数理解析研究所講究録
第 1744 巻 2011 年 77-84

Universality of Reversible Logic Elements with 1-Bit Memory (Extended Abstract)

Kenichi Morita†, Tsuyoshi Ogiro†, Artiom Alhazov†‡, and Tsuyoshi Tanizawa†
† Graduate School of Engineering, Hiroshima University
‡ Institute of Mathematics and Computer Science, Academy of Sciences of Moldova

Keywords: reversible logic element, universality, reversible computing, rotary element

1 Introduction

A reversible logic element is a building primitive for reversible computing systems, where its logical function is described by a one-to-one mapping. There are two types of reversible logic elements: one without memory, which is usually called a reversible logic gate, and one with memory. Reversible logic gates were first studied by Petri [10]. Then Toffoli [11, 12], and Fredkin and Toffoli [2] studied them in connection with physical reversibility. They showed a Toffoli gate [11] and a Fredkin gate [2] are both logically universal. Hence, every reversible Turing machine can be built by them. On the other hand, Morita [4] proposed a special type of a reversible logic element with 1-bit memory called a rotary element (RE), and showed that reversible Turing machines can be constructed from it. This construction is much simpler than to use reversible logic gates, since there is no need to synchronize signals as in the case of using gates. Morita [5] also showed that an RE can be easily realized in the Billiard Ball Model (BBM), which is a reversible physical model of computation proposed by Fredkin and Toffoli [2].

An RE is a specific 2-state 4-symbol (i.e., it has 4 input lines and 4 output lines) reversible logic element, and thus there are also many other elements of such a type. All the 2-state k-symbol reversible logic elements were classified for $k = 2, 3, 4$, and it was shown that there exist 4 ($k = 2$), 14 ($k = 3$), and 55 ($k = 4$) essentially different non-degenerate ones [7]. Note that a degenerate 2-state k-symbol reversible logic element is a one equivalent to a collection of simple connecting wires that have no meaningful logical function, or a one equivalent to some 2-state $(k-1)$-symbol reversible logic element. Hence, non-degenerate ones are the proper 2-state k-symbol reversible logic elements.

The problem whether there are universal reversible elements that are simpler than an RE was studied by Ogiro et al. [9], and it was shown that all the 14 kinds of non-degenerate 2-state 3-symbol elements are universal by showing that a Fredkin gate can be simulated by a circuit composed of each of them. Later, Ogiro et al. [8] proved each of the 14 kinds of 2-state 3-symbol elements can directly simulate an RE, hence we can construct any reversible Turing machine from it relatively simply.

In this paper, we generalize the above result by showing that every non-degenerate 2-state k-symbol reversible logic element can simulate a rotary element if $k > 2$, and thus they are all universal. One may think that if a 2-state reversible logic element has more input/output symbols, then it will be more powerful, and hence the statement for $k > 3$ is trivial. But, the result we will show here (Theorem 4) is much stronger than it. It claims that really "all" of them are universal, i.e., there exists no non-universal non-degenerate 2-state k-symbol reversible logic element if $k > 2$. We prove it by showing the following fact: for any non-degenerate 2-state k-symbol reversible logic element ($k = 3, 4, \ldots$), we can find a non-degenerate 2-state $(k-1)$-symbol reversible logic element such that the latter is realized by giving a feedback loop to the former. Since all the 14 2-state 3-symbol reversible logic elements are universal, the result follows.

2 Preliminaries

Definition 1 *A deterministic* sequential machine *(SM) M is defined by $M = (Q, \Sigma, \Gamma, \delta)$, where Q is a finite non-empty set of states, Σ and Γ are finite non-empty sets of input and output symbols, respectively. $\delta : Q \times \Sigma \rightarrow Q \times \Gamma$ is a mapping called a* move function. *M is called a* reversible sequential machine *(RSM) if δ is one-to-one (hence $|\Sigma| \leq |\Gamma|$).*

In an RSM, the previous state and the input are determined uniquely from the present state and the output. A *reversible logic elements with memory* (RLEM) is nothing but an RSM (generally with small numbers of states and symbols). In what follows, we consider only 2-state RLEMs such that $|\Sigma| = |\Gamma| = k$ $(k = 2, 3, \ldots)$. We usually omit "2-state," and call them k-symbol RLEMs, which are denoted by k-RLEMs.

A rotary element (RE) is a specific 4-RLEM defined by $M_{\mathrm{RE}} = (\{\boxminus, \boxed{\vert}\}, \{n, e, s, w\}, \{n', e', s', w'\}, \delta_{\mathrm{RE}})$ where δ_{RE} is given in Table 1. We have the following intuitive interpretation for the RE. Inside the finite-state control there is a "rotatable bar." The two states $\boxminus$ and $\boxed{\vert}$ are called state H and state V, respectively, depending on its direction. For each input/output symbol there corresponds an input/output line on which a particle (or token) is placed. When no particle exists, nothing happens on the RE. If a particle arrives at an input line from a direction parallel to the bar, then it goes out from the output line of the opposite side without affecting the direction of the bar (Fig. 1 (a)). If a particle comes from a direction orthogonal to the bar, then it makes a right turn, and rotates the bar by 90 degrees (Fig. 1 (b)). Since an RE is a 4-symbol RLEM, its operation for the cases where two or more particles arrive is undefined. Of course, it is possible to extend its definition so that it can deals with such cases. But, we do not do so, because considering only one-particle case is sufficient for investigating universality of an RE.

Present state	Input n	Input e	Input s	Input w
State H: $\boxminus$	$\boxed{\vert}\ w'$	$\boxminus\ w'$	$\boxed{\vert}\ e'$	$\boxminus\ e'$
State V: $\boxed{\vert}$	$\boxed{\vert}\ s'$	$\boxminus\ n'$	$\boxed{\vert}\ n'$	$\boxminus\ s'$

Table 1: The move function of a rotary element (RE).

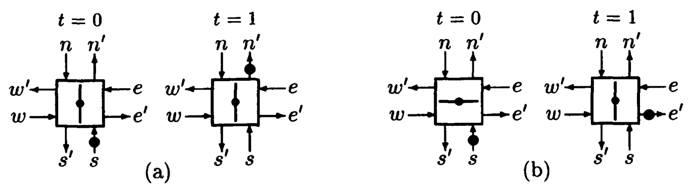


Figure 1: Operations of an RE: (a) the parallel case (state V with input s), and (b) the orthogonal case (state H with input s).

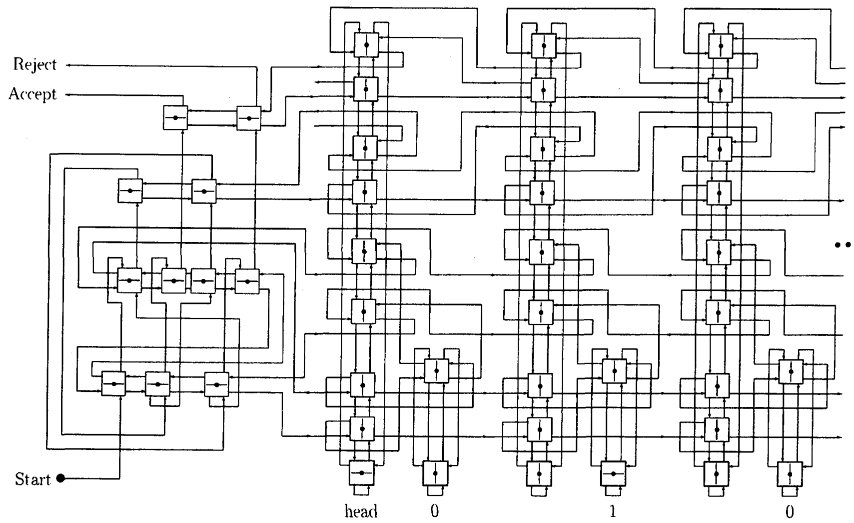


Figure 2: A reversible Turing machine realized by rotary elements [4, 6]. An example of a whole computing process of it is shown in 4,406 figures in [6].

It is known that any reversible Turing machine [1] can be simulated by a reversible logic circuit composed only of REs [4, 6]. Fig. 2 is an example of such a circuit. In this sense, an RE is a universal reversible logic element. On the other hand, it has been shown in [5] that an RE has a simple realization in a billiard ball model, which is an idealized reversible physical model of computing consisting of elastic balls and reflectors [2].

Let $M = (\{0,1\}, \{x_1, x_2, x_3, x_4\}, \{y_1, y_2, y_3, y_4\}, \delta)$ be a 4-RLEM. Since $\delta : \{0,1\} \times \{x_1, x_2, x_3, x_4\} \to \{0,1\} \times \{y_1, y_2, y_3, y_4\}$ is one-to-one, it is specified by a permutation from the set $\{0,1\} \times \{y_1, y_2, y_3, y_4\}$. Hence, there are $8! = 40320$ 4-RLEMs. They are numbered by $0, \ldots, 40319$ in the lexicographic order of permutations. Similarly, there are $6! = 720$ 3-RLEMS and $4! = 24$ 2-RLEMs, which are also numbered in this way [7]. To each number, the prefix "k-" is attached to indicate it is a k-RLEM.

Consider the move function of a 4-RLEM given by Table 2. It defines the 4-RLEM No. 4-289. We use a graphical representation for a 2-state RLEM as shown in Fig. 3. Note that again in Fig. 3, an input signal (or a particle-like object) should be given at most one input line, because each input/output line corresponds to an input/output symbol of an RSM. Therefore, we should not confuse RLEMs with conventional logic gates.

Present state	Input			
	x_1	x_2	x_3	x_4
0	0 y_1	0 y_2	1 y_1	1 y_2
1	0 y_3	0 y_4	1 y_4	1 y_3

Table 2: The move function of the 2-state 4-RLEM 4-289.

In what follows, we use graphical representations for describing RLEMs. We can now construct a *circuit* using RLEMs. Here, we pose the following constraint when composing a circuit: each output line of an RLEM can be connected to at most one input line of an RLEM in the circuit, i.e., fan-out of an output is inhibited. Otherwise, the number of particles increases at each fan-out point. This means that we are assuming a kind of conservation law besides reversibility.

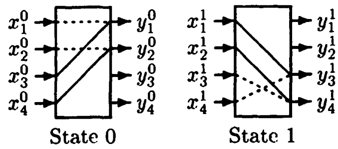


Figure 3: A graphical representation of the 4-RLEM 4-289. Solid and dotted lines describe the input-output relation for each state. A solid line shows the state changes to another, and a dotted line shows the state remains unchanged.

There are many 2-state k-RLEMs even if we limit $k = 2, 3, 4$, but we can regard two RLEMs are equivalent if one can be obtained by "renaming" the states and the input/output symbols of the other. We can see the RE is equivalent to the RLEM 4-289. Because they become exactly the same, if we rename the states, input/output symbols of the RE as follows: $\boxed{\leftarrow} \mapsto 0$, $\boxed{\uparrow} \mapsto 1$, $n \mapsto x_3$, $e \mapsto x_1$, $s \mapsto x_4$, $w \mapsto x_2$, $n' \mapsto y_3$, $e' \mapsto y_2$, $s' \mapsto y_4$, $w' \mapsto y_1$. Here we omit the precise definition of the above notion of equivalence (see e.g., [5]). The numbers of equivalence classes of 2-, 3- and 4-RLEMs are 8, 24 and 82, respectively [7]. Fig. 4 shows representatives of the equivalence classes of 2- and 3-RLEMs, where they are chosen as the ones with the least index.

Among them there are some "degenerate" RLEMs that are further equivalent to connecting wires, or equivalent to a simpler element with fewer symbols. Actually, there are three kinds of degenerate ones:

(i) An RLEM such that there is no input that makes a state change, i.e., two states are disconnected (e.g., RLEM 3-3). Thus, it is nothing but a collection of connecting wires.

(ii) An RLEM such that its relation between inputs and outputs, and the state change are exactly the same in the states 0 and 1 (e.g., RLEM 3-450). Thus, it is equivalent to a 1-state RLEM. Again, it can be regarded as a collection of simple connecting wires.

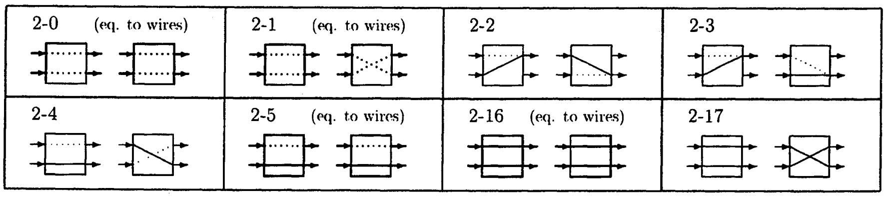
(iii) An RLEM such that there are some input x_i and some output y_j, and the input x_i gives the output y_j both in states 0 and 1 without changing the state (e.g., x_2 and y_2 in RLEM 3-6). Thus, we can see x_i and y_j play only a role of a simple wire. Hence, by removing x_i and y_j from M, it becomes equivalent to some $(k-1)$-RLEM.

An RLEM is called *non-degenerate*, if it is not degenerate. In Fig. 4, degenerate ones are indicated at the upper-right corner of each box.

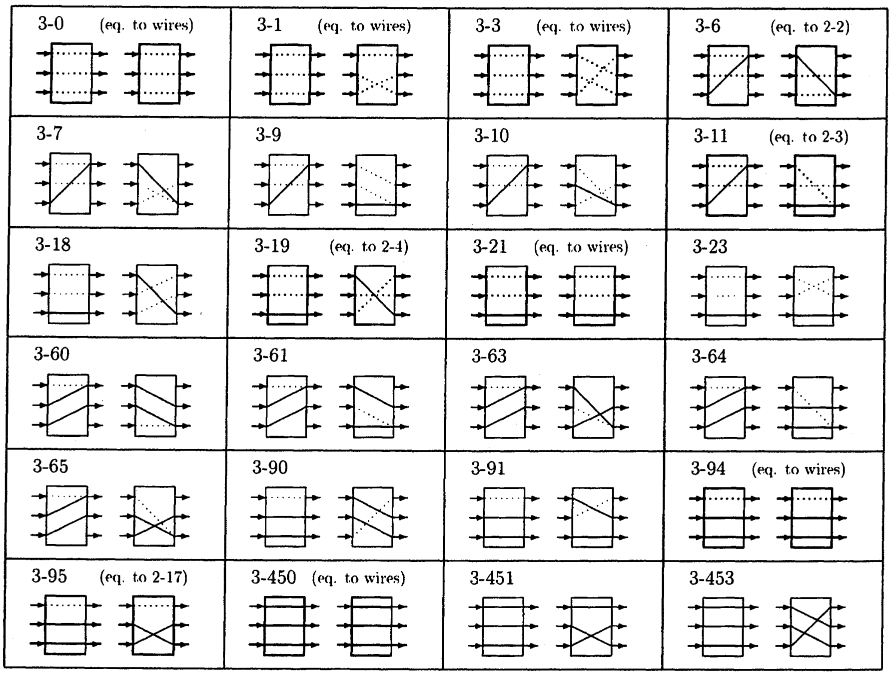
Table 3 shows a further classification result based on the definition of degeneracy. It should be noted that the conditions (i)–(iii) above are not disjoint. Therefore, when counting the number of degenerate ones of type (ii), those of type (i) are excluded. Likewise, when counting ones of type (iii), those of type (i) or (ii) are excluded. The total numbers of equivalence classes of non-degenerate 2-, 3-, and 4-RLEMs are 4, 14, and 55, respectively, and they are the important ones.

k	Total	Degenerate RLEMs			Non-degenerate RLEMs
		Type (i)	Type (ii)	Type (iii)	
2	8	2	2	0	4
3	24	3	3	4	14
4	82	5	4	18	55

Table 3: Numbers of representatives of degenerate and non-degenerate 2-, 3-, and 4-RLEMs [7].



(a)



(b)

Figure 4: Representatives of equivalence classes of (a) 2-RLEMs, and (b) 3-RLEMs [7]. Two states 0 and 1 of an RLEM are given in each box.

3 Simulating an RE by non-degenerate 3-RLEMs

Lemma 2 *[8] An RE can be constructed by any one of non-degenerate 3-RLEMs.*

This lemma has been shown in [8], but we give another method of showing it. In [8] circuits that simulate an RE are so constructed that the delay between inputs and outputs is kept constant. However, if we employ the method of constructing reversible Turing machines by REs as in Fig. 2, there is no need to adjust input/output delays because only one particle exists in the circuit. Thus, here, we construct an RE whose delay is not constant. This method simplifies the proof, and also gives reducibility among 2-, 3-RLEMs and an RE, while in [8] only partial reducibility among them was given.

We first note that Lee et al. [3] showed an RE can be made of 3-RLEM 3-10, and that 3-10 is composed of 2 kinds of 2-RLEMs 2-3 and 2-4. By this, universality of the set {2-3, 2-4} as well as universality of 3-10 are concluded (in [3], RLEMs 3-10, 2-3, and 2-4 are called "coding-decoding module", "reading toggle", and "inverse reading toggle", respectively). Fig. 5 (a) shows a method of realizing 3-10 by 2-3 and 2-4 [3]. Fig. 5 (b) gives a new method of realizing an RE by 3-10. The circuit shown in Fig. 5 (b) reduces

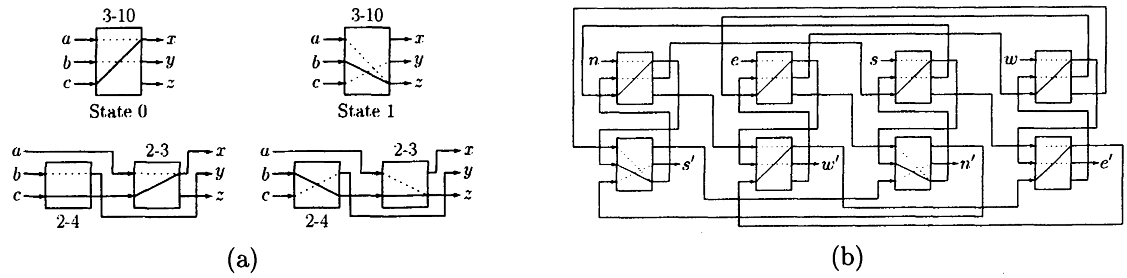


Figure 5: (a) Realizing a 3-RLEM 3-10 by 2-RLEMs 2-4 and 2-3 [3]. (b) Realizing an RE by a circuit made of RLEMs 3-10. This figure corresponds to the state H of an RE.

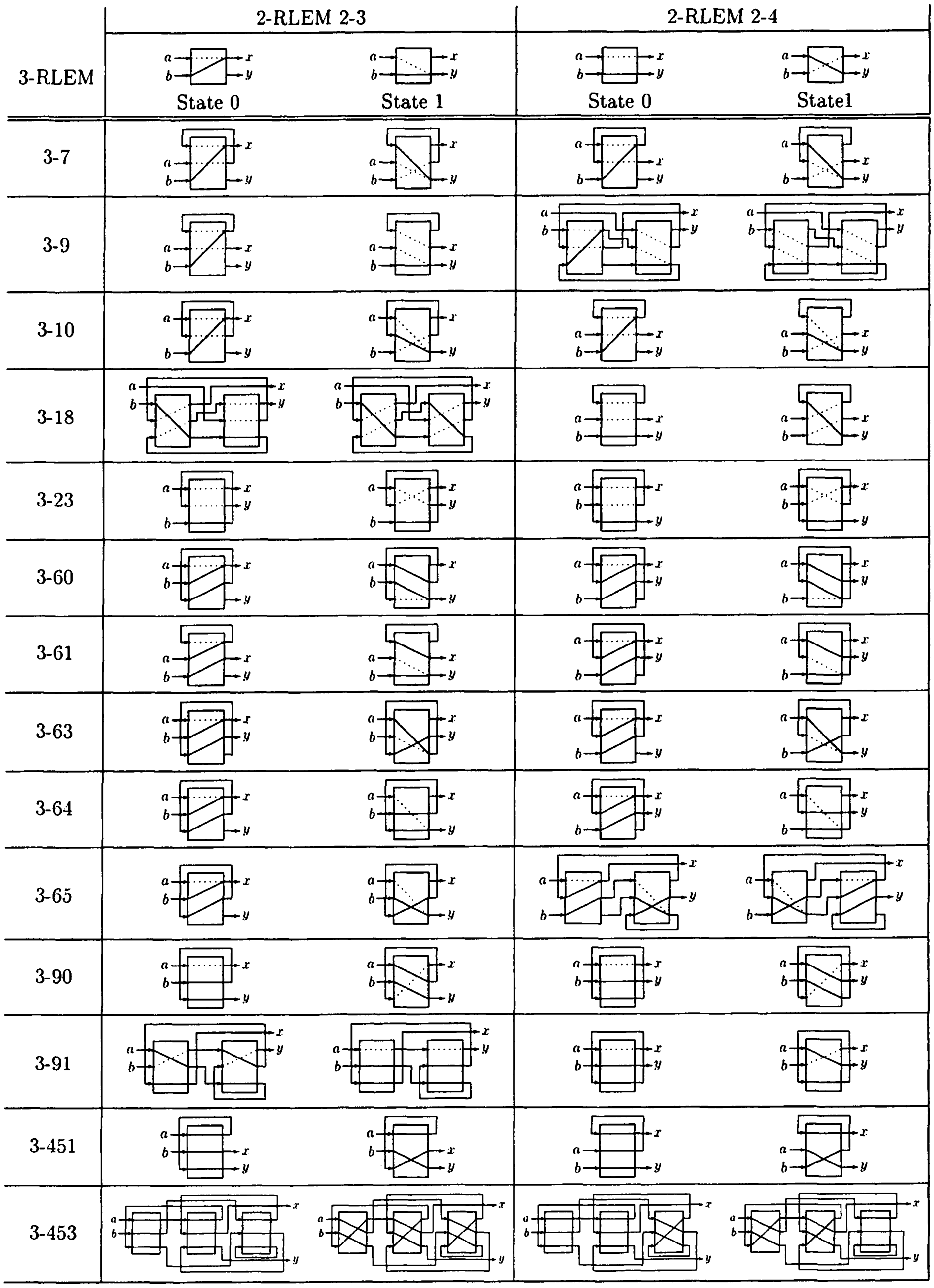


Figure 6: Realizing 2-RLEMs 2-3 and 2-4 by each of 14 kinds of non-degenerate 3-RLEMs.

the total number of needed 3RLEMs 3-10 from 12 (the method in [3]) to 8. Finally, we show that the 2-RLEMs 2-3 and 2-4 can be simulated by each of the 14 non-degenerate 3-RLEMs. It is given in Fig. 6. In most cases, 2-3 and 2-4 are obtained by adding a feed-back loop (i.e., connecting some output to some input) to a 3-RLEM. But, for 6 cases, two or three 3-RLEMs are needed to realize 2-3 or 2-4. Note that, though there is no need to simulate 2-3 and 2-4 by 3-10 (since 3-10 can directly simulate an RE), it is also included in Fig. 6 for completeness. By combining the above three steps, we can obtain a circuit composed only of one kind of non-degenerate 3-RLEMs that simulates an RE.

4 Making a non-degenerate $(k-1)$-RLEM from each of non-degenerate k-RLEMs

Lemma 3 *Let M_k be an arbitrary non-degenerate k-RLEM such that $k > 2$. Then, there exists a non-degenerate $(k-1)$-RLEM M_{k-1} that can be simulated by M_k.*

Since the precise proof of this lemma is complex, only an outline is explained. Choose one output line and one input line of M_k, and connect them to make a feedback. Apparently, a $(k-1)$-RLEM M_{k-1} is obtained. However, if the feedback loop is inappropriate, M_{k-1} will be a degenerate one. Fig. 7 shows examples of giving feedbacks to 4-RLEMs 4-26 and 4-23617. The first two cases are appropriate ones, which produce nondegenerate 3-RLEMs 3-23 and 3-451. But, if an inappropriate feedback loop is given as in the last case, the resulting 3-RLEM becomes degenerate. However, we can prove that it is always possible to find an appropriate feedback loop for any given nondegenerate k-RLEM.

From Lemmas 2 and 3, the following theorem is derived.

Theorem 4 *Every non-degenerate 2-state k-symbol RLEM can realize an RE, and thus it is universal, if $k > 2$.*

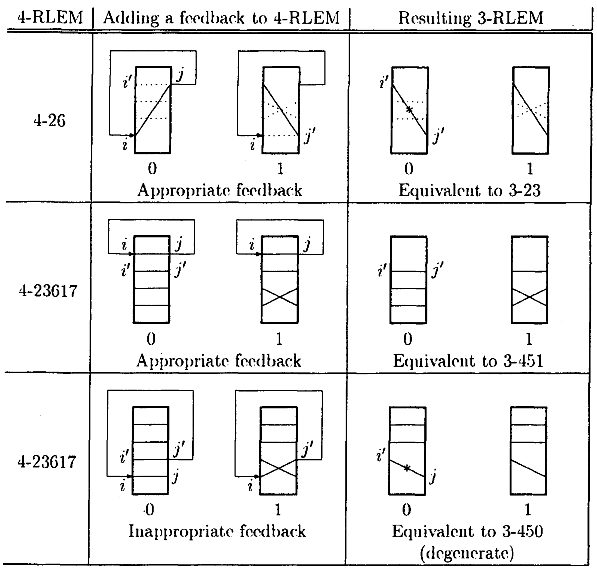


Figure 7: Giving feedback loops to some 4-RLEMs. Edges with * are newly created.

5 Concluding remarks

We proved all non-degenerate k-RLEMs can simulate a rotary element, a universal RLEM, if $k > 2$. Since any RSM can be constructed by rotary elements [4], we can see all these RLEMs are mutually reducible. However, it is an open problem whether each of 4 kinds of non-degenerate 2-RLEMs is universal or not, though it is known that an RE is realized by using both RLEMs 2-3 and 2-4 [3].

Though all non-degenerate 2-state k-RLEMs ($k > 2$) have been proved to be universal, the situation is different for the case of 3 or more states. (Non-degeneracy for many-state RLEMs should be defined appropriately.) Consider a 2-state 2-symbol RLEM e.g., 2-2. It is easy to construct a many-state many-symbol reversible sequential machine M by using only RLEM 2-2, and we can regard M as an RLEM. Therefore, if the RLEM 2-2 is proved to be non-universal, then there exist non-universal non-degenerate n-state k-symbol RLEMs for infinitely many (n, k) such that $n > 2$ and $k > 1$.

Acknowledgment: This work was supported by JSPS Grant-in-Aid for Scientific Research (C) No. 21500015, and for JSPS Fellows No. 20·08364.

References

[1] C. H. Bennett. (1973). Logical reversibility of computation. *IBM J. Res. Dev.*, 17:525–532.

[2] E. Fredkin and T. Toffoli. (1982). Conservative logic. *Int. J. Theoret. Phys.*, 21:219–253.

[3] J. Lee, F. Peper, S. Adachi, and K. Morita. (2008). An asynchronous cellular automaton implementing 2-state 2-input 2-output reversed-twin reversible elements. In *Proc. ACRI 2008,* Yokohama, LNCS-5191, pages 67–76.

[4] K. Morita. (2001). A simple universal logic element and cellular automata for reversible computing. In *Proc. 3rd Int. Conf. on Machines, Computations, and Universality,* Chisinau, LNCS-2055, pages 102–113.

[5] K. Morita. (2008). Reversible computing and cellular automata — a survey. *Theoret. Comput. Sci.*, 395(1):101–131.

[6] K. Morita. (2010). Constructing a reversible turing machine by a rotary element, a reversible logic element with memory. In *Hiroshima University Institutional Repository,* http://ir.lib.hiroshima-u.ac.jp/00029224.

[7] K. Morita, T. Ogiro, A. Kanno, K. Tanaka, and H. Kato. (2005). Classification and universality of reversible logic elements with one-bit memory. In *Proc. 4th Int. Conf. on Machines, Computations, and Universality,* Saint Petersburg, LNCS-3354, pages 245-256.

[8] T. Ogiro, A. Alhazov, T. Tanizawa, and K. Morita. (2010). Universality of 2-state 3-symbol reversible logic elements — a direct simulation method of a rotary element. In *Proc. IWNC 2009,* Himeji, PICT-2, Springer-Verlag, pages 252–259.

[9] T. Ogiro, A. Kanno, K. Tanaka, H. Kato, and K. Morita. (2005). Nondegenerate 2-state 3-symbol reversible logic elements are all universal. *Int. Journ. of Unconventional Computing,* 1:47–67.

[10] C.A. Petri. (1967). Grundsätzliches zur beschreibung diskreter prozesse. In *Proc. 3rd Colloquium über Automatentheorie,* Birkhäuser Verlag, pages 121–140.

[11] T. Toffoli. (1980). Reversible computing. In *Automata, Languages and Programming,* LNCS-85, Springer-Verlag, pages 632–644.

[12] T. Toffoli. (1981). Bicontinuous extensions of invertible combinatorial functions. *Mathematical Systems Theory,* 14:12–23.