A Model of Replication

ABRAHAM WAKSMAN

Stanford Research Institute, Menlo Park, California*

ABSTRACT. A one-dimensional array of finite-state machines is being considered as a model for sequence replication. The authors consider the initial state of the first k machines in the array as representing the sequence of k symbols to be replicated along the array. A construction scheme is developed which allows for such replication to take place. It is also shown that the speed of replication approaches synchronous speed.

KEY WORDS AND PHRASES: replication, transformation, transposition, finite-state machine, synchronous speed, Pascal triangle

CR CATEGORIES: 5.22, 5.29, 5.30, 5.39

A one-dimensional array of finite-state machines is being used as a model in demonstrating the replication of a sequence of k symbols represented by the states of the first k machines in the array before the array is turned on.

For a given sequence of length k a field is determined from which the symbols for the sequence are selected.

The sequence to be replicated undergoes a unique transformation before insertion into the array, which in turn performs the inverse transformation in the process of replicating the original sequence along the array. The construction requirements can be stated as follows.

Consider a finite (but arbitrarily long), one-dimensional array of identical, finite-state machines. The machines are synchronous, and the state of each machine at time $t + 1$ depends on its own state and that of its two neighbors at time t.

At $t = 0$ each of the first k machines is made to assume one of its internal states, thus forming a sequence of states of length k. The rest of the machines are in a single state, the quiescent state.

It is required that the states and transitions of the machines be specified in such a way that sometime in the future the terminal state of each machine will be such that a predetermined sequence of states of length k will be continually represented along the array of machines.

Solution. The solution structure is divided into two phases: ϕ_1 and ϕ_2.

During phase ϕ_1, at $t + 1$ each machine will be in a state determined by its previous state and the state of its left neighbor at the time t; specifically,

$$a_i\,|_{t+1} = a_i\,|_t + a_{i-1}\,|_t .$$

Addition is mod M where M is the size of the field from which the values of a_i were selected.

* Computer Techniques Laboratory. This research was sponsored by the Air Force Office of Scientific Research, United States Air Force, under AFOSR Grant No. 874-65 while the author was with the University of Southern California at Los Angeles, Calif.

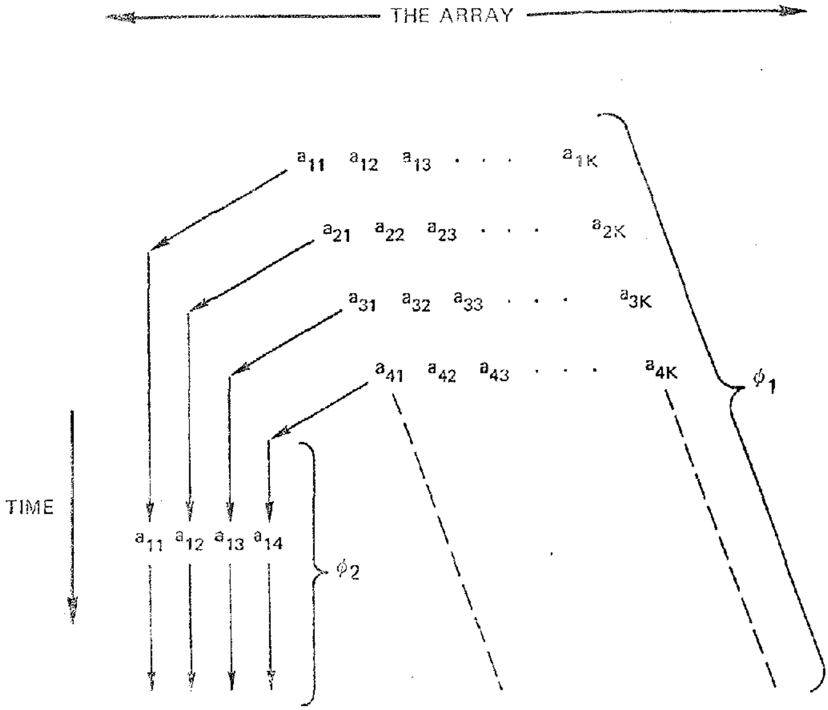

FIG. 1. The replication scheme

Thus at each interval between transitions of states, the array will display a different sequence of length k displaced one machine forward.

During ϕ_2 the first symbol in each sequence presented will be retained as the terminal state of its left neighbor. Figure 1 displays this scheme pictorially.

By reversing the rule for addition in ϕ_1, it is possible to present the propagating sequence in a matrix form:

$$\begin{matrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2k} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3k} \\ \vdots & \vdots & \vdots & & \vdots \end{matrix}$$

where

$$\begin{aligned} a_{21} &= a_{11} + a_{12} \\ a_{22} &= a_{12} + a_{13} \\ a_{23} &= a_{13} + a_{14} \\ \vdots \quad & \quad \vdots \quad\quad \vdots \end{aligned}$$

Solving for entries in the first column in terms of the entries in the first row, we obtain:

$$\begin{aligned}
a_{11} &= a_{11} \\
a_{21} &= a_{11} + a_{12} \\
a_{31} &= a_{11} + 2a_{12} + a_{13} \\
a_{41} &= a_{11} + 3a_{12} + 3a_{13} + a_{14} \\
&\vdots
\end{aligned}$$

In general the entries in the ith column of the matrix will be:

$$\begin{aligned}
a_{1i} &= a_{1i} \\
a_{2i} &= a_{1i} + a_{1(i+1)} \\
a_{3i} &= a_{1i} + 2a_{1(i+1)} + a_{1(i+2)} \\
a_{4i} &= a_{1i} + 3a_{1(i+1)} + 3a_{1(i+2)} + a_{1(i+3)} \\
&\vdots
\end{aligned}$$

Solving for the first nine entries in the ith column in terms of the entries in the first row, and putting the coefficients of the entries of the first row as entries in Table I, we can form the Pascal triangle of binomial coefficients.

Example 1. From Table 1, $a_{4i} = a_{1i} + 3a_{1(i+1)} + 3a_{1(i+2)} + a_{1(i+3)}$.

Periodicity of sequences of length k can be achieved if we restrict the entries in the first row of the matrix to a field whose size is M = g.c.d. of the entries of the $(k+1)$-th row of the Pascal triangle—ignoring first and last entries—and addition is mod (g.c.d.). This establishes a column period of length k.

This is true since for every $a_{(k+1)i}$-th entry in the matrix the only nonzero coefficient in the $(k+1)$-th row of the Pascal triangle will be 1 for a_{1i}. Thus

$$a_{(k+1)i} = a_{1i} \pmod{M},$$

and periodicity of period k of the columns of the matrix is established.

Example 2. Let $k = 5$. Then the g.c.d. of the sixth row of the Pascal triangle is

TABLE 1

	a_{1i}	$a_{1(i+1)}$	$a_{1(i+2)}$	$a_{1(i+3)}$	$a_{1(i+4)}$	$a_{1(i+5)}$	$a_{1(i+6)}$	$a_{1(i+7)}$	$a_{1(i+8)}$
a_{1i}	1								
a_{2i}	1	1							
a_{3i}	1	2	1						
a_{4i}	1	3	3	1					
a_{5i}	1	4	6	4	1				
a_{6i}	1	5	10	10	5	1			
a_{7i}	1	6	15	20	15	6	1		
a_{8i}	1	7	21	35	35	21	7	1	
a_{9i}	1	8	28	56	70	56	28	8	1

found to be 5. We now choose a sequence of length 5 with entries from a field of 5 elements (addition is mod 5).

Let the chosen sequence be 10001; then the generated matrix will be:

$$\begin{array}{ccccc} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 2 & 1 \\ 1 & 1 & 3 & 3 & 1 \\ 2 & 4 & 1 & 4 & 1 \end{array}\Bigg\} \rightarrow \text{the } 5 \times 5 \text{ matrix.}$$

$$\begin{array}{ccccc} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 2 & 1 \\ & & \vdots & & \end{array}$$

Note. It is a coincidence that the 5×5 matrix generated in Example 2 has entries from a field of 5 elements.

Let the "first transposition column" be defined as

$$a_{11}\ a_{21}\ a_{31}\ \cdots\ a_{k1}\,.$$

Let the "second transposition column" be defined as

$$b_{11}\ b_{21}\ b_{31}\ b_{41}\ \cdots\ b_{k1}$$

where $b_{11}\ b_{12}\ b_{13}\ \cdots\ b_{14}$ is a generating sequence of a new $k \times k$ matrix, and

$$\begin{aligned} b_{11} &= a_{11} \\ b_{12} &= a_{21} \\ b_{13} &= a_{31} \\ \vdots\ & \quad \vdots \\ b_{1k} &= a_{k1}\,. \end{aligned}$$

Accordingly the Jth transposition column will be the first column of the Jth matrix formed in this fashion.

This process consists of taking the first column of a $k \times k$ matrix and using it as the first row of a new matrix. The entire process is repeated J times.

Solving for the entries in the first column of the second matrix in terms of the entries in the first row of the first matrix, we find:

$$b_{11} = a_{11}$$

$$b_{21} = b_{11} + b_{12} = a_{11} + (a_{11} + a_{12})$$

$$b_{31} = b_{11} + 2b_{12} + b_{13} = a_{11} + 2(a_{11} + a_{12}) + (a_{11} + 2a_{12} + a_{13})$$

$$\begin{aligned} b_{41} = b_{11} + 3b_{12} + 3b_{13} + b_{14} &= a_{11} + 3(a_{11} + a_{12}) + 3(a_{11} + 2a_{12} + a_{13}) \\ &\quad + (a_{11} + 3a_{12} + 3a_{13} + a_{14}). \end{aligned}$$

Solving for the first four entries in the first column of the b matrix, we arrive at Table 2, which represents the coefficients for b_{11}, b_{21}, b_{31}, and b_{41}.

Example 3. $b_{31} = 4a_{11} + 4a_{12} + 4a_{13}$

Solving for the first four entries in the first column of the third matrix, we arrive at Table 3.

TABLE 2

	a_{11}	a_{12}	a_{13}	a_{14}
b_{11}	1			
b_{21}	2	1		
b_{31}	4	4	1	
b_{41}	8	12	6	1

TABLE 3

	a_{11}	a_{12}	a_{13}	a_{14}
c_{11}	1			
c_{21}	3	1		
c_{31}	9	6	1	
c_{41}	27	27	9	1

TABLE 4

	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
J_{11}	J^0				
J_{21}	J^1	J^0			
J_{31}	J^2	$2J^1$	J^0		
J_{41}	J^3	$3J^2$	$3J^1$	J^0	
J_{51}	J^4	$4J^3$	$6J^2$	$4J^1$	J^0

Example 4. $c_{31} = 9a_{11} + 6a_{12} + a_{13}$

Solving for the first five entries in the first column of the Jth matrix, we arrive at Table 4. Observe that all entries in the table of coefficients of the Jth column are a product of entries in the Pascal triangle and some power of J. Thus for $J = 0$ (mod M), all entries in Table 4 except J^0 will be zero and

$$J_{11} = a_{11}$$

$$J_{21} = a_{12}$$

$$J_{31} = a_{13}$$
$$J_{41} = a_{14}$$
$$\vdots \qquad \vdots$$
$$J_{k1} = a_{1k}\,.$$

Thus a new recursion relationship is established where the original generating sequence, the first row of the a-matrix, will appear as the first column of the Jth matrix (addition is mod M), as displayed in Example 5.

Example 5.

1st matrix (a)	*2nd matrix* (b)	*3rd matrix* (c)	*4th matrix* (d)	*5th matrix* (J)
1 0 0 0 1	1 1 1 1 2	1 2 4 3 2	1 3 4 2 2	1 4 1 4 2
1 0 0 1 1	2 2 2 3 2	3 1 2 0 2	4 2 1 4 2	0 0 0 1 2
1 0 1 2 1	4 4 0 0 2	4 3 2 2 2	1 3 0 1 2	0 0 1 3 2
1 1 3 3 1	3 4 0 2 2	2 0 4 4 2	4 3 1 3 2	0 1 4 0 2
2 4 1 4 1	2 4 2 4 2	2 4 3 1 2	2 4 4 0 2	1 0 4 2 2

$$\begin{bmatrix} \text{first row} \\ \text{first matrix} \end{bmatrix} = \begin{bmatrix} \text{first column} \\ \text{fifth matrix} \end{bmatrix}$$

The Design of the Machine

(1) Let it be desired to replicate a sequence of length k.

(2) A field size for the sequence symbols is determined by taking the g.c.d. of the entries of the $(k + 1)$-th row in the Pascal triangle.

(3) The sequence to be replicated is now constructed with symbols chosen from the determined field.

(4) The constructed sequence is now transformed into a sequence identical to the Jth transposition column.

(5) The transformed sequence is now loaded into the machine array starting from the second machine.

The Machine Code Book. The entries in the tables represent the state of the machine at $t + 1$ for the listed state and the associated left and right neighbors at time t.

The State List

Q = the quiescent state
P = the transition state

$$\phi_1 \begin{cases} a_1 \\ \vdots \\ a_M \end{cases}$$

$$\phi_2 \begin{cases} a_1 \\ \vdots \\ a_M \end{cases}$$

$\underline{Q}$

L \ R	Q	a_i	a_j
Φ	Q	a_i	a_j
a_i	a_i		
a_j	a_j		
Q	Q		

$\underline{P}$

L \ R	a_i	a_j
X	a_i	a_j

$\underline{a_i}$

L \ R	Q	a_i	a_j	P	
Q	--	P	P	--	
a_i	$2a_i$	$2a_i$	$2a_i$	--	ϕ_1
	a_i	a_i	a_i	a_i	ϕ_2
a_j	$a_j + a_i$	$a_j + a_i$	$a_j + a_i$	--	ϕ_1
	a_j	a_j	a_j	a_j	ϕ_2
P	--	P	P	--	
Φ		a_i	a_i	a_i	

Note. Φ is an external state representing: *no* left or right neighbor. Once the machine array is turned on it will remain in ϕ_1 until the transition to state P; from then on it will operate in ϕ_2, as illustrated.

$$Q \rightarrow \phi_1 \rightarrow P \rightarrow \phi_2$$

Operation Mode Graph

Example 6. Let it be desired to replicate the sequence 10001. It was found in Example 5 that the fourth transposition column is equal to 14142, which is the one loaded into the array. Figure 2 illustrates the sequence of events that take place after the array has been turned on.

THE ARRAY

TIME

0		1	4	1	4	2															
1	1	P	0	0	0	1	2														
2	1	0	P	0	0	1	3	2													
3	1	0	0	P	0	1	4	0	2												
4	1	0	0	0	P	1	0	4	2	2											
5	1	0	0	0	1	P	1	4	1	4	2										
	1	0	0	0	1	1	P	0	0	0	1	2									
	1	0	0	0	1	1	0	P	0	0	1	3	2								
	1	0	0	0	1	1	0	0	P	0	1	4	0	2							
	1	0	0	0	1	1	0	0	0	P	1	0	4	2	2						
	1	0	0	0	1	1	0	0	0	1	P	1	4	1	4	2					
	1	0	0	0	1	1	0	0	0	1	1	P	0	0	0	1	2				
	1	0	0	0	1	1	0	0	0	1	1	0	P	0	0	1	3	2			
	1	0	0	0	1	1	0	0	0	1	1	0	0	P	0	1	4	0	2		
	1	0	0	0	1	1	0	0	0	1	1	0	0	0	P	1	0	4	2	2	
	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	P	1	4	1	4	2
	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	P	0	0	0	1
	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	P	0	0	1
	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	P	0	1
	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	P	1

Fig. 2

Time Consideration. If we define a *signal* on the array as being a state of *a* machine which propagates from machine to machine in one direction as time goes on, then the fastest a state can propagate, i.e. the fastest signal on the array, is a signal that propagates at synchronous speed, which is the speed at which state transitions on the array take place. It is seen from the state-transistion table, as well as from Figure 2, that state P is a state that propagates at synchronous speed. All the machines in the array opposite to the direction of propagation of the state P are static and display a terminal state which is part of the replicating sequence.

Hence we can state that the sequence is being replicated at synchronous speed, which is as fast as possible.

Let us consider now the case of an array with slightly more complicated machines which is also capable of performing the initial transformation on the sequence to be replicated.

All that is necessary in this new array is to load the sequence to be replicated, let the array go through a transformation cycle, and then start the replication as described.

Clearly the transformation is a finite process and is dependent on the number of transpositions J required for each transformation.

Thus even for this array, for any J, the speed of replication approaches synchronous speed for sufficiently long arrays.

The Binary Sequences. From the entries of the Pascal triangle it can be seen that all the entries in the 2^nth rows (where n is an integer) have a g.c.d. of 2.

Thus following the previously established rules, all sequences of length $k = 2^n$ will be binary sequences.

Further, it is seen that when these binary sequences assume the form

$$\underbrace{0\ \overbrace{c_1\ c_1\ c_1\ c_1\ c_1}^{2^{n-1}}\ \overbrace{c_2\ c_2\ c_2}^{2^{n-2}} \cdot\cdot\cdot\cdot\cdot\ \overbrace{c^n}^{2^{n-n}}}_{2^n}$$

THE ARRAY

TIME

	0	1	1	1	1	0	0	1											
0	P	1	0	0	0	1	0	1	1										
0	1	P	1	0	0	1	1	1	0	1									
0	1	1	P	1	0	1	0	0	1	1	1								
0	1	1	1	P	1	1	1	0	1	0	0	1							
0	1	1	1	1	P	0	0	1	1	1	0	1	1						
0	1	1	1	1	0	P	0	1	0	0	0	1	0	1					
0	1	1	1	1	0	0	P	1	1	0	1	0	1	1	1				
0	1	1	1	1	0	0	1	P	0	1	1	1	1	0	0	1			
0	1	1	1	1	0	0	1	0	P	1	1	0	0	1	0	1	1		
0	1	1	1	1	0	0	1	0	1	P	0	0	0	1	1	1	0	1	
0	1	1	1	1	0	0	1	0	1	1	1	1	0	1	0	0	1	1	1
0	1	1	1	1	0	0	1	0	1	1	P	P	1	1	1	0	1	0	0
0	1	1	1	1	0	0	1	0	1	1	1	1	P	0	0	1	1	1	0
0	1	1	1	1	0	0	1	0	1	1	1	1	0	P	0	1	0	0	1
0	1	1	1	1	0	0	1	0	1	1	1	1	0	0	P	1	1	0	1
0	1	1	1	1	0	0	1	0	1	1	1	1	0	0	1	P	0	1	1
0	1	1	1	1	0	0	1	0	1	1	1	1	0	0	1	0	P	1	0
0	1	1	1	1	0	0	1	0	1	1	1	1	0	0	1	0	1	P	1

FIG. 3

where $c_i = 0$ or 1, they will generate $k \times k$ matrices where

$$\begin{aligned} a_{11} &= a_{11} \\ a_{21} &= a_{12} \\ a_{31} &= a_{13} \\ \vdots \quad & \quad \vdots \\ a_{k1} &= a_{1k} . \end{aligned}$$

Thus for this case of binary sequences there is no need for transformation before loading the sequence into the array.

Example 7. Let it be desired to replicate the sequence 01111001. Loading the sequence directly into the machine array, the sequence of events will be as in Figure 3.

APPENDIX

By the rule adopted for computing the values of the entries in the $n \times n$ matrix:

$$a_{(k+1)i} = \sum_{l=0}^{k} \binom{k}{l} a_{1(l+i)} \tag{1}$$

where $0 \leq k \leq n$ and summation is mod M, and where

$$M = \text{g.c.d.} \binom{n}{1}, \binom{n}{2}, \cdots, \binom{n}{n-1} .$$

Thus

$$a_{(n+1)i} = \sum_{k=0}^{n} \binom{n}{k} a_{1(k+i)} = a_{1i} . \tag{2}$$

THEOREM. *If the entries of the Jth transposition matrix are computed so that* $J_{1(k+1)} = (J-1)_{(k+1)1}$, then

$$J_{(k+1)1} = \sum_{l=0}^{k} J^{(k-l)} \binom{k}{l} a_{1(l+1)}. \tag{3}$$

PROOF. For the b-matrix, which is the second transposition matrix,

$$b_{1(k+1)} = a_{(k+1)1} \qquad \text{(by definition)},$$

but

$$a_{(k+1)1} = \sum_{l=0}^{k} \binom{k}{l} a_{1(l+1)} \qquad \text{(from eq. (1))},$$

so that

$$b_{(k+1)1} = \sum_{l=0}^{k} \binom{k}{l} b_{1(l+1)} = \sum_{l=0}^{k} \binom{k}{l} \sum_{m=0}^{l} \binom{l}{m} a_{1(m+1)}, \tag{4}$$

and we can rewrite eq. (4) in terms of $a_{1(m+1)}$ as follows:

$$b_{(k+1)1} = a_{1(0+1)} \sum_{l=0}^{k} \binom{k}{l}\binom{l}{0} \cdots a_{1(m+1)} \sum_{l=m}^{k} \binom{k}{l}\binom{l}{m} \cdots .$$

Now solving for the general term,

$$\begin{aligned}\sum_{l=m}^{k} \binom{k}{l}\binom{l}{m} &= \frac{k!}{m!} \sum_{u=0}^{k-m} \frac{1}{(k-m-u)!u!} \\ &= \frac{k!}{(k-m)!m!} \sum_{u=0}^{k-m} \frac{(k-m)!}{(k-m-u)!u!} = \binom{k}{m} \sum_{u=0}^{k-m} \binom{k-m}{u} \\ &= \binom{k}{m} 2^{(k-m)}.\end{aligned}$$

Summing over all m, $0 \le m \le k$,

$$b_{(k+1)1} = \sum_{m=0}^{k} 2^{(k-m)} \binom{k}{m} a_{1(m+1)}. \tag{5}$$

The inductive step follows. Assume that

$$J_{(k+1)1} = \sum_{l=0}^{k} J^{(k-l)} \binom{k}{l} a_{1(l+1)}$$

is true. Then by definition,

$$(J+1)_{(k+1)1} = \sum_{l=0}^{k} \binom{k}{l} (J+1)_{1(l+1)} = \sum_{l=0}^{k} \binom{k}{l} J_{(l+1)1}$$

so that

$$(J+1)_{(k+1)1} = \sum_{l=0}^{k} \binom{k}{l} \sum_{m=0}^{l} J^{(l-m)} \binom{l}{m} a_{1(m+1)}.$$

Rewriting in terms of $a_{1(m+1)}$,

$$(J+1)_{(k+1)1} = a_{1(0+1)} \sum_{l=0}^{k} \binom{k}{l}\binom{l}{0} J^{l} + \cdots + a_{1(m+1)} \sum_{l=m}^{k} \binom{k}{l}\binom{l}{m} J^{(l-m)}.$$

Solving for the general term,

$$\sum_{l=m}^{k} \binom{k}{l}\binom{l}{m} J^{(l-m)} = \frac{k!}{m!} \sum_{l=m}^{k} \frac{J^{(l-m)}}{(k-l)!(l-m)!}$$
$$= \frac{k!}{m!(k-m)!} \sum_{u=0}^{k-m} \frac{J^{u}(k-m)!}{u!(k-m-u)!} = \binom{k}{m} \sum_{u=0}^{k-m} \binom{k-m}{u} J^{u}$$
$$= \binom{k}{m} (J+1)^{k-m}.$$

Summing over all m, $0 \le m \le k$,

$$(J+1)_{(k+1)1} = \sum_{m=0}^{k} \binom{k}{m} (J+1)^{(k-m)} a_{1(m+1)}.$$

This establishes eq. (3) for all J.

A new recurrence relationship can be established where the $(J+1)$-th matrix is identical to the a-matrix if we take $J = 0 \pmod{M}$, so that $J_{(k+1)1} = a_{1(k+1)}$.

ACKNOWLEDGMENT. The author would like to express his gratitude to Professors S. W. Golomb and I. S. Reed for their encouragement and support on this work, among others. Thanks are also due the referee for pointing out the need for the section on time consideration, which was missing from the original manuscript.

REFERENCE

1. Moore, E. F. Machine models of self-reproduction. Proc. Symp. in Applied Math. Vol. 14, 1962, Amer. Math. Soc., Providence, R.I.

RECEIVED MARCH, 1967; REVISED JUNE, 1968