a, \delta_a, \lambda_a \rangle$  be strongly cyclic. If  $c \geq 2 = w_{2,M}$ ,  $X = Y = \{0, 1\}$  and there exist single-valued mappings  $h_0$  from  $X^{c-1} \times S_a$  to  $\{0, 1\}$ ,  $h_1$  from  $X^{c-2} \times S_a$  to  $\{0, 1\}$ ,  $f_0$  from  $X^c \times S_a$  to  $Y$ ,  $f_1$  from  $X^{c-1} \times S_a$  to  $Y$ , and  $f_2$  from  $X^{c-2} \times S_a$  to  $Y$ , such that*

$$\begin{aligned} h_0(x_{-2}, \dots, x_{-c}, s_a) = 0 &\rightarrow h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) = 1, \\ h_1(x_{-3}, \dots, x_{-c}, s_a) = 1 &\rightarrow h_0(x_{-3}, \dots, x_{-c-1}, \delta_a^{-1}(s_a)) = 0, \end{aligned} \quad (3)$$

and

$$f(x_0, \dots, x_{-c}, s_a) = \begin{cases} f_2(x_{-3}, \dots, x_{-c}, s_a) \oplus x_{-2}, & \text{if } h_0(x_{-2}, \dots, x_{-c}, s_a) = 1 \ \& \ h_1(x_{-3}, \dots, x_{-c}, s_a) = 1, \\ f_1(x_{-2}, \dots, x_{-c}, s_a) \oplus x_{-1}, & \text{if } h_0(x_{-2}, \dots, x_{-c}, s_a) = 1 \ \& \ h_1(x_{-3}, \dots, x_{-c}, s_a) = 0, \\ f_0(x_{-1}, \dots, x_{-c}, s_a) \oplus x_0, & \text{if } h_0(x_{-2}, \dots, x_{-c}, s_a) = 0 \ \& \ h_1(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a)) = 1, \\ f_1(x_{-2}, \dots, x_{-c}, s_a) \oplus x_0 \oplus x_{-1} \oplus x_{-1}h_2(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a)), & \text{if } h_0(x_{-2}, \dots, x_{-c}, s_a) = 0 \ \& \ h_1(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a)) = 0, \end{cases}$$

then  $M$  is weakly invertible with delay 2, where

$$h_2(x_{-3}, \dots, x_{-c}, s_a) = f_1(0, x_{-3}, \dots, x_{-c}, s_a) \oplus f_1(1, x_{-3}, \dots, x_{-c}, s_a).$$

A proof is given in [5 page 173]. We modify the proof to give a weak inverse of  $M$  as follows. For any state  $s = \langle x_{-1}, x_{-2}, \dots, x_{-c}, s_a \rangle$  and inputs  $x_0, x_1, x_2$  in  $X$ , let  $y_0 y_1 y_2 = \lambda(s, x_0 x_1 x_2)$ . There are several cases to consider.

In the case of  $h_0(x_{-2}, \dots, x_{-c}, s_a) = 0$ , we have  $y_0 = f'_0(x_{-1}, \dots, x_{-c}, s_a) \oplus x_0$  for any  $x_0 \in X$ , where  $f'_0(x_{-1}, \dots, x_{-c}, s_a) = f_0(x_{-1}, \dots, x_{-c}, s_a)$  if  $h_1(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a)) = 1$ , or  $f_1(x_{-2}, \dots, x_{-c}, s_a) \oplus x_{-1} \oplus x_{-1}h_2(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a))$  if  $h_1(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a)) = 0$ . Thus  $x_0 = f'_0(x_{-1}, \dots, x_{-c}, s_a) \oplus y_0$ .

In the case of  $h_0(x_{-2}, \dots, x_{-c}, s_a) = 1$ , we have  $h_1(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a)) = 0$ . We further consider  $h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a))$  and  $h_1(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a))$ .

In the subcase of  $h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) = 1$ , since  $h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) = 1 \ \& \ h_1(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a)) = 0$ , we have  $y_1 = f_1(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) \oplus x_0$  for any  $x_0 \in X$ . Thus  $x_0 = f_1(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) \oplus y_1$ .

In the subcase of  $h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) = 0 \ \& \ h_1(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) = 1$ ,  $h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) = 0$  derives  $h_0(x_0, \dots, x_{-c+2}, \delta_a^2(s_a)) = 1$  for any  $x_0 \in X$ . Since  $h_0(x_0, \dots, x_{-c+2}, \delta_a^2(s_a)) = 1 \ \& \ h_1(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) = 1$ , we have  $y_2 = f_2(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus x_0$  for any  $x_0 \in X$ . Thus  $x_0 = f_2(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus y_2$ .

In the subcase of  $h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) = 0 \ \& \ h_1(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) = 0$ ,  $h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) = 0$  derives  $h_0(x_0, \dots, x_{-c+2}, \delta_a^2(s_a)) = 1$  for any  $x_0 \in X$ . For any  $x_0 \in X$ , since  $h_0(x_0, \dots, x_{-c+2}, \delta_a^2(s_a)) = 1 \ \& \ h_1(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) = 0$ , we have  $y_2 = f_1(x_0, x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus x_1$ . Since  $h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) = 0 \ \& \ h_1(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) = 0$ , we have

$$\begin{aligned} y_1 &= f_1(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) \oplus x_1 \oplus x_0 \oplus x_0 h_2(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \\ &= f_1(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) \oplus x_1 \oplus x_0 \oplus \\ &\quad x_0(f_1(0, x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus f_1(1, x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a))). \end{aligned}$$

It follows that

$$\begin{aligned} y_2 \oplus y_1 &= f_1(x_0, x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus x_1 \oplus f_1(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) \oplus x_1 \oplus \\ &\quad x_0 \oplus x_0(f_1(0, x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus f_1(1, x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a))) \\ &= f_1(0, x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus f_1(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) \oplus x_0. \end{aligned}$$

Thus  $x_0 = f_1(0, x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus f_1(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) \oplus y_2 \oplus y_1$ .

Define a mapping  $f'$  from  $Y^3 \times X^c \times S_a$  to  $X$  by

$$f'(y'_0, y'_{-1}, y'_{-2}, x_{-1}, \dots, x_{-c}, s_a) = \begin{cases} f_0(x_{-1}, \dots, x_{-c}, s_a) \oplus y'_{-2}, & \text{if } h_0(x_{-2}, \dots, x_{-c}, s_a) = 0 \ \& \ h_1(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a)) = 1, \\ f_1(x_{-2}, \dots, x_{-c}, s_a) \oplus x_{-1} \oplus x_{-1}h_2(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a)) \oplus y'_{-2}, & \text{if } h_0(x_{-2}, \dots, x_{-c}, s_a) = 0 \ \& \ h_1(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a)) = 0, \\ f_1(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) \oplus y'_{-1}, & \text{if } h_0(x_{-2}, \dots, x_{-c}, s_a) = 1 \ \& \ h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) = 1, \\ f_2(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus y'_0, & \text{if } h_0(x_{-2}, \dots, x_{-c}, s_a) = 1 \ \& \ h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) = 0 \\ & \ \& \ h_1(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) = 1, \\ f_1(0, x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus f_1(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) \oplus y'_0 \oplus y'_{-1}, & \text{if } h_0(x_{-2}, \dots, x_{-c}, s_a) = 1 \ \& \ h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) = 0 \\ & \ \& \ h_1(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) = 0. \end{cases}$$

Then we have  $x_0 = f'(y_2, y_1, y_0, x_{-1}, \dots, x_{-c}, s_a)$ . We conclude that  $M$  is weakly invertible with delay 2.

Let  $M' = \langle Y, X, Y^2 \times X^c \times S_a \times \{0, 1, 2\}, \delta', \lambda' \rangle$  be a finite automaton with  $X = Y = \{0, 1\}$  defined by

$$\begin{aligned} &\delta'(\langle y'_{-1}, y'_{-2}, x_{-1}, \dots, x_{-c}, s_a, b \rangle, y'_0) \\ &= \begin{cases} \langle y'_0, y'_{-1}, x_{-1}, \dots, x_{-c}, s_a, b+1 \rangle, & \text{if } b < 2, \\ \langle y'_0, y'_{-1}, x_0, x_{-1}, \dots, x_{-c+1}, \delta_a(s_a), 2 \rangle, & \text{if } b = 2, \end{cases} \\ &\lambda'(\langle y'_{-1}, y'_{-2}, x_{-1}, \dots, x_{-c}, s_a, b \rangle, y'_0) = x_0, \\ &y'_0, y'_{-1}, y'_{-2} \in Y, \ x_{-1}, \dots, x_{-c} \in X, \ s_a \in S_a, \ b = 0, 1, 2, \\ &x_0 \text{ being } f'(y'_0, y'_{-1}, y'_{-2}, x_{-1}, \dots, x_{-c}, s_a). \end{aligned}$$

As shown above,  $y_i y_{i+1} y_{i+2} = \lambda(\langle x_{i-1}, \dots, x_{i-c}, \delta_a^i(s_a) \rangle, x_i x_{i+1} x_{i+2})$  derives  $x_i = f'(y_{i+2}, y_{i+1}, y_i, x_{i-1}, \dots, x_{i-c}, \delta_a^i(s_a))$ . Then for any state  $s_0 = \langle x_{-1}, \dots, x_{-c}, s_a \rangle$  of  $M$ ,  $\lambda(s_0, x_0 x_1 \dots)$  derives  $\delta(s_0, x_0 \dots x_{i-1}) = s_i$  and  $x_i = f'(y_{i+2}, y_{i+1}, y_i, x_{i-1}, \dots, x_{i-c}, \delta_a^i(s_a))$  for any  $i \geq 0$ , where  $s_i = \langle x_{i-1}, \dots, x_{i-c}, \delta_a^i(s_a) \rangle$ . Thus for a state  $s'_0 = \langle y_{-1}, y_{-2}, x_{-1}, \dots, x_{-c}, s_a, 0 \rangle$  of  $M'$ , it is easy to prove by induction on  $i$  that  $\lambda(s_0, x_0 x_1 \dots) = y_0 y_1 \dots$  derives  $\delta'(s'_0, y_0 \dots y_{i+1}) = s'_{i+2}$  and  $\lambda'(s'_{i+2}, y_{i+2}) = x_i$  for any  $i \geq 0$ , where

$s'_{i+2} = \langle y_{i+1}, y_i, x_{i-1}, \dots, x_{i-c}, \delta_a^i(s_a), 2 \rangle$ . Therefore,  $\lambda(s_0, x_0 x_1 \dots) = y_0 y_1 \dots$  derives  $\lambda'(\delta'(s'_0, y_0 y_1), y_2 y_3 \dots) = \lambda'(s'_2, y_2 y_3 \dots) = x_0 x_1 \dots$ . That is,  $s'_0$  2-matches  $s_0$ . Thus  $M'$  is a weak inverse with delay 2 of  $M$ .

On the other hand, from the last paragraph of the proof of Theorem 2.2.1 in [5 page 55], since  $M$  is stronger connected,  $s_0$  2-matches any state  $s''_2 = \langle y'_{-1}, y'_{-2}, x_{-1}, \dots, x_{-c}, s_a, 2 \rangle$  of  $M'$ ,  $y'_{-2} y'_{-1} \in W_{2, s_0}^M$ . Let  $M''$  be the subautomaton with a state alphabet  $S''$ , where

$$S'' = \{ \langle y_1, y_0, x_{-1}, \dots, x_{-c}, s_a, 2 \rangle \mid y_i = f(x_i, \dots, x_{i-c}, \delta_a^i(s_a)), i, x_0, x_1 = 0, 1 \}.$$

Then  $M$  is a weak inverse with delay 2 of  $M''$  and for any state  $s'' = \langle y_1, y_0, x_{-1}, \dots, x_{-c}, s_a, 2 \rangle$  of  $M''$ , the state  $s = \langle x_{-1}, \dots, x_{-c}, s_a \rangle$  of  $M$  2-matches  $s''$ .

The result can be enhanced as that the state  $\langle x_{-1}, \dots, x_{-c}, s_a \rangle$  of  $M$  2-matches the state  $s' = \langle y'_{-1}, y'_{-2}, x_{-1}, \dots, x_{-c}, s_a, 2 \rangle$  of  $M'$  for any  $y'_{-1}$  and  $y'_{-2}$ .

Let  $\lambda'(s', y'_0 y'_1 y'_2) = x_0 x_1 x_2$ . We use  $h_0, h'_0, h''_0$  to denote  $h_0(x_{-2}, \dots, x_{-c}, s_a)$ ,  $h_0(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a))$ ,  $h_0(x_0, \dots, x_{-c+2}, \delta_a^2(s_a))$ , respectively, and  $h_1, h'_1, h''_1, h'''_1$  to denote  $h_1(x_{-3}, \dots, x_{-c}, s_a)$ ,  $h_1(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a))$ ,  $h_1(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a))$ ,  $h_1(x_0, \dots, x_{-c+3}, \delta_a^3(s_a))$ , respectively.

There are three cases to consider.

In the case of  $h_0 = 0$  or  $h_0 = 1$  &  $h'_0 = 1$ , from (3), we have  $h'_0 = 1$  and  $h''_1 = 0$ . We further consider  $h''_0$  and  $h'''_1$ .

In the subcase of  $h''_0 = 1$ , since  $h'_0 = 1$  &  $h''_0 = 1$ , from the definition of  $f'$ , we have  $x_1 = f_1(x_0, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus y'_0$ . Since  $h''_0 = 1$  &  $h'_1 = 0$ , from the definition of  $f$ , we have  $y'_0 = f_1(x_0, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus x_1 = f(x_2, \dots, x_{-c+2}, \delta_a^2(s_a))$ .

In the subcase of  $h''_0 = 0$  &  $h'''_1 = 1$ , from the definition of  $f'$ ,  $x_2 = f_0(x_1, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus y'_0$  holds. Since  $h''_0 = 0$  &  $h'''_1 = 1$ , from the definition of  $f$ , we have  $y'_0 = f_0(x_1, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus x_2 = f_0(x_2, \dots, x_{-c+2}, \delta_a^2(s_a))$ .

In the subcase of  $h''_0 = 0$  &  $h'''_1 = 0$ , from the definition of  $f'$ ,  $x_2 = f_1(x_0, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus x_1 \oplus x_1 h_2(x_0, \dots, x_{-c+3}, \delta_a^3(s_a)) \oplus y'_0$  holds. Since  $h''_0 = 0$  &  $h'''_1 = 0$ , from the definition of  $f$ ,  $y'_0 = f_1(x_0, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus x_1 \oplus x_1 h_2(x_0, \dots, x_{-c+3}, \delta_a^3(s_a)) \oplus x_2 = f(x_2, \dots, x_{-c+2}, \delta_a^2(s_a))$  holds.

In the case of  $h_0 = 1$  &  $h'_0 = 0$  &  $h''_1 = 1$ , from the definition of  $f'$ , we have  $x_0 = f_2(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus y'_0$ . From (3),  $h''_0 = 1$  holds. Since  $h''_0 = 1$  &  $h''_1 = 1$ , from the definition of  $f$ , we have  $y'_0 = f_2(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus x_0 = f(x_2, \dots, x_{-c+2}, \delta_a^2(s_a))$ .

In the case of  $h_0 = 1$  &  $h'_0 = 0$  &  $h''_1 = 0$ , from the definition of  $f'$ , we have

$$\begin{aligned} x_0 &= f_1(0, x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus f_1(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) \oplus y'_0 \oplus y'_{-1}, \\ x_1 &= f_1(x_{-1}, \dots, x_{-c+1}, \delta_a(s_a)) \oplus x_0 \oplus x_0 h_2(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus y'_{-1}. \end{aligned}$$

It follows that

$$\begin{aligned} x_1 &= f_1(0, x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus x_0 h_2(x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus y'_0 \\ &= f_1(x_0, x_{-1}, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus y'_0. \end{aligned}$$

From (3),  $h_0'' = 1$  holds. Since  $h_0'' = 1$  &  $h_1'' = 0$ , from the definition of  $f$ , we have  $y_0' = f_1(x_0, \dots, x_{-c+2}, \delta_a^2(s_a)) \oplus x_1 = f(x_2, \dots, x_{-c+2}, \delta_a^2(s_a))$ .

To sum up, if  $\lambda'(\langle y_{-1}', y_{-2}', x_{-1}, \dots, x_{-c}, s_a, 2 \rangle, y_0' y_1' y_2') = x_0 x_1 x_2$ , then  $f(x_2, \dots, x_{-c+2}, \delta_a^2(s_a)) = y_0'$ . Using this observation, it is easy to see that if  $\lambda'(\langle y_{-1}', y_{-2}', x_{-1}, \dots, x_{-c}, s_a, 2 \rangle, y_0' y_1' \dots) = x_0 x_1 \dots$ , then  $f(x_{i+2}, \dots, x_{i-c+2}, \delta_a^{i+2}(s_a)) = y_i'$ ,  $i = 0, 1, \dots$ , i.e.  $\lambda(\langle x_{-1}, \dots, x_{-c}, s_a \rangle, x_0 x_1 \dots) = y_0 y_1 y_0' y_1' \dots$  for some  $y_0, y_1 \in Y$ . Thus the state  $\langle x_{-1}, \dots, x_{-c}, s_a \rangle$  of  $M$  2-matches the state  $\langle y_{-1}', y_{-2}', x_{-1}, \dots, x_{-c}, s_a, 2 \rangle$  of  $M'$  for any  $y_{-1}', y_{-2}'$  in  $Y$ .

Let  $\bar{M}' = \langle Y, X, Y^2 \times X^c \times S_a, \bar{\delta}', \bar{\lambda}' \rangle$  be a finite automaton with  $X = Y = \{0, 1\}$  defined by

$$\begin{aligned} \bar{\delta}'(\langle y_{-1}', y_{-2}', x_{-1}, \dots, x_{-c}, s_a \rangle, y_0') &= \langle y_0', y_{-1}', x_0, x_{-1}, \dots, x_{-c+1}, \delta_a(s_a) \rangle, \\ \bar{\lambda}'(\langle y_{-1}', y_{-2}', x_{-1}, \dots, x_{-c}, s_a \rangle, y_0') &= x_0, \\ y_0', y_{-1}', y_{-2}' &\in Y, \quad x_{-1}, \dots, x_{-c} \in X, \quad s_a \in S_a, \\ x_0 &\text{ being } f'(y_0', y_{-1}', y_{-2}', x_{-1}, \dots, x_{-c}, s_a). \end{aligned} \quad (4)$$

Evidently, the state  $\langle y_{-1}', y_{-2}', x_{-1}, \dots, x_{-c}, s_a, 2 \rangle$  of  $M'$  and the state  $\langle y_{-1}', y_{-2}', x_{-1}, \dots, x_{-c}, s_a \rangle$  of  $\bar{M}'$  are equivalent. Thus the state  $\langle x_{-1}, \dots, x_{-c}, s_a \rangle$  of  $M$  2-matches the state  $\langle y_{-1}', y_{-2}', x_{-1}, \dots, x_{-c}, s_a \rangle$  of  $\bar{M}'$  for any  $y_{-1}', y_{-2}'$  in  $Y$ .

For hardware implementation, it is preferable to merged the encoder  $\bar{M}'$  and the decoder  $M$  into one finite automaton with two work modes.

Let

$$\begin{aligned} &\tilde{f}(x_0, \dots, x_{-c}, h_0, h_1', h_1, f_{10}', f_{10}, f_{11}', f_{11}, s_a) \\ &= \begin{cases} f_2(x_{-3}, \dots, x_{-c}, s_a) \oplus x_{-2}, & \text{if } h_0 = 1 \text{ \& } h_1 = 1, \\ x_{-2} f_{11} \oplus \bar{x}_{-2} f_{10} \oplus x_{-1}, & \text{if } h_0 = 1 \text{ \& } h_1 = 0, \\ f_0(x_{-1}, \dots, x_{-c}, s_a) \oplus x_0, & \text{if } h_0 = 0 \text{ \& } h_1' = 1, \\ x_{-2} f_{11} \oplus \bar{x}_{-2} f_{10} \oplus x_0 \oplus x_{-1} \oplus x_{-1}(f_{10}' \oplus f_{11}'), & \text{if } h_0 = 0 \text{ \& } h_1' = 0, \end{cases} \end{aligned}$$

and

$$\begin{aligned} &\tilde{f}'(y_0', y_{-1}', y_{-2}', x_{-1}, \dots, x_{-c}, h_0, h_1', h_1, f_{10}', f_{10}, f_{11}', f_{11}, s_a'', s_a', s_a) \\ &= \begin{cases} f_0(x_{-1}, \dots, x_{-c}, s_a) \oplus y_{-2}', & \text{if } h_0 = 0 \text{ \& } h_1' = 1, \\ (f_{11} x_{-2} \oplus f_{10} \bar{x}_{-2}) \oplus x_{-1} \oplus x_{-1}(f_{11}' \oplus f_{10}') \oplus y_{-2}', & \text{if } h_0 = 0 \text{ \& } h_1' = 0, \\ (f_{11}' x_{-1} \oplus f_{10}' \bar{x}_{-1}) \oplus y_{-1}', & \text{if } h_0 = 1 \text{ \& } h_0(x_{-1}, \dots, x_{-c+1}, s_a') = 1, \\ f_2(x_{-1}, \dots, x_{-c+2}, s_a'') \oplus y_0', & \text{if } h_0 = 1 \text{ \& } h_0(x_{-1}, \dots, x_{-c+1}, s_a') = 0 \text{ \& } h_1(x_{-1}, \dots, x_{-c+2}, s_a'') = 1, \\ f_1(0, x_{-1}, \dots, x_{-c+2}, s_a'') \oplus (f_{11}' x_{-1} \oplus f_{10}' \bar{x}_{-1}) \oplus y_0' \oplus y_{-1}', & \text{if } h_0 = 1 \text{ \& } h_0(x_{-1}, \dots, x_{-c+1}, s_a') = 0 \text{ \& } h_1(x_{-1}, \dots, x_{-c+2}, s_a'') = 0, \end{cases} \end{aligned}$$

where  $\bar{x}_i$  stands for  $x_i \oplus 1$ . Let  $\tilde{M} = \langle U, U, Y^2 \times X^c \times U^7 \times S_a^3 \times \{0, 1, 2\} \times \{E, D\}, \tilde{\delta}, \tilde{\lambda} \rangle$  with  $U = X = Y = \{0, 1\}$  be a finite automaton defined by

$$\begin{aligned} & \tilde{\delta}(\langle y_{-1}, y_{-2}, x_{-1}, \dots, x_{-c}, f'_{10}, f'_{11}, f_{10}, f_{11}, h_0, h'_1, h_1, s''_a, s'_a, s_a, a, b \rangle, u_0) \\ &= \begin{cases} \langle y_{-1}, y_{-2}, x_{-c}, x_{-1}, \dots, x_{-c+1}, f''_{10}, f''_{11}, f'_{10}, f'_{11}, h'_0, h'_1, h'_1, \delta_a(s''_a), s''_a, s'_a, a+1, b \rangle, \\ \quad \text{if } a < 2, \\ \quad \text{where } f''_{1i} = f_1(i, x_{-1}, \dots, x_{-c+2}, s''_a), i = 0, 1, \\ \quad \quad h'_0 = h_0(x_{-1}, \dots, x_{-c+1}, s'_a), h'_1 = h_1(x_{-1}, \dots, x_{-c+2}, s''_a), \\ \langle u_0, y_{-1}, x_0, x_{-1}, \dots, x_{-c+1}, f''_{10}, f''_{11}, f'_{10}, f'_{11}, h'_0, h'_1, h'_1, \delta_a(s''_a), s''_a, s'_a, a, b \rangle, \\ \quad \text{if } b = E \text{ \& } a = 2, \\ \quad \text{where } x_0 = \tilde{f}'(u_0, y_{-1}, y_{-2}, x_{-1}, \dots, x_{-c}, h_0, h'_1, h_1, f'_{10}, f_{10}, f'_{11}, f_{11}, s''_a, s'_a, s_a), \\ \langle y_0, y_{-1}, u_0, x_{-1}, \dots, x_{-c+1}, f''_{10}, f''_{11}, f'_{10}, f'_{11}, h'_0, h'_1, h'_1, \delta_a(s''_a), s''_a, s'_a, a, b \rangle, \\ \quad \text{if } b = D \text{ \& } a = 2, \\ \quad \text{where } y_0 = \tilde{f}(u_0, x_{-1}, \dots, x_{-c}, h_0, h'_1, h_1, f'_{10}, f_{10}, f'_{11}, f_{11}, s_a), \end{cases} \\ & \tilde{\lambda}(\langle y_{-1}, y_{-2}, x_{-1}, \dots, x_{-c}, f'_{10}, f'_{11}, f_{10}, f_{11}, h_0, h'_1, h_1, s''_a, s'_a, s_a, a, b \rangle, u_0) \tag{5} \\ &= \begin{cases} \tilde{f}'(u_0, y_{-1}, y_{-2}, x_{-1}, \dots, x_{-c}, h_0, h'_1, h_1, f'_{10}, f_{10}, f'_{11}, f_{11}, s''_a, s'_a, s_a), & \text{if } b = E, \\ \tilde{f}(u_0, x_{-1}, \dots, x_{-c}, h_0, h'_1, h_1, f'_{10}, f_{10}, f'_{11}, f_{11}, s_a), & \text{if } b = D, \end{cases} \\ & y_{-1}, y_{-2} \in Y, x_{-1}, \dots, x_{-c} \in X, s''_a, s'_a, s_a \in S_a, b = E, D, \\ & h_0, h'_1, h_1, f'_{10}, f_{10}, f'_{11}, f_{11}, u_0 = 0, 1. \end{aligned}$$

Point out that if  $M_a$  is an  $n$ -order binary shift register, then  $s''_a, s'_a, s_a$  occupy  $n+2$  bits memory, not  $3n$  bits.

The key of the cipher  $\tilde{M}$  is  $s = \langle x_{-1}, \dots, x_{-c}, s_a \rangle$  (and the construction). From the definition of  $\tilde{M}$ , for any key  $s = \langle x_{-1}, \dots, x_{-c}, s_a \rangle$  of the cipher  $\tilde{M}$ , it is easy to verify that

$$\begin{aligned} & \tilde{\delta}(\langle \bar{y}_{-1}, \bar{y}_{-2}, x_{-3}, x_{-4}, \dots, x_{-c}, x_{-1}, x_{-2}, \bar{f}'_{10}, \bar{f}'_{11}, \bar{f}_{10}, \bar{f}_{11}, \bar{h}_0, \bar{h}'_1, \bar{h}_1, s_a, \bar{s}'_a, \bar{s}_a, 0, b \rangle, u_0 u_1) \\ &= \langle \bar{y}_{-1}, \bar{y}_{-2}, x_{-1}, \dots, x_{-c}, f'_{10}, f'_{11}, f_{10}, f_{11}, h_0, h'_1, h_1, \delta_a^2(s_a), \delta_a(s_a), s_a, 2, b \rangle, \end{aligned}$$

where

$$\begin{aligned} & f'_{1i} = f_1(i, x_{-2}, \dots, x_{-c+1}, \delta_a(s_a)), f_{1i} = f_1(i, x_{-3}, \dots, x_{-c}, s_a), i = 0, 1, \\ & h_0 = h_0(x_{-2}, \dots, x_{-c}, s_a), h'_1 = h_1(x_{-2}, \dots, x_{-c+1}, \delta_a(s_a)), h_1 = h_1(x_{-3}, \dots, x_{-c}, s_a). \end{aligned}$$

From the definitions of  $\tilde{M}$ ,  $\bar{M}'$  and  $M$ ,

$$\tilde{\delta}(\langle \bar{y}_{-1}, \bar{y}_{-2}, x_{-3}, x_{-4}, \dots, x_{-c}, x_{-1}, x_{-2}, \bar{f}'_{10}, \bar{f}'_{11}, \bar{f}_{10}, \bar{f}_{11}, \bar{h}_0, \bar{h}'_1, \bar{h}_1, s_a, \bar{s}'_a, \bar{s}_a, 0, D \rangle, u_0 u_1)$$

is equivalent to the state  $s = \langle x_{-1}, x_{-2}, \dots, x_{-c}, s_a \rangle$  of  $M$ , and

$$\tilde{\delta}(\langle \check{y}_{-1}, \check{y}_{-2}, x_{-3}, x_{-4}, \dots, x_{-c}, x_{-1}, x_{-2}, \check{f}'_{10}, \check{f}'_{11}, \check{f}_{10}, \check{f}_{11}, \check{h}_0, \check{h}'_1, \check{h}_1, s_a, \check{s}'_a, \check{s}_a, 0, E \rangle, \check{u}_0 \check{u}_1)$$

is equivalent to the state  $s' = \langle \check{y}_{-1}, \check{y}_{-2}, x_{-1}, x_{-2}, \dots, x_{-c}, s_a \rangle$  of  $\bar{M}'$ . Since  $s$  2-matches  $s'$ ,

$$\tilde{\delta}(\langle \bar{y}_{-1}, \bar{y}_{-2}, x_{-3}, x_{-4}, \dots, x_{-c}, x_{-1}, x_{-2}, \bar{f}'_{10}, \bar{f}'_{11}, \bar{f}_{10}, \bar{f}_{11}, \bar{h}_0, \bar{h}'_1, \bar{h}_1, s_a, \bar{s}'_a, \bar{s}_a, 0, D \rangle, u_0 u_1)$$

2-matches

$$\tilde{\delta}(\langle \check{y}_{-1}, \check{y}_{-2}, x_{-3}, x_{-4}, \dots, x_{-c}, x_{-1}, x_{-2}, \check{f}'_{10}, \check{f}'_{11}, \check{f}_{10}, \check{f}_{11}, \check{h}_0, \check{h}'_1, \check{h}_1, s_a, \check{s}'_a, \check{s}_a, 0, E \rangle, \check{u}_0 \check{u}_1).$$

To encrypt a plaintext  $y_0y_1 \dots y_l$  using a key  $\langle x_{-1}, \dots, x_{-c}, s_a \rangle$ , we first choose arbitrarily an initial state

$$s_e = \langle \check{y}_{-1}, \check{y}_{-2}, x_{-3}, x_{-4}, \dots, x_{-c}, x_{-1}, x_{-2}, \check{f}'_{10}, \check{f}'_{11}, \check{f}_{10}, \check{f}_{11}, \check{h}_0, \check{h}'_1, \check{h}_1, s_a, \check{s}'_a, \check{s}_a, 0, E \rangle$$

of  $\tilde{M}$  and  $y_i$ ,  $i = -1, -2, l+1, l+2$  in  $Y$ , then a ciphertext is

$$x_0x_1 \dots x_{l+2} = \tilde{\lambda}(\tilde{\delta}(s_e, y_{-2}y_{-1}), y_0y_1 \dots y_ly_{l+1}y_{l+2}).$$

(The ciphertext only depends on the key  $s$  and  $\check{y}_{-1}, \check{y}_{-2}, y_{l+1}, y_{l+2}$ .) To decrypt, choose arbitrarily an initial state

$$s_d = \langle \bar{y}_{-1}, \bar{y}_{-2}, x_{-3}, x_{-4}, \dots, x_{-c}, x_{-1}, x_{-2}, \bar{f}'_{10}, \bar{f}'_{11}, \bar{f}_{10}, \bar{f}_{11}, \bar{h}_0, \bar{h}'_1, \bar{h}_1, s_a, \bar{s}'_a, \bar{s}_a, 0, D \rangle,$$

$$y'_0y'_1 \dots y'_{l+2} = \tilde{\lambda}(\tilde{\delta}(s_d, 00), x_0x_1 \dots x_{l+2}), \text{ then we have } y_0y_1 \dots y_l = y'_2y'_3 \dots y'_{l+2}.$$

Although  $\tilde{M}$  is not of semi-input-memory, the error propagation length is  $c$ , that is, one bit cipher error has influence on  $\leq c+1$  bits plaintext.

By the way, the automata  $M$  and  $\bar{M}'$  may be regarded as one of candidates of component automata for the generalized algorithms of the finite automaton public key cryptosystems, as they satisfy the conditions  $PN_1(M, \bar{M}', 2)$  and  $PN_2(\bar{M}', M, 2)$ . (cf. [5 pages 374-375])

In fact, we have proven that  $\lambda(s, x_0x_1 \dots) = y_0y_1 \dots$  derives  $\lambda'(s'_2, y_2y_3 \dots) = x_0x_1 \dots$  and that  $\lambda'(\langle y'_{-1}, y'_{-2}, x_{-1}, \dots, x_{-c}, s_a, 2 \rangle, y'_0y'_1 \dots) = x_0x_1 \dots$  derives  $\lambda(s, x_0x_1 \dots) = y_0y_1y'_0y'_1 \dots$  for some  $y_0, y_1 \in Y$ , where  $s = \langle x_{-1}, \dots, x_{-c}, s_a \rangle$  and  $s'_2 = \langle y_1, y_0, x_{-1}, \dots, x_{-c}, s_a, 2 \rangle$ . Since  $s'_2$  is equivalent to the state  $s''$  of  $\bar{M}'$ ,  $\lambda(s, x_0x_1 \dots) = y_0y_1 \dots$  derives  $\bar{\lambda}'(s'', y_2y_3 \dots) = x_0x_1 \dots$ , where  $s'' = \langle y_1, y_0, x_{-1}, \dots, x_{-c}, s_a \rangle$ . Thus the condition  $PN_1(M, \bar{M}', 2)$  is satisfied. Since  $M$  is of semi-input-memory, we have  $\delta(s, x_0x_1) = s_2$ , where  $s_2 = \langle x_1, x_0, \dots, x_{-c+2}, \delta_a^2(s_a) \rangle$ . Clearly, the state  $\langle y'_{-1}, y'_{-2}, x_{-1}, \dots, x_{-c}, s_a, 2 \rangle$  of  $M'$  and the state  $\langle y'_{-1}, y'_{-2}, x_{-1}, \dots, x_{-c}, s_a \rangle$  of  $\bar{M}'$  are equivalent. Thus  $\bar{\lambda}'(\langle y'_{-1}, y'_{-2}, x_{-1}, \dots, x_{-c}, s_a \rangle, y'_0y'_1 \dots) = x_0x_1 \dots$  derives  $\lambda(s_2, x_2x_3 \dots) = y'_0y'_1 \dots$ . That is, the condition  $PN_2(\bar{M}', M, 2)$  is satisfied.