INFORMATION AND CONTROL 15, 353-367 (1969)

The Uniform Halting Problem for Generalized One-State Turing Machines*

GABOR T. HERMAN

IBM (U.K.) Ltd., Education Centre, Greenford, Middlesex, England†

It is shown that the uniform halting problem for one-state Turing machines is solvable. It remains solvable for various generalizations like one-state Turing machines with two-dimensional tape and jumping reading head. Other generalizations, for example, one-state Turing machines with two tapes, have an unsolvable uniform halting problem. The history of the problem is summarized.

INTRODUCTION

The uniform halting problem (UH) can be stated as follows:

Give a decision procedure which for any given Turing machine (TM) will decide whether or not it has an immortal instantaneous description (ID).

An ID is called immortal if it has no terminal successor. As it is generally the case in the literature (see, e.g., Minsky, 1967, p. 118), we assume that in an ID the tape must be blank except for some finite number of squares. If we remove this restriction, the UH becomes the immortality problem (IP).

The unsolvability of the IP was shown by Hooper (1966). It can be seen from Part V (p. 225) of his paper that his method also shows the unsolvability of the UH. A much simpler solution of the unsolvability of UH can be found in Herman (1970). The initialized UH, whether or not a TM has an immortal ID when started in a specified state, is also known to be undecidable (see, e.g., Minsky, 1967, p. 151). For one-state TM's the two problems are of course equivalent. Hooper (1966, Part VI, 5) found that the IP is solvable for two-state TM's. However, one

* An earlier version of this paper was presented at the 1968 Ninth Annual Symposium for Switching and Automata Theory.

† Present address: Department of Computer Science, State University of New York at Buffalo, Buffalo, N. Y. 14214.

can show (Herman, 1969) that the UH (just like the halting problem) is unsolvable for two state TM's.

The halting problem is known to be solvable for one-state TM's, even if we allow them to have a many-dimensional tape (see Herman, 1968). Can the same be said about the UH?

In this paper we show the solvability of the UH for one-state TM's. These are not entirely trivial structures. As we show, they can do certain things that finite automata cannot do.

We consider in what ways we can generalize one-state TM's so that we retain the solvability of the UH. We find that the UH for one-state TM's with two-dimensional tape and jumping reading head (i.e., the reading head can move to squares not immediately neighboring the previously scanned square) is also solvable. Our method does not generalize for three-or-more-dimensional tape, unless we put in some conditions on the directions in which the reading head can move.

On the other hand, we find that one-state TM's with two or more tapes or with two or more reading heads on the same tape have an unsolvable UH.

Finally we consider how a different choice of formalism would influence our results. (Where it is not otherwise stated we shall assume the original quintuple formalism for TM's, discussed, e.g., by Fischer, 1965, Section 2.)

1. THE COMPUTING POWER OF ONE-STATE TM's

At first sight, one-state TM's appear to be very trivial. The next action of such a machine is completely determined by the symbol in the square it is scanning.

In spite of this, one-state TM's have some computing power. Let us call a string S of symbols "accepted" by a one-state TM, if the ID in which the tape has nothing but S on it and first symbol of S is scanned is mortal.

There exists a one-state TM such that among all the strings of the form Aa^nb^mB, exactly those are accepted by this machine for which $n < m$. Such a machine will have the table:

A	C	R	a	c	R
b	b'	L	c	c'	R
b'	b''	R	c'	c''	L
b''	b'	L	c''	c'	R
B	B	N	C	halt	

It can easily be shown that there exists no finite, single-headed automaton which among all the strings of the form Aa^nb^mB will accept exactly those for which $n < m$. So one-state TM's can do certain things that finite automata cannot do.

2. THE SOLVABILITY OF THE UH FOR ONE-STATE TM'S

The uniform halting problem for one-state Turing machines is solvable.

Proof. Let T be the given TM.

Case α. Once T scans an empty square, it will halt either without leaving that square at all or at the end of the step, when it leaves that square for the first time.

In this case an ID can be immortal only if neither in it nor in any of its successors is the scanned square empty. Since in all ID's all but finitely many of the squares are empty, in any immortal ID there must be two squares (which may coincide) such that they are both scanned infinitely often and all squares outside the segment enclosed by them are scanned at most finitely often.

Let us call a symbol t to be a successor of a symbol s, if, whenever a square contains s, it will also contain t, provided it is scanned often enough. (E.g., the successors of a in the machine of Section 1 are a, c, c', c''.) Then the above argument gives that there can only be an immortal ID if there exists a symbol in the alphabet of T seeing which, or any of its successors, the machine never moves to the right and never halts and another symbol seeing which, or any of its successors, the machine never moves to the left and never halts. (B in the machine of Section 1 is of both these types.)

But if such symbols exist (since there are only finitely many symbols in the alphabet, we can easily find out whether or not this is the case), we can produce an immortal ID as follows.

Let a be a symbol seeing which, or any successor of which, the machine never moves to the left and never halts and b be a symbol seeing which, or any successor of which, the machine never moves to the right and never halts. Then the ID in Fig. 1 is clearly immortal.

Let us summarize for Case α. The TM T has an immortal description if and only if its alphabet has at least one symbol of type a and at least one symbol of type b.

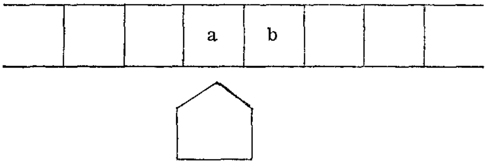


FIG. 1. Immortal ID for one-state TM

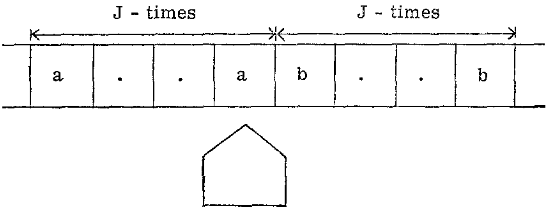


FIG. 2. Immortal ID for one-state TM with jumping reading head

Case β. Once the machine scans an empty square, it either
(i) will never leave this square and never halt, or,
(ii) will sooner or later scan another square in its non-halting state.

In this case T has an immortal ID, namely, the empty tape.

Since we can easily decide whether α or β is the case, and since these cases exhaust all possibilities, our algorithm is complete.

3. GENERALIZATION FOR ONE-STATE TM'S WITH JUMPING READING HEAD

In a TM with jumping reading head, the reading head is allowed to move to squares not immediately neighboring the one presently scanned. The argument of Section 2 can be repeated for this case as well, except at one point. In case α, given two symbols of type a and b, an immortal ID is given by Fig. 2. (J is the greatest number of squares the reading head may jump.)

It is interesting to note that the solvability of the halting problem for such machines is an open question.

4. GENERALIZATION FOR ONE-STATE TM'S WITH TWO-DIMENSIONAL TAPE (NOT JUMPING)

A two-dimensional TM is one whose tape is a plane divided into equal squares. At any time all but finitely many of the squares are empty. During a step (which in the case of a one-state, two-dimensional TM

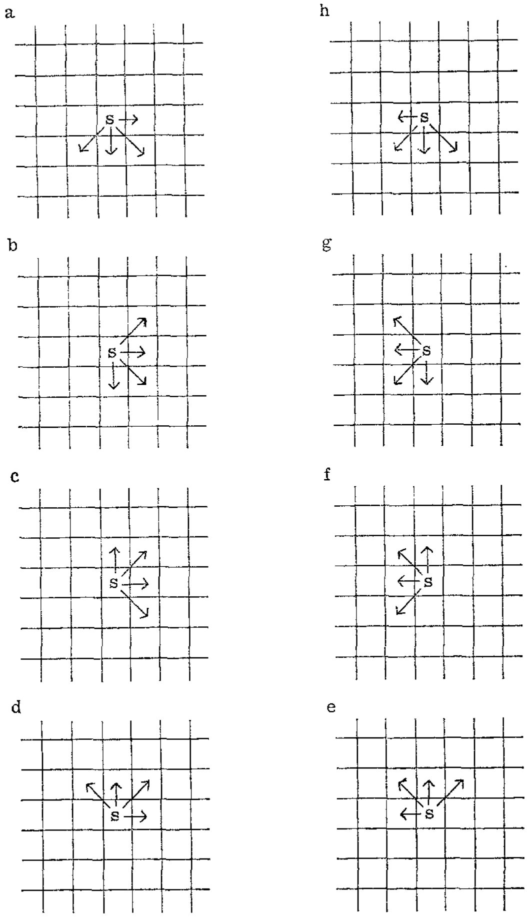
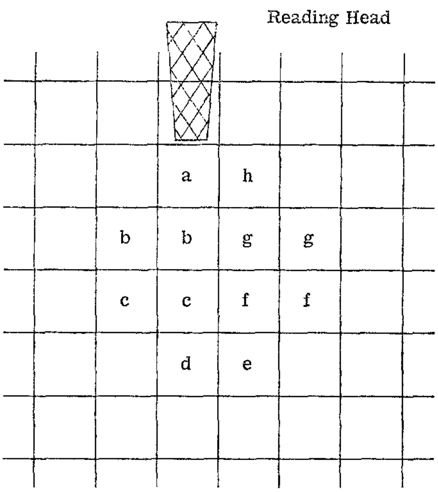


FIG. 3. The eight corner types

is entirely determined by the scanned symbol), the machine may move to any of the eight neighboring squares. The uniform halting problem is solvable for such machines.

Proof. For this proof we need a new notion, that of a *corner type.* Given a Turing machine T, a symbol s of T is said to be of a certain corner type if and only if it is such that when it or any of its successors is scanned, the machine will not move into a square other than those indicated by arrows in Fig. 3, nor will it halt at the end of the step.



		a	h		
	b	b	g	g	
	c	c	f	f	
		d	e		

FIG. 4. Immortal ID for one-state TM's with two-dimensional tape

We distinguish between Cases α and β as in Section 2.

In Case α, if there is an immortal ID, then the alphabet of T must contain at least one symbol of each of the eight corner types. On the other hand, if the alphabet of T contains symbols a, b, c, d, e, f, g, and h of the eight corner types described in Fig. 3, then the ID in Fig. 4 is clearly immortal. The choice of the symbols within the array is to a large extent, but not altogether, arbitrary. There will in most cases be other immortal ID's. For instance, the internal symbol of type b can always be replaced by a symbol of type a but cannot in general be replaced by a symbol of type f.

In case β we again have a trivial immortal ID.

5. GENERALIZATION FOR ONE-STATE TM'S WITH TWO-DIMENSIONAL TAPE AND JUMPING READING HEAD

A jumping two-dimensional TM is a two-dimensional TM such that the machine is allowed to move to squares not immediately neighboring the one presently scanned.

It is an open question whether or not the halting problem for one-state jumping two-dimensional TM's is solvable. However, the uniform halting problem for such machines is solvable.

The proof here is very similar to that in Section 4 except that the number of corner types is increased. First we find out what is the greatest

TABLE 1

Corner Types With $J = 1$

a	b	c	d	e	f	g	h
$\frac{1}{0}$	$\frac{1}{1}$	$\frac{0}{1}$	$\frac{-1}{1}$	$\frac{-1}{0}$	$\frac{-1}{-1}$	$\frac{0}{-1}$	$\frac{1}{-1}$

TABLE 2

Corner Types With $J = 2$

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
$\frac{2}{0}$	$\frac{2}{1}$	$\frac{1}{1}$	$\frac{1}{2}$	$\frac{0}{2}$	$\frac{-1}{2}$	$\frac{-1}{1}$	$\frac{-2}{1}$	$\frac{-2}{0}$	$\frac{-2}{-1}$	$\frac{-1}{-1}$	$\frac{-1}{-2}$	$\frac{0}{-2}$	$\frac{1}{-2}$	$\frac{1}{-1}$	$\frac{2}{1}$

possible horizontal or vertical component (denoted by J) of a jump for the machine in question. From that we can prepare a table of all possible corner types.

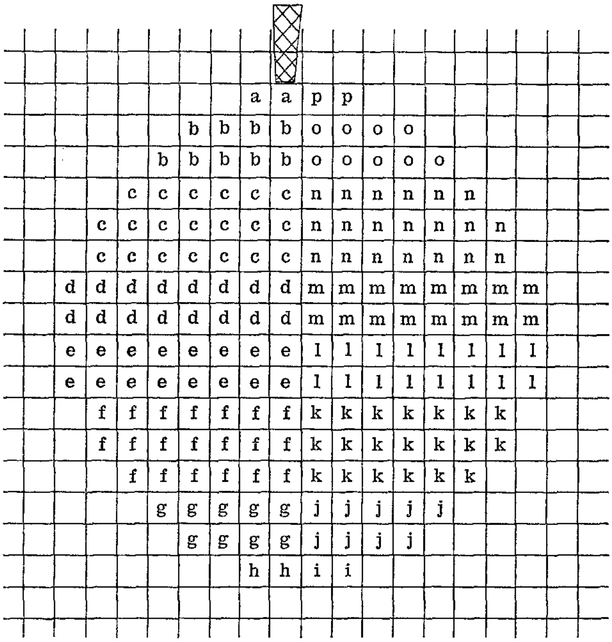
A corner type is essentially a direction in which the machine is allowed to move together with all the directions in the clockwise direction from the given one up to, but excluding, the direction opposite to the given one.

Using this definition, corner type h in Section 4 is the direction $1/-1$, together with the directions $0/-1$, $-1/-1$, and $-1/0$. It can be said that this corner type is determined by the direction $1/-1$.

The eight corner types in Section 4 are determined by the directions given in Table 1.

Similarly, if the greatest possible horizontal or vertical component of a jump for the given machine is $J = 2$, all the possible corner types are determined by Table 2.

If a one-state jumping two-dimensional TM has at least one symbol of each of the corner types a–p (i.e., a symbol such that whenever it or any of its successors is scanned, the machine will not move into a direc-



								▨									
							a	a	p	p							
					b	b	b	b	o	o	o	o					
				b	b	b	b	b	o	o	o	o	o				
			c	c	c	c	c	c	n	n	n	n	n	n			
		c	c	c	c	c	c	c	n	n	n	n	n	n	n		
		c	c	c	c	c	c	c	n	n	n	n	n	n	n		
	d	d	d	d	d	d	d	d	m	m	m	m	m	m	m	m	
	d	d	d	d	d	d	d	d	m	m	m	m	m	m	m	m	
	e	e	e	e	e	e	e	e	l	l	l	l	l	l	l	l	
	e	e	e	e	e	e	e	e	l	l	l	l	l	l	l	l	
		f	f	f	f	f	f	f	k	k	k	k	k	k	k		
		f	f	f	f	f	f	f	k	k	k	k	k	k	k		
			f	f	f	f	f	f	k	k	k	k	k	k			
				g	g	g	g	g	j	j	j	j	j				
					g	g	g	g	j	j	j	j					
							h	h	i	i							

FIG. 5. Immortal ID for one-state TM with two-dimensional tape and $J = 2$

tion other than a direction of the corner type, nor will it halt at the end of the step), then it has an immortal ID (see Fig. 5).

Here the symbol of corner type f, for example, is such that when it or any of its successors is scanned, the machine will not move into a square other than those indicated by an arrow in Fig. 6.

As shown above, there are eight corner types if $J = 1$.

If n is a positive integer, let us denote by $\phi(n)$ the number of positive integers less than or equal to n which are prime to n (see, e.g., Hunter, 1964, p. 32). Then $\phi(1) = 1, \phi(2) = 1, \phi(3) = 2, \phi(4) = 2, \phi(5) = 4$, etc.

Let $\Psi(J)$ denote the number of corner types for any particular J. We now show that

$$\Psi(J) = 8 \cdot \sum_{n=1}^{J} \phi(n).$$

For $J = 1$, this is shown above.

Assume the theorem is true for J. All corner types for J will be corner types for $J + 1$.

All new corner types must be determined by one of the $8 \cdot (J + 1)$ directions

$$\frac{x}{J+1}, \frac{-x}{J+1}, \frac{x}{-(J+1)}, \frac{-x}{-(J+1)}, \frac{J+1}{x}, \frac{J+1}{-x}, \frac{-(J+1)}{x}, \frac{-(J+1)}{-x},$$

where $1 \leq x \leq J + 1$. If x is relatively prime to $J + 1$, all the eight directions above are new, if x is not relatively prime to $J + 1$, none of them are new. Hence,

$$\Psi(J + 1) = \Psi(J) + 8 \cdot \phi(J + 1) = 8 \cdot \sum_{n=1}^{J+1} \phi(n).$$

By induction this completes the proof of our claim. This formula gives that $\Psi(2) = 16$, $\Psi(3) = 32$, $\Psi(4) = 48$, $\Psi(5) = 80$, etc.

We distinguish between Cases α and β as in Section 2. In Case α, there is an immortal ID of T if and only if the alphabet of T contains at least one symbol of each of the $\Psi(J)$ corner types. In Case β, there is a trivial immortal ID of T.

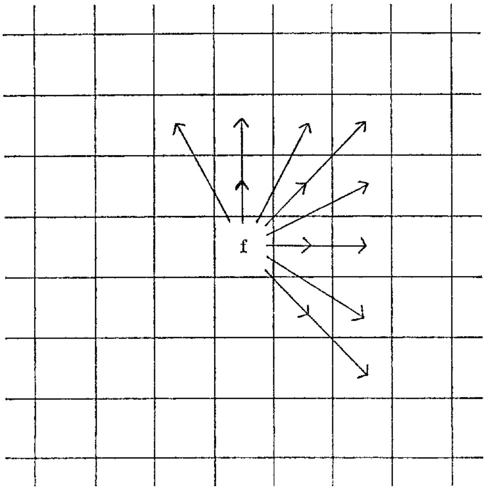


FIG. 6. The corner Type f for $J = 2$

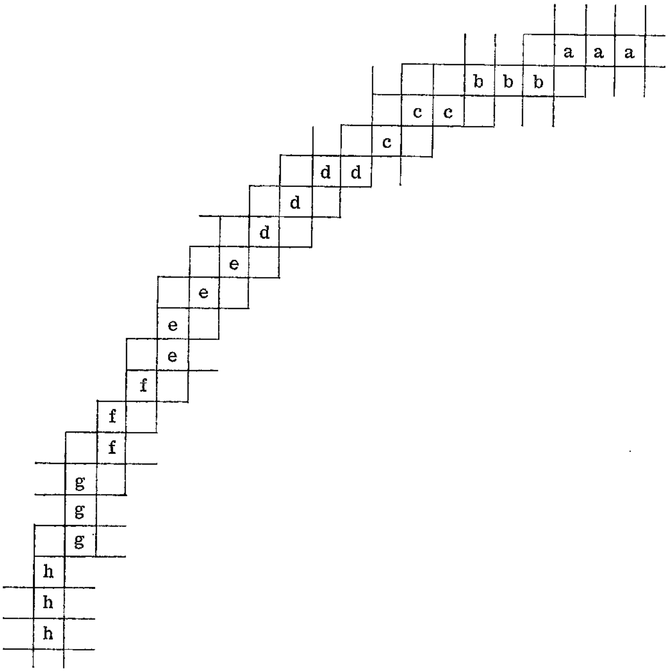


FIG. 7. A quarter of the perimeter of an immortal ID for one-state TM with two-dimensional tape and $J = 3$.

Figure 7 shows the perimeter of a quarter of an immortal ID with $J = 3$.

6. GENERALIZATION FOR ONE-STATE TM'S WITH MANY-DIMENSIONAL TAPE

Although the concept of corner type generalizes easily for three or more dimensions, the proof above does not. This is because it is impossible to construct a solid made up from cubes which corresponds to Fig. 8, i.e., in which each cube has the property that the directions drawn from it to neighboring cubes has a corner type as a subset. In view of the lack of regular solids, this is not surprising.

Hence, this method of corner types is not very satisfactory. A more general (and precise) approach is still to be found.

It may be pointed out here, that if we allow moves only in the main directions (i.e., parallel to the edges of the hypercubes), the method for the two-dimensional case becomes simpler (the number of corner types reduces to four) and it will also become immediately generalizable (e.g., eight three-dimensional corner types will make up a cube).

7. THE UNSOLVABILITY OF THE UH FOR ONE-STATE TM'S WITH TWO OR MORE TAPES

We can simulate the working of a TM T by a one-state TM T_2 with two tapes as follows.

The alphabet of T_2 is the same as the alphabet of T plus additional symbols, one for each state of T. If T_2 encounters a state symbol on its first tape or a non-state symbol on its second tape, it halts. Otherwise, it acts on the first tape as T would act on its tape when encountering the same symbol and being in the state indicated by the scanned symbol on the second tape. This symbol on the second tape is changed into the symbol representing the next state of T.

It is clear that T_2 has immortal ID if and only if T has one, and so the unsolvability of the uniform halting problem for TM's implies that for one-state TM's with two or more tapes.

8. THE UNSOLVABILITY OF THE UH FOR ONE-STATE TM'S WITH TWO READING HEADS

Given any Turing machine T, we first produce a Turing machine T^1, which simulates T but only uses half of the computing tape. We do this by numbering the squares of T as in Fig. 9, and making the squares on the right half of the tape of T^1 correspond to them as in Fig. 10.

The alphabet of T^1 is the same as that of T, except that it is repeated twice, once with the suffix 0 to indicate the symbol in square 0. The number of states in T^1 is six times that of T, the extra states being used to show whether the scanned square is on the left or right of 0 on the original tape and also to help to move two squares to the originally

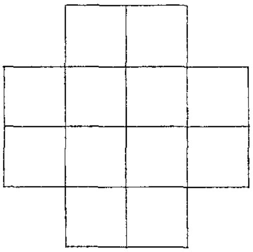


FIG. 8. The shape of the ID in Fig. 4

. . .	-4	-3	-2	-1	0	1	2	3	4	. . .

FIG. 9. Numbering of the squares of T

. . .					0	-1	1	-2	2	-3	3	-4	4	. . .

FIG. 10. Numbering of the squares of T^1

adjacent symbol. By causing T^1 to halt in all ID's which could not possibly correspond to an ID of T, we can make sure that T^1 has an immortal ID if and only if T has one. We give a detailed proof of this construction in the Appendix.

Now we can produce a one-state TM T^2 with two reading heads, which simulates T^1 just as T_2 simulated T in Section 7 (the left-hand reading head is used for scanning state symbols). T^2 will have an immortal ID if and only if T has one, and hence the UH for one-state Turing machines with two reading heads is unsolvable.

9. THE UH FOR SMALL POST MACHINES

All the above results have been established for the original quintuple formalism of Turing machines (see, e.g., Fischer, 1965, Section 2). We now consider what happens if we use the more restrictive quadruple formalism. Following Fischer (1965, Section 3), we shall refer to machines in the quadruple formalism as Post machines (PM's).

It is clear that the solvability of the UH for one-state PM's with two-dimensional tape and jumping reading head follows from our result. It has been shown (Herman, 1969) that the UH is unsolvable for three-state PM's.

Aanderaa and Fischer (1967) show that PM's differ from TM's inasmuch as two-state PM's have a solvable halting problem, while two-state TM's do not (see Shannon, 1965).

Whether this difference also exists in the case of the UH as well is not known.

The property of one-state TM's that the symbol in a square is determined by the symbol initially in that square, and the number of times that square has been scanned is more or less shared by two-state PM's (see Aanderaa and Fischer, 1967, Lemma 4). This and other properties of two-state PM's lead us to conjecture that the UH for two-state PM's is solvable, but a correct proof is still to be found.

APPENDIX: THE SIMULATION OF TM'S BY HALF-TAPE TM'S WITHOUT INTRODUCING IMMORTAL ID'S

Every TM can be simulated by a TM in which the fourth element of a quintuple is never N. Furthermore, this simulation can be made to be such that the simulating machine will have an immortal ID if and only if the simulated machine has one. This fairly obvious observation is made even clearer by the considerations of Fischer (1965, Section 2). For the rest of the appendix we shall assume that in a TM the fourth element of every quintuple is either R or L. For notational convenience we sometimes denote R by D_1, and L by D_2.

Let T be a given TM with states $q_1, \cdots, q_n$ and symbols $S_0, S_1, \cdots, S_{m-1}$, where S_0 is the blank. Then the TM T^1, which simulates it using only half-tape, will have states $q_1, \cdots, q_{6n}$ and symbols $S_0, S_1, \cdots, S_{m-1}, S_{0,0}, S_{1,0}, \cdots, S_{m-1,0}$.

For every quintuple

$$\begin{array}{lllll} q_i & S_j & S_k & D_t & q_l \end{array}$$

of T, T^1 will have the quintuples

$$\begin{array}{rlllll}
\text{(i)} & q_i & S_j & S_k & D_t & q_{tn+l} \\
\text{(ii)} & q_{3n+i} & S_j & S_k & D_{3-t} & q_{(6-t)n+l} \\
\text{(iii)} & q_{2n+i} & S_{j,0} & S_{k,0} & R & q_{4(t-1)t+l} \\
\text{(iv)} & q_{3n+i} & S_{j,0} & S_{k,0} & R & q_{4(t-1)n+l}
\end{array}$$

The T^1 has additional quintuples of the form

$$\begin{array}{rlllll} \text{(v)} & q_{(t+u)n+i} & S_j & S_j & D_t & q_{un+i} \end{array}$$

for $t \in \{1, 2\}, u \in \{0, 3\}, 1 \leq i \leq n, 0 \leq j \leq m - 1$.

Given any ID of T, it can be described by an expression E of the form

$$q_i\ ;\quad S_{j_{-v}} \cdots S_{j_{-2}} S_{j_{-1}} \underline{S_j} S_{j_1} S_{j_2} \cdots S_{j_v}\ ,$$

where q_i is the state and the scanned symbol is underlined. The corresponding ID of T^1 is described by the expression E^1

$$q_{n+i}\ ;\quad \underline{S_{j,0}} S_{j_1} S_{j_{-1}} S_{j_2} S_{j_{-2}} \cdots S_{j_v} S_{j_{-v}}\ .$$

First of all, it is easy to see that T^1 will never scan a square to the left of the one scanned in E^1.

Secondly, T^1 simulates T. The initially scanned square of E and the 1st, 2nd, 3rd, etc., squares on the left of it, correspond to the initially scanned square of E^1 and the 2nd, 4th, 6th, etc., squares on the right of it. The 1st, 2nd, 3rd, etc., squares on the right of the initially scanned square of E correspond to the 1st, 3rd, 5th, etc., squares on the right of the initially scanned square of E^1.

The simulation is step by step. The simulation of one step in T is generally done by two steps in T^1. We know whether T is scanning a square on the right of the initially scanned square in the simulated computation, because the state q_i of T^1 will at the corresponding point in the simulating computation be such that $1 \leq i \leq n$, while otherwise it will be such that $3n + 1 \leq i \leq 4n$. Now the machine will have to move over two squares (see lines (i), (ii), and (v)) to get to the square corresponding to the next scanned square, and it does this using states q_i in the ranges $n + 1 \leq i \leq 3n$ and $4n + 1 \leq i \leq 6n$. The only exceptional situation is if the originally scanned square gets scanned again. This possibility is taken care of by lines (iii) and (iv).

This discussion shows that T can be simulated by a TM which never moves to the left of its initially scanned square. This could have been proved in a much simpler way, but the above method provides us with an important additional result, the one required in Section 8.

T^1 will have an immortal ID, if and only if, T has one.

Proof. Since T^1 simulates T, if T has an immortal ID, so has T^1.

Conversely, let E^1 describe an immortal ID of T^1. Consider the part P of the tape inscription of E^1 which occupies squares which will get scanned during the computation starting with E^1. Since we consider only that part of the tape inscription which is explicitly described by E^1, P is finite.

If P contains no symbols of the form $S_{j,0}$, then the ID obtained from P by taking every second square (and possibly reversing the order depending on the state of E^1) will be an immortal ID of T.

If P contains a symbol of the form $S_{j,0}$, then let T^1 compute until the rightmost of these symbols is scanned for the first time. Then T^1 will never move to the left of the presently scanned square, and the part of the tape inscription which lies on the right of (including) this square provides us with an immortal ID of T.

RECEIVED: March 20, 1969

REFERENCES

AANDERAA, S. AND FISCHER, P. C. (1967), The solvability of the halting problem for 2-state Post machines, *J. Assoc. Comp. Mach.* **14**, 677–682.

FISCHER, P. C. (1965). On formalisms for Turing machines. *J. Assoc. Comp. Mach.* **12**, 570–580.

HERMAN, G. T. (1968), The halting problem of one-state Turing machines with *n*-dimensional tape. *Z. Math. Logik Grundl. Math.*, **14**, 185–191.

HERMAN, G. T. (1969), The unsolvability of the uniform halting problem for two-state Turing machines. *J. Symb. Logic* **34**, 161–165.

HERMAN, G. T., A simple solution of the uniform halting problem. *The J. Symb. Logic* **35**, (in press).

HOOPER, P. K. (1966), The undecidability of the Turing machine immortality problem. *The J. Symb. Logic* **31**, 219–234.

HUNTER, J. (1964), "Number Theory." Oliver and Boyd, London.

MINSKY, M. L. (1967), Computation: Finite and Infinite machines." Prentice-Hall, Englewood Cliffs, N.J.

SHANNON, C. E. (1965), A universal Turing machine with two internal states. *In* "Automata Studies," pp. 157–165. Princeton Univ. Press, Princeton, N.J.