THE TRANSACTIONS OF THE IEICE, VOL. E **72**, NO. **3** MARCH **1989**

PAPER

A 1-Tape 2-Symbol Reversible Turing Machine

Kenichi MORITA†, *Member,* Akihiko SHIRASAKI††, *Nonmember* *and* Yoshifumi GONO†††, *Member*

SUMMARY Bennett proved that any irreversible Turing machine can be simulated by a reversible one. However, Bennett's reversible machine uses 3 tapes and many tape symbols. Previously, Gono and Morita showed that the number of symbols can be reduced to 2. In this paper, by improving these methods, we give a procedure to convert an irreversible machine into an equivalent 1-tape 2-symbol reversible machine. First, it is shown that the "state-degeneration degree" of any Turing machine can be reduced to 2 or less. Using this result and some other techniques, a given irreversible machine is converted into a 1-tape 32-symbol (i. e., 5-track 2-symbol) reversible machine. Finally the 32-symbol machine is converted into a 1-tape 2-symbol reversible machine. From this result, it is seen that a 1-tape 2-symbol reversible Turing machine is computation universal.

1. Introduction

A reversible Turing machine is a "backward deterministic" Turing machine, and thus every computational configuration of it has at most one predecessor. Usual Turing machines are, in general, irreversible since they can "forget" their previous internal states or can "erase" a symbol on a tape. Reversible computations are very important when considering the minimal power dissipation in a computation theoretically. Many researchers have been studying them from this standpoint[(2)-(4),(6),(7)], and suggested that it is ideally possible to devise a power dissipationless computing mechanism by making use of their reversibility.

Bennett[(1)] proved that every irreversible Turing machine can be simulated by a 3-tape many-symbol reversible Turing machine. Of course, it is easy to simulate an irreversible machine by a reversible one by recording all the movements (history) of the former machine step by step. This method, however, leaves large amount of garbage informations on the tape at the end of the computation. The important point of Bennett's construction is that it is possible to reversibly erase these garbage records without erasing the computed results.

However, the reversible machine constructed by Bennett uses 3 tapes and many tape symbols. Previously, Gono et al.[(5)] showed that a 3-tape 2-symbol machine suffices to reversibly simulate an irreversible one.

In this paper, by improving the methods in Refs. (1) and (5), we prove that any irreversible Turing machine can be converted into an equivalent 1-tape 2-symbol (i. e., of the simplest form) reversible Turing machine. First we define the state-degeneration degree of a Turing machine, and show that the state-degeneration degree of any machine can be reduced to 2 or less (Lemma 1). Using this result and some other techniques, a given irreversible machine is converted into a 1-tape 32-symbol (i. e., 5-track 2-symbol) reversible machine (Lemma 2). Finally the 32-symbol reversible machine is converted into a 1-tape 2-symbol reversible machine (Lemma 3).

2. Definitions

A 1-tape Turing machine T is a system defined by

$$T=(Q, S, q(0), q(f), t_0, F),$$

where

(1) Q is a non-empty finite set of states,
(2) S is a non-empty finite set of tape symbols,
(3) $q(0)$ is an initial state ($q(0)\in Q$),
(4) $q(f)$ is a final state ($q(f)\in Q$),
(5) t_0 is a special blank symbol ($t_0\in S$), and
(6) F is a subset of $Q\times S\times S\times Q\cup Q\times\{/\}\times\{-, 0, +\}\times Q$.

(In what follows, we assume each state of a given Turing machine is denoted in the form $q(X)$.)

Note that F is a move function in quadruple form. Although usually F is written in quintuple form, we adopt quadruple notation according to Bennett[(1)]. Because, it is convenient to use quadruples when reversibility is in issue. The reason is as follows: (1) reversibility of a Turing machine can be easily defined from its quadruple set F, and (2) it is also easy to construct a "reverse quadruple" corresponding to the reverse move of a given quadruple.

Each quadruple is of the form $[q(r), t, t', q(s)]$ or $[q(r), /, d, q(s)]$, where $q(r), q(s)\in Q$, $t, t'\in S$, and $d\in\{-, 0, +\}$. The symbols "$-$", "0", and "$+$" denote "left-shift", "zero-shift", and "right-shift", respectively. $[q(r), t, t', q(s)]$ means that if T reads the symbol t in

Manuscript received September 5, 1988.

† The author is with the Faculty of Engineering, Yamagata University, Yonezawa-shi, 992 Japan.
†† The author is with Dainippon Screen Mfg., Co., Ltd., Kyoto-shi, 602 Japan.
††† The author is with Shinko Electric Industries Co., Ltd., Nagano-shi, 380 Japan.

state $q(r)$, write t' and go to state $q(s)$. $[q(r), /, d, q(s)]$ means that if T is in state $q(r)$, shift the head to the direction d and go to state $q(s)$.

Let α_1 and α_2 be two quadruples in F.

$$\alpha_1=[q(r_1), b_1, c_1, q(s_1)]$$

$$\alpha_2=[q(r_2), b_2, c_2, q(s_2)]$$

We say that α_1 and α_2 overlap in domain iff
(i) $q(r_1)=q(r_2)$ and $b_1=b_2$, or
(ii) $q(r_1)=q(r_2)$ and b_1 or b_2 is "/".
We say that α_1 and α_2 overlap in range iff
(i) $q(s_1)=q(s_2)$ and $c_1=c_2$, or
(ii) $q(s_1)=q(s_2)$ and b_1 or b_2 is "/".
A quadruple α is said to be deterministic (in F) iff there is no other quadruple in F with which α overlaps in domain. On the other hand, α is said to be reversible (in F) iff there is no other quadruple in F with which α overlaps in range. A Turing machine T is called deterministic iff every quadruple in F is deterministic, and is called reversible iff every quadruple in F is reversible. In what follows, we consider only deterministic 1-tape Turing machines, and discuss their reversibility.

Let $q(s)$ be a state of T. $q(s)$ is called state-degenerative if there are at least two distinct quadruples $[q(r_1), b_1, c_1, q(s)]$ and $[q(r_2), b_2, c_2, q(s)]$ in F. If there are exactly k such quadruples in F, we say that state-degeneration degree of $q(s)$ is k, and denote it as $\mathrm{sdeg}(q(s))=k$. That is,

$$\mathrm{sdeg}(q(s))=|\{\alpha|\alpha=[q(r), b, c, q(s)]\in F\}|,$$

where $|A|$ denotes the number of elements of A. Note that if $q(s)$ is not state-degenerative, then $\mathrm{sdeg}(q(s))\leqq 1$. State-degeneration degree of T is defined as

$$\mathrm{sdeg}(T)=\max\{\mathrm{sdeg}(q(s))|q(s)\in Q\}.$$

3. Converting an Irreversible Turing Machine into a 1-Tape 32-Symbol Reversible Turing Machine

In this section, we show a method to convert an arbitrary irreversible 1-tape Turing machine into a 1-tape 32-symbol (i. e., 5-track 2-symbol) reversible Turing machine (Lemma 2).

As a preliminary, we show Lemma 1. The proof of this lemma is essentially the same as in Ref. (5), where somewhat different notion of degeneration (overlapping in range of quadruples) is discussed.

[Lemma 1] For any Turing machine T, we can construct an equivalent Turing machine T' such that $\mathrm{sdeg}(T')\leqq 2$.

(Proof) Let $T=(Q, S, q(0), q(f), t_0, F)$, and let $q(s)\in Q$ be a state such that $\mathrm{sdeg}(q(s))>2$ (if no such $q(s)$ exists, the lemma is proved). If $\mathrm{sdeg}(q(s))=k$, there are k distinct quadruples in F as follows.

$$[q(r_1), b_1, c_1, q(s)]$$
$$[q(r_2), b_2, c_2, q(s)]$$
$$[q(r_3), b_3, c_3, q(s)]$$
$$\vdots$$
$$[q(r_k), b_k, c_k, q(s)]$$

In T', the above k quadruples are replaced by the following $2k-2$ quadruples.

(1)	$[q(r_1)$	$, b_1$	$, c_1$	$, q([s, 1, *\ ])]$
	$[q(r_2)$	$, b_2$	$, c_2$	$, q([s, 1, *\ ])]$
(2)	$[q([s, 1, *\ ])$	$, /$	$, 0$	$, q([s, 2, *\ ])]$
	$[q(r_3)$	$, b_3$	$, c_3$	$, q([s, 2, *\ ])]$
		$\vdots$		
(i)	$[q([s, i-1, *\ ])$	$, /$	$, 0$	$, q([s, i, *\ ])]$
	$[q(r_{i+1})$	$, b_{i+1}$	$, c_{i+1}$	$, q([s, i, *\ ])]$
		$\vdots$		
(k-2)	$[q([s, k-3, *\ ])$	$, /$	$, 0$	$, q([s, k-2, *\ ])]$
	$[q(r_{k-1})$	$, b_{k-1}$	$, c_{k-1}$	$, q([s, k-2, *\ ])]$
(k-1)	$[q([s, k-2, *\ ])$	$, /$	$, 0$	$, q(s)\ ]$
	$[q(r_k)$	$, b_k$	$, c_k$	$, q(s)\ ]$

$q([s, i, *\ ])$ $(1\leqq i\leqq k-2)$ are new states, and added to the state set of T'. Repeating this procedure for all $q(s)$ such that $\mathrm{sdeg}(q(s))>2$, we can obtain T' with $\mathrm{sdeg}(T')\leqq 2$. It is clear that T' is equivalent to T. (Q. E. D.)

[Lemma 2] For any irreversible 1-tape Turing machine T, we can construct a 1-tape 32-symbol (i. e., 5-track 2-symbol) reversible Turing machine R which simulates T.

(Proof) Let

$$T=(Q, S, q(0), q(f), t_0, F)$$

be a given deterministic irreversible Turing machine. We can assume, without loss of generality, T holds the following conditions (it is easy to convert T so that it satisfies these conditions).

(a) The tape is one-way (rightward) infinite.
(b) The set S of tape symbols is $\{0, 1\}$ (0 is the blank symbol).
(c) In order to finitely determine the used portion of the tape, symbol sequences "10" and "11" are used as "codes" to represent two distinct (macro) symbols. And thus "00" never appears between any two 1's.
(d) The leftmost two squares of the tape are always 0's.
(e) When T starts or stops, the head is at the leftmost square.
(f) The initial state $q(0)$ never appears at the fourth position of a quadruple in F (i. e., $\mathrm{sdeg}(q(0))=0$).
(g) $\mathrm{sdeg}(T)=2$ (by Lemma 1).

R is constructed from T in the following way. We now write R as

$$R=(Q', S', q(0), p(0), [0, 0, 0, 0, 0], F'),$$

where $S'=\{[t_1, t_2, t_3, t_4, t_5] | t_i \in \{0,1\}\}$ (t_i represents the contents of the i-th track). Note that the initial state of R coincides with that of T's.

The reversible Turing machine shown Ref. (1) uses three tapes (i.e., the working tape, the history tape, and the output tape) to simulate T. In R, these three tapes are simulated using five tracks.

1st track : working tape (i.e., T's tape)
2nd track : head position of the working tape
3rd track : history tape
4th track : head position of the history tape
5th track : output tape

The tape of R is also one-way (rightward) infinite. Its leftmost square always contains $[0, 1, 0, 1, 0]$. Initially, the contents of each track except the leftmost square is as follows.

1st track : T's initial tape
2nd track : 100…
3rd track : 000…
4th track : 100…
5th track : 000…

Head positions of the working and the history tapes are represented by 1's on the tracks 2 and 4. The initial head position of R is at the second square from the left (Fig. 1).

The entire computation process of R is divided into three stages as in Ref. (1) (provided that T halts). They are the compute stage, the copy stage, and the retrace stage. Let F_1, F_2 and F_3 be quadruple sets of R in these three stages (thus $F'=F_1 \cup F_2 \cup F_3$). These quadruple sets are defined as follows.

[Compute stage] This stage is a forward simulation process of T. R simulates T using the tracks 1 and 2. When T executes a reversible quadruple, R simulates T in a straightforward manner. On the other hand, when T executes an irreversible quadruple, R does an additional movements. That is, R records the information which quadruple is used on the 3rd track in order to keep R reversible.

The quadruple set F_1 of R in the compute stage is as follows.

(1) For each reversible quadruple

$$\alpha=[q(r), t, t', q(s)] \in F \quad (t, t' \in S),$$

include the following quadruples in F_1.

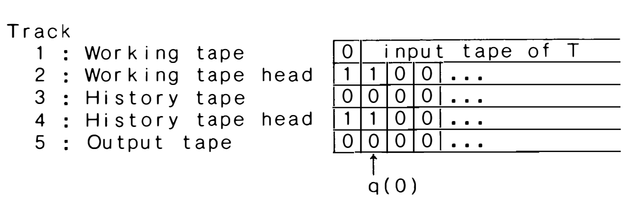


Fig. 1 The initial configuration of R.

$$[q(r), [t, 1, x, y, 0], [t', 1, x, y, 0], q(s)]$$
$$(x, y \in \{0, 1\})$$

(Note that the above is a "quadruple scheme" which represents 4 quadruples, because x, $y \in \{0, 1\}$.)

(2) For each reversible quadruple

$$\alpha=[q(r), /, d, q(s)] \in F \quad (d \in \{-, 0, +\}),$$

include the following quadruples in F_1.

$$[\ q(r),\ [v, 1, x, y, 0], [v, 0, x, y, 0], q([r, ¢])]$$
$$[q([r, ¢]),\ /,\ d,\ q([r, \$])]$$
$$[q([r, \$]), [v, 0, x, y, 0], [v, 1, x, y, 0],\ q(s)\ ]$$
$$(v, x, y \in \{0, 1\})$$

(3) For each pair of irreversible quadruples

$$\alpha_1=[q(r_1), b_1, c_1, q(s)] \in F \text{ and}$$
$$\alpha_2=[q(r_2), b_2, c_2, q(s)] \in F$$

which overlap in range, do (i)-(iii).

(i) For each $h(=1, 2)$, if $b_h=/$ then include the following quadruples in F_1.

$$[\ q(r_h),\ [v, 1, x, y, 0],\ [v, 0, x, y, 0],\ q([s, 0, h])]$$
$$[q([s, 0, h]),\ /,\ c_h,\ q([s, 1, h])]$$
$$[q([s, 1, h]), [v, 0, x, y, 0],\ [v, 1, x, y, 0],\ q([s, 2, h])]$$
$$[q([s, 2, h]),\ /,\ -,\ q([s, 3, h])]$$
$$[q([s, 3, h]), [v, 0, x, y, 0],\ [v, 0, x, y, 0],\ q([s, 2, h])]$$
$$[q([s, 3, h]), [0, 1, 0, 1, 0],\ [0, 1, 0, 1, 0],\ q([s, 4, h])]$$
$$[q([s, 4, h]),\ /,\ +,\ q([s, 5, h])]$$
$$[q([s, 5, h]), [v, w, x, 0, 0],\ [v, w, x, 0, 0],\ q([s, 4, h])]$$
$$[q([s, 5, h]), [v, w, x, 1, 0],\ [v, w, x, 0, 0],\ q([s, 6, h])]$$
$$[q([s, 6, h]),\ /,\ +,\ q([s, 7, h])]$$
$$[q([s, 7, h]), [v, w, 0, 0, 0], [v, w, h-1, 1, 0], q([s, 0])\ ]$$
$$(v, w, x, y \in \{0, 1\})$$

(ii) For each $h(=1, 2)$, if $b_h \in S$ then include the following quadruples in F_1.

$$[\ q(r_h),\ [b_h, 1, x, y, 0], [c_h, 1, x, y, 0], q([s, 2, h])]$$
$$[q([s, 2, h]),\ /,\ -,\ q([s, 3, h])]$$
$$[q([s, 3, h]), [v, 0, x, y, 0], [v, 0, x, y, 0], q([s, 2, h])]$$
$$[q([s, 3, h]), [0, 1, 0, 1, 0], [0, 1, 0, 1, 0], q([s, 4, h])]$$
$$[q([s, 4, h]),\ /,\ +,\ q([s, 5, h])]$$
$$[q([s, 5, h]), [v, w, x, 0, 0], [v, w, x, 0, 0], q([s, 4, h])]$$
$$[q([s, 5, h]), [v, w, x, 1, 0], [v, w, x, 0, 0], q([s, 6, h])]$$
$$[q([s, 6, h]),\ /,\ +,\ q([s, 7, h])]$$
$$[q([s, 7, h]), [v, w, 0, 0, 0], [v, w, h-1, 1, 0], q([s, 0])\ ]$$

$$(v, w, x, y \in \{0, 1\})$$

(iii) Include the following quadruples in F_1.

$$[q([s,0]),\quad / \quad,\quad - \quad, q([s,1])]$$
$$[q([s,1]), [v, w, x, 0, 0], [v, w, x, 0, 0], q([s,0])]$$
$$[q([s,1]), [0,1,0,1,0], [0,1,0,1,0], q([s,2])]$$
$$[q([s,2]),\quad / \quad,\quad + \quad, q([s,3])]$$
$$[q([s,3]), [v,0,x,y,0], [v,0,x,y,0], q([s,2])]$$
$$[q([s,3]), [v,1,x,y,0], [v,1,x,y,0], q(s)]$$
$$(v, w, x, y \in \{0, 1\})$$

It is seen that R can simulate T by the above quadruples. Furthermore, we can see that each quadruple in F_1 is deterministic and reversible. It can be verified by noticing the following facts and by a careful inspection.

In the above procedure (2), the newly added states $q([r, ¢])$ and $q([r, \$])$ uniquely correspond to α in (2) (i. e., they appear only in these three quadruples), since α is deterministic and thus $q(r)$ never appears at the first position of other quadruples in F. In (3), the newly added states $q([s, i, h])$ $(i=0,\cdots,7)$ uniquely correspond to α_h, and $q([s, i])$ $(i=0,\cdots,3)$ uniquely correspond to the pair $\{\alpha_1, \alpha_2\}$. Because $\mathrm{sdeg}(T)=2$ and thus $q(s)$ never appears at the fourth position of other quadruples in F.

[Copy stage] In this stage, R simply copies the contents of the 1st track into the 5th track. In order to determine the portion to be copied, the assumptions (c) and (d) for T is used. The quadruple set F_2 of the copy stage is as follows, where $c([i, j])$ $(i=1,\cdots,4, j=1,2)$ and $p(f)$ are newly added states.

$$[q(f), [0,w,x,y,0], [0,w,x,y,0], c([1,0])]$$
$$[c([1,0]),\quad / \quad,\quad + \quad, c([2,0])]$$
$$[c([2,0]), [v,w,x,y,0], [v,w,x,y,v], c([3,0])]$$
$$[c([3,0]),\quad / \quad,\quad + \quad, c([4,0])]$$
$$[c([4,0]), [1,w,x,y,0], [1,w,x,y,1], c([1,0])]$$
$$[c([4,0]), [0,w,x,y,0], [0,w,x,y,0], c([4,1])]$$
$$[c([4,1]),\quad / \quad,\quad - \quad, c([3,1])]$$
$$[c([3,1]), [v,w,x,y,v], [v,w,x,y,v], c([2,1])]$$
$$[c([2,1]),\quad / \quad,\quad - \quad, c([1,1])]$$
$$[c([1,1]), [1,w,x,y,1], [1,w,x,y,1], c([4,1])]$$
$$[c([1,1]), [0,w,x,y,0], [0,w,x,y,0], p(f)]$$
$$(v, w, x, y \in \{0, 1\})$$

It is easily seen that these quadruples are all deterministic and reversible.

[Retrace stage] This stage is a backward simulation process of T in order to reversibly erase the history track. This process is performed by executing the "reverse quadruples" of F_1. The quadruple set F_3 of the retrace stage is as follows, where $p(X)$ are newly added states.

(1) For each quadruple

$$[q(r), [v, w, x, y, 0], [v', w', x', y', 0], q(s)] \in F_1$$

include the following quadruples in F_3.

$$[p(s), [v', w', x', y', 0], [v, w, x, y, 0], p(r)]$$
$$[p(s), [v', w', x', y', 1], [v, w, x, y, 1], p(r)]$$

Note that, although the quadruples of the form $[q(r), [v, w, x, y, 1], [v', w', x', y', 1], q(s)]$ might be included in F_1, they are useless since the output (5th) track is entirely blank in the compute stage. In the retrace stage, however, the above two kinds of quadruples should be included.

(2) For each quadruple

$$[q(r), /, d, q(s)] \in F_1,$$

include the following quadruple in F_3, where $\mathrm{rev}(-)=+$, $\mathrm{rev}(0)=0$, $\mathrm{rev}(+)=-$.

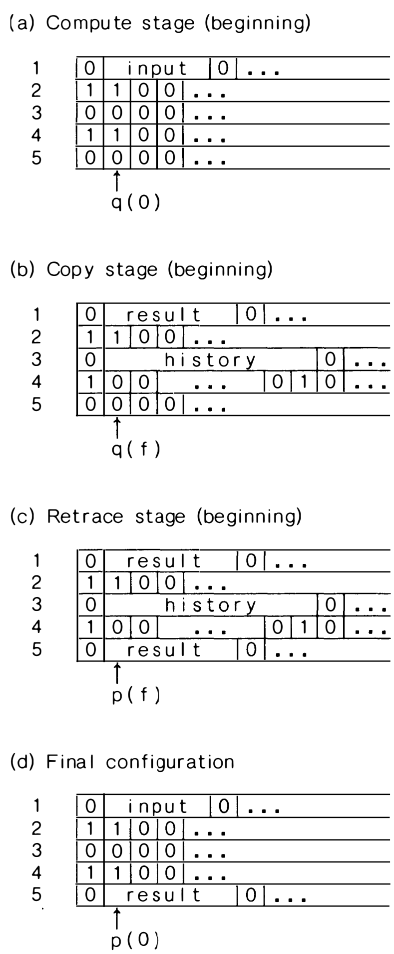


Fig. 2 The computation process of R.

$[p(s), /, \mathrm{rev}(d), p(r)]$

Since quadruples in F_1 are deterministic and reversible, these "reverse quadruples" in F_3 are also deterministic and reversible. These quadruples undo the computation in the compute stage, retaining the output track unchanged. Eventually R reaches to $p(0)$, which is assured to be a halting state by the assumption (f).

The computation process of R is shown in Fig. 2. If T halts, R also halts in the state $p(0)$ leaving the result in the track 5. (Q. E. D.)

4. Converting a 32-Symbol Reversible Turing Machine into a 2-Symbol Reversible Turing Machine

In this section, we show a method to convert an arbitrary 1-tape 32-symbol reversible Turing machine into a 1-tape 2-symbol reversible Turing machine (this method is easily generalized to a Turing machine with any number of tape symbols).

[Lemma 3] For any 1-tape 32-symbol (i. e., 5-track 2-symbol) reversible Turing machine R_1, we can construct a 1-tape 2-symbol reversible Turing machine R_2 which simulates R_1.

(Proof) Let

$$R_1=(Q_1, S, q(0), q(f), [0, 0, 0, 0, 0], F_1)$$

be a given 1-tape 32-symbol reversible Turing machine, where $S=\{[t_1, t_2, t_3, t_4, t_5] | t_i \in \{0, 1\}\}$.

R_2 is constructed from R_1 as follows.

$$R_2=(Q_2, \{0, 1\}, q(0), q(f), 0, F_2)$$

Note that the initial and the final states coincide with those of R_1's. R_2 simulates one square of R_1's tape using consecutive 5 squares of R_2's tape. The quadruple set F_2 is given as follows.

(1) For each quadruple

$$[q(r), /, 0, q(s)] \in F_1$$

include the following quadruple in F_2.

$$[q(r), /, 0, q(s)]$$

(2) For each quadruple

$$[q(r), /, d, q(s)] \in F_1 \quad (d \neq 0)$$

include the following 5 quadruples in F_2.

$$[\ q(r)\ , /, d, q([r, 1])]$$
$$[q([r, 1]), /, d, q([r, 2])]$$
$$[q([r, 2]), /, d, q([r, 3])]$$
$$[q([r, 3]), /, d, q([r, 4])]$$
$$[q([r, 4]), /, d, \ q(s)\ ]$$

(3) For each quadruple

$$\alpha=[q(r), [a, b, c, d, e], [f, g, h, i, j], q(s)] \in F_1$$

add (i. e., take the set union) the following 17 quadruples to F_2. These quadruples simulate α, keeping the symbols that have been read or should be written by the newly added states.

$$[\ q(r)\ , a, 0, q([r, [a, -, -, -], 0])]$$
$$[q([r, [a, -, -, -], 0]), /, +, q([r, [a, -, -, -], 1])]$$
$$[q([r, [a, -, -, -], 1]), b, 0, q([r, [a, b, -, -], 0])]$$
$$[q([r, [a, b, -, -], 0]), /, +, q([r, [a, b, -, -], 1])]$$
$$[q([r, [a, b, -, -], 1]), c, 0, q([r, [a, b, c, -], 0])]$$
$$[q([r, [a, b, c, -], 0]), /, +, q([r, [a, b, c, -], 1])]$$
$$[q([r, [a, b, c, -], 1]), d, 0, q([r, [a, b, c, d], 0])]$$
$$[q([r, [a, b, c, d], 0]), /, +, q([r, [a, b, c, d], 1])]$$
$$[q([r, [a, b, c, d], 1]), e, j, q([s, [f, g, h, i], 2])]$$
$$[q([s, [f, g, h, i], 2]), /, -, q([s, [f, g, h, i], 3])]$$
$$[q([s, [f, g, h, i], 3]), 0, i, q([s, [f, g, h, -], 2])]$$
$$[q([s, [f, g, h, -], 2]), /, -, q([s, [f, g, h, -], 3])]$$
$$[q([s, [f, g, h, -], 3]), 0, h, q([s, [f, g, -, -], 2])]$$
$$[q([s, [f, g, -, -], 2]), /, -, q([s, [f, g, -, -], 3])]$$
$$[q([s, [f, g, -, -], 3]), 0, g, q([s, [f, -, -, -], 2])]$$
$$[q([s, [f, -, -, -], 2]), /, -, q([s, [f, -, -, -], 3])]$$
$$[q([s, [f, -, -, -], 3]), 0, f, q(s)\ ]$$

Note that the above 17 quadruples do not uniquely correspond to α. For example, 17 quadruples converted from

$$[q(r), [0, 1, 0, 0, 0], [1, 1, 0, 0, 0], q(s)]$$

have the first 6 and last 2 quadruples in common with those from

$$[q(r), [0, 1, 0, 1, 0], [1, 0, 0, 1, 0], q(s)].$$

Of course, this does not break the reversibility of R_2.

It is easy to see that R_2 simulates R_1. (Q. E. D.)

From Lemmas 1-3, we obtain the following theorem.

[Theorem 1] For any irreversible Turing machine T, we can construct a 1-tape 2-symbol reversible Turing machine R which simulates T.

5. Conclusion

We have shown that any irreversibie Turing machine can be converted into a 1-tape 2-symbol reversible Turing machine (Theorem 1). At the end of the computation, the reversible machine leaves only the input and the result (answer) on the tape (leaves no garbage informations).

In order to test the conversion methods given in Lemmas 1-3, we implemented a software system which performs these conversions and simulates the movements of converted Turing machines. Examples of the conversion and the simulation results are shown in Ref. (8).

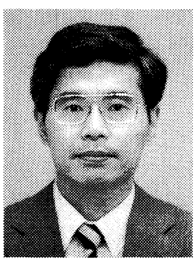
Theorem 1 shows that a 1-tape 2-symbol reversible Turing machine (i. e., in some sense, the simplest form of a reversible machine) is computation universal. This result seems useful to study other reversible systems' computing ability (e. g., reversible cellular automata, or a system constructed by reversible logic elements (such as Fredkin gate[4]), etc.).

Acknowledgement

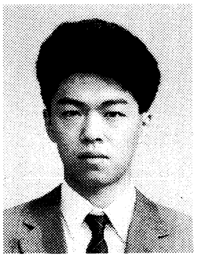
One of the authors K. Morita wishes to thank Professor Masateru Harao of Yamagata University for his helpful discussions and encouragement.

References

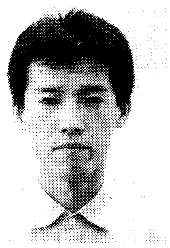
(1) C. H. Bennett: "Logical reversibility of computation", IBM J. Res. & Dev., **17**, 6, pp. 525-532(1973).
(2) C. H. Bennett: "The thermodynamics of computation — A review", Int. J. Theoretical Physics, **21**, 12, pp. 905-940 (1982).
(3) C. H. Bennett and R. Landauer: "The fundamental physical limits of computation", Scientific American, **253**, 1, pp. 38-46(July 1985).
(4) E. Fredkin and T. Toffoli: "Conservative logic", Int. J. Theoretical Physics, **21**, 3/4, pp. 219-253(1982).
(5) Y. Gono and K. Morita: "Construction of a 2-symbol 3-tape reversible Turing machine", Trans. IEICE, **J70-D**, 5, pp. 1047-1050(May 1987).
(6) R. W. Keyes and R. Landauer: "Minimal energy dissipation in logic", IBM J. Res. & Dev., **14**, pp. 152-157(1970).
(7) R. Landauer: "Irreversibility and heat generation in the computing process", IBM J. Res. & Dev., **5**, pp. 183-191 (1961).
(8) K. Morita, A. Shirasaki, and Y. Gono: "A 1-tape 2-symbol reversible Turing machine", IEICE Technical Report, **COMP88**-37(Sept. 1988).



Kenichi Morita was born in Osaka, on March 30, 1949. He received the B. E., M. E., and Dr. E. degrees from Osaka University in 1971, 1973, and 1978, respectively. From 1974 to 1987, he was a Research Associate of the Faculty of Engineering Science, Osaka University. Since 1987 he has been an Associate Professor of the Faculty of Engineering, Yamagata University. He has been engaged in the research of automata theory, computational complexity, formal language theory, and logic systems for knowledge and language processing. Dr. Morita is a member of Information Processing Society of Japan, LA Symposium, Mathematical Linguistic Society of Japan, Japanese Society for Artificial Intelligence, ACM, and ACL.



Akihiko Shirasaki received the B. E. degree from Faculty of Engineering Science, Osaka University in 1987. Since 1987, he is a staff of Dainippon Screen Mfg., Co., Ltd. in Kyoto. In Osaka University, he engaged in the research of automata theory.



Yoshifumi Gono was born in Maizuru, Kyoto Prefecture, on Nov. 10, 1960. He received the B. E. and M. E. degrees from Faculty of Engineering Science, Osaka University in 1985 and 1987, respectively. During this period, he engaged in the research of automata theory. Since 1987, he is a staff of Shinko Electric Industries Co., Ltd. in Nagano.