PAPER

Computation Universality of One-Dimensional Reversible (Injective) Cellular Automata

Kenichi MORITA† *and* Masateru HARAO††, *Members*

SUMMARY A reversible cellular automaton (CA) is a "backward deterministic" CA, i. e., every configuration of it has at most one predecessor. Toffoli showed that a two-dimensional reversible cellular automaton is computation universal. He posed an open problem whether a one-dimensional reversible CA is computation universal. In this paper, we solve this problem affirmatively. This result is proved by using the previous result of Morita et al. that a 1-tape reversible Turing machine is computation universal. We give a construction method of a reversible CA which simulates a given 1-tape reversible Turing machine. To do this, we introduce a "one-dimensional partitioned cellular automaton" (1-PCA). 1-PCA has the property that the local reversibility (i. e., injectivity of a local function) is equivalent to the global reversibility, and thus it facilitates to design a reversible CA.

1. Introduction

A cellular automaton (CA) is called reversible (or injective) if every configuration of it has at most one predecessor (of course, usual CA is in general irreversible). Since Moore[6] proved the "Garden-of-Eden theorem", reversibility in CA, together with its surjectivity, has been studied by many researchers. Reversible CA is important, not only because it has mathematically interesting properties, but also it is considered as an abstract model of a physically reversible space[2],[5].

As for computation universality of a reversible CA, Toffoli[5] showed an interesting result. He proved that every k-dimensional (irreversible) CA can be simulated by a $(k+1)$-dimensional reversible CA. From this, computation universality of a two-dimensional reversible CA is derived, because any Turing machine can be simulated by a one-dimensional CA.

However, the problem whether a one-dimensional reversible CA is computation universal is left open. In this paper, we solve this problem affirmatively. It is proved by embedding a reversible Turing machine in a one-dimensional reversible CA.

A reversible Turing machine is first defined by Bennett[1]. He proved that any (irreversible) Turing machine can be simulated by a 3-tape reversible Turing machine (i. e., a 3-tape reversible Turing machine is computation universal). Previously, by improving Bennett's result, Morita et al.[4]. showed that a 1-tape 2-symbol reversible Turing machine is computation universal. In this paper, we show that any given 1-tape reversible Turing machine can be simulated by a one-dimensional reversible CA.

For this purpose, we introduce a one-dimensional partitioned cellular automaton (1-PCA). 1-PCA is a variant of CA, where each cell is partitioned into the left, center and right parts. State transition in 1-PCA is as follows: the next state of each cell is determined depending on the present states of the right part of the left cell, the center part of the center cell, and the left part of the right cell. Since the domain and the range of the local function have the same finite cardinality, we can define the notion of the local reversibility (i. e., the injectivity of a local function) as well as the (usual) global reversibility. It is shown that these two notions of reversibility are equivalent. Therefore, we can design a globally reversible 1-PCA which simulates a reversible 1-tape Turing machine by giving a reversible local function. In usual CA, it is not so easy to design a reversible one. Since a reversible 1-PCA can be easily simulated by a reversible 1-CA, computation universality of a reversible 1-CA is concluded.

2. Definitions

A deterministic one-dimensional cellular automaton (1-CA) (with the nearest neighborhood) is a system defined by

$$A=(\boldsymbol{Z},\ Q, f_A),$$

where

(1) $\boldsymbol{Z}$ is the set of all integers,

(2) Q is a non-empty finite set of internal states of each cell, and

(3) $f_A : Q^3 \rightarrow Q$ is a mapping called a local function.

A configuration of A is a mapping $c : \boldsymbol{Z} \rightarrow Q$. Let Conf (Q) denotes the set of all configurations over Q, i. e.,

$$\mathrm{Conf}\,(Q)=\{c \mid c : \boldsymbol{Z} \rightarrow Q\}.$$

The function F_A: Conf $(Q) \rightarrow$ Conf (Q) which is defined as follows is called the global function determined by A.

$$F_A(c)(i)=f_A(c(i-1), c(i), c(i+1)) \quad (i \in \boldsymbol{Z})$$

Manuscript received January 27, 1989.

† The author is with the Faculty of Engineering, Yamagata University, Yonezawa-shi, 992 Japan.

†† The author is with the Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Izuka-shi, 820 Japan.

We say A is reversible (or injective) iff F_A is a one-to-one mapping.

A deterministic one-dimensional partitioned cellular automaton (1-PCA) is a system defined by

$$P=(\boldsymbol{Z}, L, C, R, f_P),$$

where
(1) $\boldsymbol{Z}$ is the set of all integers,
(2) L is a non-empty finite set of left internal states of each cell,
(3) C is a non-empty finite set of center internal states of each cell,
(4) R is a non-empty finite set of right internal states of each cell, and
(5) $f_P: R\times C\times L \rightarrow L\times C\times R$ is a mapping called a local function.

A configuration of P is a mapping $c: \boldsymbol{Z} \rightarrow L\times C\times R$. The set of all configurations over $L\times C\times R$ is denoted by Conf $(L\times C\times R)$ as in 1-CA. Let "LEFT" ("CENTER", "RIGHT", respectively) be the projection function which picks out the left (center, right) element of a triple in $L\times C\times R$. The global function F_P : Conf $(L\times C\times R) \rightarrow$ Conf $(L\times C\times R)$ determined by P is defined as follows.

$$F_P(c)(i)=f_P(\mathrm{RIGHT}(c(i-1)), \mathrm{CENTER}\ (c(i)), \mathrm{LEFT}\ (c(i+1))) \qquad (i \in \boldsymbol{Z})$$

We say P is globally reversible iff F_P is a one-to-one mapping. We say P is locally reversible iff the local function f_P is one-to-one.

Intuitively speaking, 1-PCA is a subclass of 1-CA, where each cell is partitioned into three parts (i. e., left, center, and right parts) and the next state of each cell is determined depending on the right part of the left cell, the center part of the center cell, and the left part of the right cell (not depending on the entire three cells).

A 1-tape Turing machine (1-TM) is a system defined by

$$T=(Q, S, q_0, q_f, s_0, M),$$

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) s_0 is a special blank symbol ($s_0 \in S$), and
(6) M is a subset of $Q\times S\times S\times Q \cup Q\times\{/\}\times\{-, 0, +\}\times Q$.

Note that M is a move function in quadruple form. Each quadruple is of the form $[q_r, s, s', q_t]$ or $[q_r, /, d, q_t]$, where $q_r, q_t \in Q$, $s, s' \in S$, and $d \in \{-, 0, +\}$. The symbols "$-$", "0", and "$+$" denote "left-shift", "zero-shift", and "right-shift", respectively. $[q_r, s, s', q_t]$ means that if T reads the symbol s in state q_r, write s' and go to state q_t. $[q_r, /, d, q_t]$ means that if T is in state q_r, shift the head to the direction d and go to state q_t.

Let $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ be two quadruples in M.

$$\boldsymbol{D}_1=[q_{r1}, b_1, c_1, q_{t1}]$$

$$\boldsymbol{D}_2=[q_{r2}, b_2, c_2, q_{t2}]$$

We say that $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ overlap in domain iff

(i) $q_{r1}=q_{r2}$ and $b_1=b_2$, or

(ii) $q_{r1}=q_{r2}$ and b_1 or b_2 is "/".

We say that $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ overlap in range iff

(i) $q_{t1}=q_{t2}$ and $c_1=c_2$, or

(ii) $q_{t1}=q_{t2}$ and b_1 or b_2 is "/".

A quadruple $\boldsymbol{D}$ is said to be deterministic (in M) iff there is no other quadruple in M with which $\boldsymbol{D}$ overlaps in domain. On the other hand, $\boldsymbol{D}$ is said to be reversible (in M) iff there is no other quadruple in M with which $\boldsymbol{D}$ overlaps in range. A Turing machine T is called deterministic iff every quadruple in M is deterministic, and is called reversible iff every quadruple in M is reversible.

3. Properties of 1-PCA

[Lemma 1] Let $P=(\boldsymbol{Z}, L, C, R, f_P)$ be a 1-PCA. P is globally reversible iff P is locally reversible.

(Proof) First, we show the "if" part. Assume P is locally reversible but not globally reversible. Since P is not globally reversible, there are two different configurations c_1 and c_2 such that $F_P(c_1)=F_P(c_2)$. Thus, for any $i \in \boldsymbol{Z}$,

$$F_P(c_1)(i)=F_P(c_2)(i)$$

holds. Therefore,

$$f_P(\mathrm{RIGHT}(c_1(i-1)), \mathrm{CENTER}\ (c_1(i)), \mathrm{LEFT}\ (c_1(i+1)))=f_P(\mathrm{RIGHT}\ (c_2(i-1)), \mathrm{CENTER}\ (c_2(i)), \mathrm{LEFT}\ (c_2(i+1))) \qquad (1)$$

holds for any i (from the definition of the global function). On the other hand, since $c_1 \neq c_2$, there must be $k \in \boldsymbol{Z}$ which satisfies

$\mathrm{RIGHT}\ (c_1(k-1)) \neq \mathrm{RIGHT}\ (c_2(k-1))$, or

$\mathrm{CENTER}\ (c_1(k)) \neq \mathrm{CENTER}\ (c_2(k))$, or

$\mathrm{LEFT}\ (c_1(k+1)) \neq \mathrm{LEFT}\ (c_2(k+1))$.

This contradicts the assumption of local reversibility of P, since Eq. (1) must hold for such k.

Next, we show the "only if" part. Assume P is globally reversible but not locally reversible. Since P is not locally reversible, there are $u_1, u_2 \in R$, $v_1, v_2 \in C$, $w_1, w_2 \in L$, which satisfies

$f_P(u_1, v_1, w_1)=f_P(u_2, v_2, w_2)$, and

$u_1 \neq u_2$ or $v_1 \neq v_2$ or $w_1 \neq w_2$.

Let c_1 and c_2 be two configurations satisfying the following conditions.

$$\text{RIGHT}\ (c_1(-1))=u_1$$
$$\text{CENTER}\ (c_1(0))=v_1$$
$$\text{LEFT}\ (c_1(1))=w_1$$
$$\text{RIGHT}\ (c_2(-1))=u_2$$
$$\text{CENTER}\ (c_2(0))=v_2$$
$$\text{LEFT}\ (c_2(1))=w_2$$
$$\text{RIGHT}\ (c_1(i-1))=\text{RIGHT}\ (c_2(i-1)) \text{ for all } i(\neq 0)$$
$$\text{CENTER}\ (c_1(i))=\text{CENTER}\ (c_2(i)) \text{ for all } i(\neq 0)$$
$$\text{LEFT}\ (c_1(i+1))=\text{LEFT}\ (c_2(i+1)) \text{ for all } i(\neq 0)$$

Apparently $c_1 \neq c_2$. Furthermore,

$$f_P\ (\text{RIGHT}\ (c_1(i-1)), \text{CENTER}\ (c_1(i)), \text{LEFT}\ (c_1(i+1)))=f_P(\text{RIGHT}(c_2(i-1)), \text{CENTER}\ (c_2(i)), \text{LEFT}\ (c_2(i+1))$$

holds for all i because of the above conditions. Thus, $F_P(c_1)=F_P(c_2)$ is concluded, and this contradicts the assumption of global reversibility of P. (Q. E. D.)

The relation between 1-PCA and 1-CA is stated in the following two lemmas.

[Lemma 2] For any 1-PCA $P=(\boldsymbol{Z}, L, C, R, f_P)$, there exists a 1-CA $A=(\boldsymbol{Z}, Q, f_A)$ whose global function F_A agrees with F_P.

(Proof) Q and f_A are defined as follows.

$$Q=L\times C\times R$$
$$f_A((u_{-1}, v_{-1}, w_{-1}), (u_0, v_0, w_0), (u_1, v_1, w_1))$$
$$=f_P(w_{-1}, v_0, u_1)$$
$$(\text{for any } u_{-1}, u_0, u_1\in L, v_{-1}, v_0, v_1\in C,\ w_{-1}, w_0, w_1\in R)$$

From this construction, it is easily seen that the global function F_A agrees with F_P. (Q. E. D.)

[Lemma 3] For any 1-CA $A=(\boldsymbol{Z}, Q, f_A)$, there exists a 1-PCA $P=(\boldsymbol{Z}, L, C, R, f_P)$, which simulates A.

(Proof) L, C, R, f_P are defined as follows.

$$L=Q$$
$$C=Q$$
$$R=Q$$
$$f_P(u, v, w)=[f_A(u, v, w), f_A(u, v, w), f_A(u, v, w)]$$
$$(\text{for any } u, v, w, \in Q)$$

If the initial configuration of A is

$$\cdots\ q_{i-1}\ q_i\ \ q_{i+1}\cdots \qquad (q_i\in Q, \quad i\in\boldsymbol{Z})$$

then that of P is as follows.

$$\cdots[q_{i-1}, q_{i-1}, q_{i-1}][q_i, q_i, q_i][q_{i+1}, q_{i+1}, q_{i+1}]\cdots$$

It is obvious that P correctly simulates A. (Q. E. D.)

(Remarks) In a similar manner as in Lemma 1, we can prove that local surjectivity (i. e., surjectivity of a local function) and global surjectivity (i. e., surjectivity of a global function) in 1-PCA are equivalent. Furthermore, since the domain and the range of a local function of 1-PCA have the same finite cardinality, local surjectivity and local reversibility are also equivalent.

Therefore, in 1-PCA, the notions of global surjectivity and global reversibility coincide. On the other hand, it has been known that, in 1-CA, while reversibility implies surjectivity, its converse does not hold (see e. g. (7)). Indeed, there is a 1-CA A which is surjective but not reversible.

Lemmas 2 and 3 show that 1-CA and 1-PCA can simulate each other. Especially, 1-CA can simulate 1-PCA in the strong sense (i. e., with the same global function). However, 1-PCA cannot simulate 1-CA in the strong sense in general. If otherwise, there must be a 1-PCA P which is surjective but not reversible, and this contradicts the equivalence of surjectivity and reversibility in 1-PCA.

4. Computation Universality of a Reversible 1-CA

As for reversible multi-dimensional cellular automata, Toffoli[5] has been shown the following results.

[Proposition 1] [5] An arbitrary cellular automaton having d dimensions is embeddable in a reversible one having $d+1$ dimensions.

Next corollary is derived from Proposition 1 and the well known result that a one-dimensional cellular automaton can simulate any Turing machine.

[Corollary 1] [5] A two-dimensional reversible cellular automaton is computation universal.

On the other hand, as for a reversible Turing machine, Bennett[1] proved the following result.

[Proposition 2] [1] For any irreversible Turing machine T, there is a 3-tape many-symbol reversible Turing machine R which simulates T.

By improving Bennett's method, Morita et al.[3],[4]. proved the computation universality of simpler Turing machines.

[Proposition 3] [4] For any irreversible Turing machine T, we can construct a 1-tape 2-symbol reversible Turing machine R which simulates T.

We now show that any reversible 1-TM can be embedded in a reversible 1-PCA.

[Lemma 4] For any deterministic reversible 1-TM $T=(Q, S, q_0, q_f, s_0, M)$ (which may not be 2-symbol), there exists a locally reversible 1-PCA $P=(\boldsymbol{Z}, L, C, R, f_P)$ which simulates T.

(Proof) L, C, and R, are defined as follows.

$$L=Q\cup\{*,\ \#\}$$

$C=(Q\cup\{*,\ \#\})\times S$

$R=Q\cup\{*,\ \#\}$

Next, we define the local function f_P, which should be one-to-one. As for a one-to-one mapping on a finite set, the following proposition holds (the proof is obvious).

[Proposition] Let A and B be finite sets with the same cardinal numbers, and let g be a mapping $A' \to B$, where A' is a subset of A. If g is one-to-one, then threre exists a one-to-one mapping $g': A \to B$ which is an extension of g.

Therefore, it is sufficient to define the values of f_P only on a subset of $R\times C\times L$ (that are needed to simulate T) on which f_P is assured to be one-to-one.

[1] If $[p, s, t, q]\in M$, then
$f_P([\#, [p, s], \#])=[\#, [q, t], \#]$.

[2] If $[p, /, -, q]\in M$, then
$f_P([\#, [p, s], \#])=[q, [\#, s], \#]$ for all $s\in S$,
and
$f_P[(\#,\ [\#, t], q])=\ [\#, [q, t], \#]$ for all $t\in S$.

[3] If $[p, /, 0, q]\in M$, then
$f_P([\#, [p, s], \#])=\ [\#, [q, s], \#]$ for all $s\in S$.

[4] If $[p, /, +, q]\in M$, then
$f_P([\#, [p, s], \#])=[\#, [\#, s], q]$ for all $s\in S$,
and
$f_P([q, [\#, t], \#])=[\#, [q, t], \#]$ for all $t\in S$.

[5] For all $s\in S$,
$f_P([\#, [\#, s], \#])=[\#, [\#, s], \#]$.

[6] For all $s\in S$,
$f_P([*, [*, s], \#])=[\#, [q_0, s], \#]$,
$f_P([\#, [q_f, s], \#])=[*, [*, s], \#]$,
$f_P([*, [\#, s], \#])=[\#, [\#, s], *]$,
$f_P([\#, [\#, s], *])=[*, [\#, s], \#]$, and
$f_P([\#, [*, s], \#])=[\#, [*, s], \#]$.

It can be verified (by a careful inspection) that f_P satisfies one-to-one-ness because T is deterministic and reversible.

The rules in [1] - [5] are for simulating the movements of T. The rules in [6] keep the final tape (the result of a computation) of T unchanged after T halts. (To be symmetric, the rules for keeping the initial tape in the negative time direction are also added in [6].)

If the initial computational configuration of T is

$$\cdots s_0 t_1 \cdots \underset{\substack{\uparrow \\ q_0}}{t_i} \cdots t_n s_0 \cdots$$

then set P to the following initial configuration.

$\cdots, [\#, [\#, s_0], \#], [\#, [\#, t_1], \#], \cdots, [\#, [q_0, t_i], \#], \cdots$
$\cdots, [\#, [\#, t_n], \#], [\#, [\#, s_0], \#], \cdots$

It is easily seen that, from this configuration, P can correctly simulates T step by step (An example is shown in the appendix). (Q. E. D.)

From Lemmas 1, 2, 4, and Proposition 3, the following theorem, which states computation universality of a reversible 1-CA, is derived.

[Theorem 1] For any 1-TM T, there is a reversible 1-CA A which simulates T.

5. Conclusion

We have shown that any Turing machine can be simulated by a one-dimensional reversible cellular automaton. This means that one-dimensional reversible space supports the universal computing ability.

The problem whether we can make this reversible space more simple (e. g., reducing the number of states of a cell, etc.) is left to the future study.

References

(1) C. H. Bennett: "Logical reversibility of computation", IBM J. Res. & Dev., 17, **6**, pp. 525-532 (1973).

(2) E. Fredkin and T. Toffoli: "Conservative logic", Int. J. of Theoretical Physics, **21**, 3/4, pp. 219-253 (1982).

(3) 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).

(4) K. Morita, A. Shirasaki and Y. Gono: "A 1-tape 2-symbol reversible Turing machine", Trans. IEICE, **E72**, pp. 223-228 (March 1989).

(5) T. Toffoli: "Computation and construction universality of reversible cellular automata", J. Comput. & Syst. Sci., **15**, pp. 213-231 (1977).

(6) E. F. Moore: "Machine models of self-reproduction", AMS Proc. 14 th Symp. Appl. Math., pp. 17-33 (1962).

(7) D. Richardson: "Tessellations with local transformations", J. Comput. & Syst. Sci., **6**, pp. 373-388 (1972).

Appendix: An Example

Consider a reversible 1-TM T which is defined by the quadruple set in Fig. A•1. T is a machine which copies a given unary number on the tape (see Fig. A•2). Applying the conversion procedure given in Lemma 4, we can obtain a reversible 1-PCA P which simulates T. Fig. A•3 shows the local function of P, and Fig. A•4 shows the computation process of P.

```
(1)  [q0,0,0,q1]
(2)  [q1,/,+,q2]
(3)  [q2,0,0,qf]
(4)  [q2,1,0,q3]
(5)  [q3,/,+,q4]
(6)  [q4,0,0,q5]
(7)  [q4,1,1,q3]
(8)  [q5,/,+,q6]
(9)  [q6,0,1,q7]
(10) [q6,1,1,q5]
(11) [q7,/,+,q8]
(12) [q8,0,0,q9]
(13) [q9,/,-,qa]
(14) [qa,0,0,qb]
(15) [qa,1,1,q9]
(16) [qb,/,-,qc]
(17) [qc,0,1,q1]
(18) [qc,1,1,qb]
```

Fig. A•1 The quadruple set of the reversible 1-TM T.

```
t=0    00110000
       ↑
       q0

t=1    00110000
        ↑
        q1

t=2    00110000
         ↑
         q2

        .
        .

t=39   00110110
          ↑
          qf
```

Fig. A•2 A computation of T.

(0)	[#,[#,0],#]	→ [#,[#,0],#]
	[#,[#,1],#]	→ [#,[#,1],#]
	[*,[*,0],#]	→ [#,[q0,0],#]
	[*,[*,1],#]	→ [#,[q0,1],#]
	[#,[qf,0],#]	→ [*,[*,0],#]
	[#,[qf,1],#]	→ [*,[*,1],#]
	[*,[#,0],#]	→ [#,[#,0],*]
	[*,[#,1],#]	→ [#,[#,1],*]
	[#,[#,0],*]	→ [*,[#,0],#]
	[#,[#,1],*]	→ [*,[#,1],#]
	[#,[*,0],#]	→ [#,[*,0],#]
	[#,[*,1],#]	→ [#,[*,1],#]
(1)	[#,[q0,0],#]	→ [#;[q1,0],#]
(2)	[#,[q1,0],#]	→ [#,[#,0],q2]
	[#,[q1,1],#]	→ [#,[#,1],q2]
	[q2,[#,0],#]	→ [#,[q2,0],#]
	[q2,[#,1],#]	→ [#,[q2,1],#]
(3)	[#,[q2,0],#]	→ [#,[qf,0],#]
(4)	[#,[q2,1],#]	→ [#,[q3,0],#]
(5)	[#,[q3,0],#]	→ [#,[#,0],q4]
	[#,[q3,1],#]	→ [#,[#,1],q4]
	[q4,[#,0],#]	→ [#,[q4,0],#]
	[q4,[#,1],#]	→ [#,[q4,1],#]
(6)	[#,[q4,0],#]	→ [#,[q5,0],#]
(7)	[#,[q4,1],#]	→ [#,[q3,1],#]
(8)	[#,[q5,0],#]	→ [#,[#,0],q6]
	[#,[q5,1],#]	→ [#,[#,1],q6]
	[q6,[#,0],#]	→ [#,[q6,0],#]
	[q6,[#,1],#]	→ [#,[q6,1],#]
(9)	[#,[q6,0],#]	→ [#,[q7,1],#]
(10)	[#,[q6,1],#]	→ [#,[q5,1],#]
(11)	[#,[q7,0],#]	→ [#,[#,0],q8]
	[#,[q7,1],#]	→ [#,[#,1],q8]
	[q8,[#,0],#]	→ [#,[q8,0],#]
	[q8,[#,1],#]	→ [#,[q8,1],#]
(12)	[#,[q8,0],#]	→ [#,[q9,0],#]
(13)	[#,[q9,0],#]	→ [qa,[#,0],#]
	[#,[q9,1],#]	→ [qa,[#,1],#]
	[#,[#,0],qa]	→ [#,[qa,0],#]
	[#,[#,1],qa]	→ [#,[qa,1],#]
(14)	[#,[qa,0],#]	→ [#,[qb,0],#]
(15)	[#,[qa,1],#]	→ [#,[q9,1],#]
(16)	[#,[qb,0],#]	→ [qc,[#,0],#]
	[#,[qb,1],#]	→ [qc,[#,1],#]
	[#,[#,0],qc]	→ [#,[qc,0],#]
	[#,[#,1],qc]	→ [#,[qc,1],#]
(17)	[#,[qc,0],#]	→ [#,[q1,1],#]
(18)	[#,[qc,1],#]	→ [#,[qb,1],#]

Fig. A･3 The local function of the reversible 1-PCA P which simulates T (it is written in the form $X \to Y$ instead of $f_P(X)=Y$).

t=0	# # # 0	# q0 # 0	# # # 1	# # # 1	# # # 0	# # # 0	# # # 0	# # # 0
t=1	# # # 0	# q1 # 0	# # # 1	# # # 1	# # # 0	# # # 0	# # # 0	# # # 0
t=2	# # # 0	# # q2 0	# # # 1	# # # 1	# # # 0	# # # 0	# # # 0	# # # 0
t=3	# # # 0	# # # 0	# q2 # 1	# # # 1	# # # 0	# # # 0	# # # 0	# # # 0
:								
t=58	# # # 0	# # # 0	# # # 1	# # # 1	# qf # 0	# # # 1	# # # 1	# # # 0
t=59	# # # 0	# # # 0	# # # 1	# # # 1	* * # 0	# # # 1	# # # 1	# # # 0

Fig. A･4 A computation process of P which simulates T.

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 Japan, Mathematical Linguistic Society of Japan, Japanese Society for Artificial Intelligence, ACM, and ACL.

Masateru Harao received the B. E. degree in the electrical engineering from the Kyushu Institute of Technology, in 1966, and the M. E. and the D. E. degrees in electrical and communication engineering from Tohoku University, in 1969 and 1972 respectively. In 1972, he joined with the Research Institute of Electrical Communication, Tohoku University, where he was a research assistant from 1972 to 1976, and associate professor from 1976 to 1984, respectively. From 1984 to 1989, he was a professor in the department of information engineering at Yamagata University, Yonezawa. Since 1989, he is a professor in the department of artificial intelligence, Kyushu Institute of Technology, Izuka. His current research interests include computation theory, parallel processing and AI, especially knowledge information processing based on logics. Prof. Harao is a member of the Information Processing Soc. of Japan, Japanese Soc. for AI, Japan soc. for Software science and Technology and IEEE Computer Society.