EXISTENCE OF FINITE AUTOMATA CAPABLE OF BEING SYNCHRONIZED ALTHOUGH ARBITRARILY CONNECTED AND NUMEROUS*†

Pierre Rosenstiehl
Ecole Pratique des Hautes Etudes
Paris

Summary - A collection of n finite, identical automata are considered, where each one, at each unit time step, takes a new state as a function of the state taken at the preceding step by itself and by certain other automata in the collection, called its neighbors, arbitrarily chosen, but limited in number. The neighborhood relation is assumed symmetric. One is asked to determine for a given d, and independently of n, the number of states and the state-transition function of an automaton of the type which has the following property: All automata in the collection are put in the resting state except for one of them, $\hat{A}$, arbitrarily chosen, put into an initial state distinct from the resting state; then at the end of a finite length of time all automata in the collection depending, by the neighborhood relation, directly or indirectly on $\hat{A}$, are put simultaneously and for the first time into a final state agreed upon in advance. It is said then that our arbitrarily numerous automata are connected into an arbitrary network of automata of degree d, and that those in the component connected to $\hat{A}$ have been synchronized.

Our problem is a generalization of the "Firing Squad Synchronization Problem" of John Myhill, which envisions n soldiers disposed along a line - that is to say, with a neighbor on the right and a neighbor on the left - and gives the role of $\hat{A}$ to a soldier at the end of the line, the general. For this problem R. M. Balzer has published an 8-state solution with a minimum "time-to-fire", 2n-2. We show the general problem also has a solution by proposing an $(8^d x 5^d)$-state automaton with a time-to-fire of 4n-6.

Our networks of automata constitute a generalization of the "Iterative Arrays of Finite-State Machines" of von Neumann and Moore.

---

*Translated by Alvy Ray Smith III, Department of Electrical Engineering, New York University, Bronx, New York 10453, November 1971 (unauthorized).
†Published in ICC Bulletin, 1966, Vol. 5, pp. 245-261 (in French).

I. STATEMENT OF THE PROBLEM

By finite automaton A, it is understood here that given is a finite set E called the set of states, a finite set V called the set of inputs, and a mapping

$$S: E \times V \to E$$

It is said that A takes, for each time $t \in N$, its state $e(t)$ from the set E, and its input $v(t)$ from the set V. At time $t+1$, its state is determined by the equation:

$$e(t+1) = S(e(t),v(t)) \quad t \in N.$$

We note that it is not possible for an automaton in the class so defined to count arbitrarily high.

More precisely, it is understood here by <u>automaton</u> (E,d,S) <u>of degree</u> d, a finite automaton A which takes its input from the set E^d. It is therefore defined by a finite set E, an integer d, and a mapping S:

$$S: E \times E^d \to E.$$

The input to an automaton (E,d,S) of degree d is a d-tuple, each element of which is a member of E. The set of indices of this d-tuple will be denoted by Δ:

$$\Delta = \{1,2,\ldots,d\};$$

thus it will be called the set of "input addresses" for the automaton (E,d,S) of degree d.

It is understood here by a <u>graph</u> G <u>of degree at most</u> d that given are an integer d and a quadruple (X,U,I,T) defined in the following way:

Two finite sets X and U called respectively the set of nodes and the set of edges of G, and two mappings

$$I: U \to X \text{ and } T: U \to X$$

called respectively "initial end" and "terminal end", satisfy these relations:

$$I(u) \neq T(u) \quad \forall u \in U$$

$$\text{and cardinality of } \omega(x) \leq d \qquad \forall x \in X$$

where $\omega(x)$ designates the set of edges which have x as an end (initial or terminal).

We now define what is meant by <u>network</u> R <u>of</u> <u>automata</u> (E,d,S).

Assume a graph G of degree at most d: $G = (X,U,I,T)$.

Assume an automaton of degree d: $A = (E,d,S)$.

At each node i of G one "attaches" an automaton of type A, denoted A_i, in the following way: One maps injectively the set $\omega(i)$ of edges of G with end i, into the set of input addresses of A_i:

$$\forall i \in X, \text{ there is an injection: } \phi_i : \omega(i) \rightarrow \Delta$$

and requires that, upon calling e_i the state of A_i and v_i its d-tuple of inputs:

$$k\text{-th element of } v_i = e_i \text{ if } k = \phi_i(u)$$

and if j is the end of edge u other than i.

$$k\text{-th element of } v_i = 0 \text{ if } k \quad \mathrm{Im}\ \phi_i$$

where 0 designates one of the elements of set E.

It follows then that at time t+1 automaton A_i situated on node i takes a state as a function of the state at time t of itself and of its neighbors[1] in the graph.

Since G is an arbitrary graph of degree at most d, everything happens exactly as if our automata had d terminals, each subject to being connected to a terminal of <u>another</u> automaton, and as if they had randomly connected their terminals to read from their neighbors their respective states, each having, voluntarily, let certain terminals go unconnected and what we will call free.

The networks of automata constitute a generalization of the "Iterative Arrays of Finite-State Machines" of von Neumann and Moore (See Cole [3]).

---

[1] The orientation of the edges of G might seem superfluous: It is effectively arbitrary, and will serve only for the writing of graph words in paragraph III.

The problem we pose is the following:

A collection of arbitrary size n of automata (E,d,S) forms an arbitrary network of automata R of degree d. Other than the neutral state 0, three elements of E are specified: state D (dormant), state M (march!), state F (fire).

It is required to determine E and S in such a way that, if A is an arbitrarily chosen automaton in the collection and if:

at time 1: $e_i(1) = M$ if $A_i = A$, and $e_i(1) = D$ if $A_i \neq A$
then there exists a finite integer θ such that:

at time θ: $e_i(\theta) = F$ for each automaton A_i of just that component connected to A in R,

and such that:

for all times $t < \theta$: $e_i(t) \neq F$, for $i = 1,2,\ldots,n$.

We say then that in time θ our automata have been synchronized.

Clearly it is a matter of defining E and S as a function of d only, that is to say, independently of n and also independently of the graph G associated with R.

Our result is that there exists an automaton (E,d,S) - that is, a finite automaton - which solves this problem.

This problem has been posed and solved for the case where d = 2 under the name of the "Firing Squad Synchronization Problem".

Since the automaton (E,d,S), which solves the general problem, has among its components the automata (E,2,S) which solve the particular problem (d = 2), we devote our paragraph II to the "Firing Squad Synchronization Problem", essentially to facilitate the reading of the sequel.

II. ON THE "FIRING SQUAD SYNCHRONIZATION PROBLEM"

The problem entitled the "Firing Squad Synchronization Problem" defined in 1957 by J. Myhill, first solved by J. McCarthy and M. Minsky and E. Goto [4], is treated in the literature by E. F. Moore [5], A. Waksman [10], and R. M. Balzer [1].

Statement. Assume a line of n similar automata A_i of degree 2, subscripts from 1 to n, are connected in line in the order of the indices[1]. A_i is defined by:

a finite set $E = \{0, \hat{M}, D, F, \ldots\}$ from which it takes its state $e_i(t)$ at time t,

a function $S: E \times E^2 \to E$

the relation for state transition:

$$e_i(t+1) = S(e_i(t), e_{i-1}(t), e_{i+1}(t)) \text{ for } t > 1 \text{ and } i = 1,2,\ldots,n$$

for which one assumes for convenience: $e_0(t) = e_{n+1}(t) = 0$ for $t \in N$.

It is required to define E and S in such a way that, if

at time 1: $e_1(1) = \hat{M}$, $e_2(1) = e_3(1) = \ldots = e_n(1) = D$

there exists a finite time θ so that

at time θ: $e_1(\theta) = e_2(\theta) = \ldots = e_n(\theta) = F$, and so that

for all times $t < \theta$: $e_i(t) \neq F$, for $i = 1,2,\ldots,n$.

---

[1] In the literature on the subject one speaks of n soldiers in line, of which one, situated at the end of the line, is the general; he gives the signal to fire, messages are exchanged from neighbor to neighbor and suddenly, all together, for the first time the soldiers fire. Our generalization could be called the "Scattered Firing Squad Synchronization Problem".

A Simple Solution

We propose a simple solution of this problem. The set of states of the automaton solution will be denoted:

$$E = \{ D, \hat{M}, \bar{M}, M, \vec{R}, \overleftarrow{R}, \vec{1}, \vec{2}, \vec{3}, \overleftarrow{1}, \overleftarrow{2}, \overleftarrow{3}, F, 0 \}$$

say, 14 symbols to which we have given a mnemonic form:

D = dormant state

$\hat{M}$ = march! state

$\bar{M}$ = active boundary state

M = dead boundary state

$\vec{R}$ = fast signal sent from left to right

$\vec{1}$ = slow signal of age 1 starting from the left

$\overleftarrow{3}$ = slow signal of age 3 starting from the right

F = final state (Fire)

0 = state of the free input element of A_1 and of A_n

We define the function S. The essential idea is the following: If the leftmost automaton in the line, ($\hat{M}$), emits on one hand a fast signal ($\vec{R}$) which at each time step progresses through an automaton and rebounds from the end of the line, and on the other hand at the following time step, becomes ($\bar{M}$), emits a slow signal ($\vec{1}, \vec{2}, \vec{3}$) which progresses three times less rapidly, these two signals will cross in the middle of the line and will determine one or two middle automata which will be put into the march! state ($\hat{M}$); these in turn become the head automata of two equal lines of automata, one half as long as the first considered. After a certain number of successive divisions, each automaton exists in state M, $\bar{M}$, or $\hat{M}$, and it is the first time that at least three successive letters M appear: The final state F is then adopted by each at the same time.

Represented in Figure 1 are the histories of the "Firing Squads" of n soldiers for $n = 1,2,\ldots,8$.

The function S is entirely described if we give for each of the thirteen states e, distinct from 0, the list of the x, which solve the equation:

$$S(x) = e, \quad \text{with } e \in E \text{ and } e \neq 0.$$

Each solution x will be denoted by a word of three letters

$$e_{i-1} e_i e_{i+1}$$

indicating the state of three successive automata $A_{i-1} A_i A_{i+1}$ at time t; e is then the state of automaton A_i at time t+1; it is for an obvious simplification of expression that we have inverted the first two elements of the triple x.

Hierarchical table of the solutions of the equation $S(x) = e$.

Directions for use

(1) (.) designates as not important that element of E

(2) $\dot{e}$ designates each state which is written with the letter e not overlined or where the overlining is not important

(3) x is a triple solution of $S(x) = e_r$, if one does not have $S(x) = e_p$ for $p < r$.

r	e_r	x						
1	F	$\dot{M}\,\dot{M}\,\dot{M}$	$O\,\dot{M}\,\dot{M}$	$\dot{M}\,\dot{M}\,O$				
2	$\hat{M}$	$\vec{3}\,.\,\overleftarrow{R}$	$\vec{R}\,.\,\overleftarrow{3}$	$\vec{2}\,\overleftarrow{R}\,.$	$\vec{R}\,\overleftarrow{2}\,.$	$.\,\vec{2}\,\overleftarrow{R}$	$.\,\vec{R}\,\overleftarrow{2}$	$\dot{M}\,.\,\dot{M}$
3	$\bar{M}$	$.\,\hat{M}\,.$						
4	M	$.\,D\,O$	$O\,D\,.$	$.\,\bar{M}\,.$	$.\,M\,.$			
5	$\vec{R}$	$\vec{R}\,.\,.$	$M\,\overleftarrow{R}\,.$	$\hat{M}\,.\,.$				
6	$\overleftarrow{R}$	$.\,.\,\overleftarrow{R}$	$.\,\vec{R}\,M$	$.\,.\,\hat{M}$				
7	$\vec{1}$	$\bar{M}\,.\,.$	$\vec{3}\,.\,.$					
8	$\overleftarrow{1}$	$.\,.\,\bar{M}$	$.\,.\,\overleftarrow{3}$					
9	$\vec{2}$	$.\,\vec{1}\,.$						
10	$\overleftarrow{2}$	$.\,\overleftarrow{1}\,.$						
11	$\vec{3}$	$.\,\vec{2}\,.$						
12	$\overleftarrow{3}$	$.\,\overleftarrow{2}\,.$						
13	D	$.\,.\,.$						

Known results on the "Firing Squad Synchronization Problem"

The solution of the "Firing Squad Synchronization Problem" which we proceed to sketch is an automaton of 13 states (there is no reason for counting the symbol 0). As for the time-to-fire, $\theta(n)$, the time at the end of which the n automata of the line prove to be synchronized, that is to say, in state F, it is for this solution less than 3n, and more precisely such that $\theta(n)/3n$ tends toward 1 as n increases.

E. F. Moore showed in first publishing this problem that, whatever solution (E,2,S) is given, 2n-2 constitutes a lower limit for $\theta(n)$; E. Goto has effectively exhibited a solution (E,2,S) for which the size of E is large but the time-to-fire is equal to 2n-2. A. Waksman [10] has produced recently a solution of 16 states for which the time-to-fire is equal to the minimum time 2n-2. Finally, even more recently R. M. Balzer [1] has produced a solution of 8 states for which the time-to-fire is also minimum.

In the sequel we will denote by q the minimum number of states - a number actually unknown - of the solutions (E,2,S) for the "Firing Squad Synchronization Problem" for which the time-to-fire is 2n-2.

III. THE TREEWORDS OF A CONNECTED GRAPH

As we have stated, we are going to reduce in a certain way the general problem of a network of automata $\tilde{R}$ (d arbitrary) to the special case of a line of automata (d = 2) studied in paragraph II.

One tempting way at first sight would be to class the automata of the component connected to $\hat{A}$ in $\tilde{R}$ in successive levels $0,1,2,\ldots,\ell,\ldots$ according to their shortest distance ℓ (in number of edges) to $\hat{A}$. $\hat{A}$ forms by itself level 0. This numbering can be made by the automata themselves, level by level commencing with $\hat{A}$; and as Moore [6] has remarked, one can restrict himself to a numbering modulo 3, so that thanks to only 3 states each automaton knows of its neighbors, those belonging to the same level, to the preceding level and to the succeeding level. Can one then treat the ordered levels of automata like a line of automata? In fact not, because in general there will exist automata without neighbors in the succeeding level and which, without knowing it, do not exist in the highest numbered level.

This first envisioned way, although particularly elegant, leads to a solution only for special networks, such as the hypercube and the regular polyhedron, where each node has an opposite pole. For some such networks, which will not be characterized here, an obvious solution with only q+3 states realizes the synchronization in a time of only 2d-2, if d is the diameter of the graph. We remark that d is less than the number of automata n.

For the most general case of a network $\tilde{R}$ of degree d, we have resorted to another method of alignment: the treewords of a connected graph.

Definition of treewords

Assume a graph G, that is to say two sets and two functions:

$$X = \{A_1, A_2, \ldots, A_n\} \qquad I: U \to X \qquad \text{with } I(u) \neq T(u)$$
$$U = \{u_1, u_2, \ldots, u_m\} \qquad T: U \to X$$

These will be called:

the quadruplet (X,U,I,T)	:	graph G
the elements of X	:	the nodes of G
the elements of U	:	the edges of G
the image I(u)	:	the initial end of u
the image T(u)	:	the terminal end of u

The functions I and T define for each edge of G, on the one hand two end nodes, and on the other hand an orientation (chosen arbitrarily in what follows). One calls a word on G of p letters, each word

$$\sigma = \ell_1\ell_2\ldots\ell_p$$

written in the alphabet $\tilde{A} = \{u_1,\bar{u}_1,u_2,\bar{u}_2,\ldots,u_m,\bar{u}_m\}$ and which is such that

$$\begin{array}{llll} & d(\ell_i) & = g(\ell_{i+1}) & \text{for } i = 1,2,\ldots,p-1 \\ \text{where} & d(\ell) & = T(u) & \text{if } \ell = u \\ & & \quad I(u) & \text{if } \ell = \bar{u} \\ \text{and} & g(\ell) & = I(u) & \text{if } \ell = u \\ & & \quad T(u) & \text{if } \ell = \bar{u}. \end{array}$$

d(ℓ) and g(ℓ) are read respectively "right end of ℓ" and "left end of ℓ".†

The set of words on graph G is denoted by L.

The functions d and g from $\tilde{A}$ into X are extended to the non-empty set of words on G by taking:

$$g(\sigma) = g(\ell_1) \quad \text{and} \quad d(\sigma) = d(\ell_p).$$

G is said to be connected if and only if:

$$\forall x,y\in X, \text{ with } x \neq y, \ \exists\sigma\in L: x = g(\sigma) \text{ and } d(\sigma) = y.$$

---

†[From droite and gauche, respectively. A.R.S.]

The symbol $\bar{\ell}$ will denote the letter $\bar{u}$ if $\ell = u$, and the letter u if $\ell = \bar{u}$; and it will be said that ℓ and $\bar{\ell}$ correspond to the same edge u and are conjugates.

In our graph G, it is assumed that at each node x, the edges u with end x, that is to say for which $I(u) = x$ or $T(u) = x$, are totally ordered. This local order permits us to speak of the "first edge" of x if it has at least one, and in that case of the "first letter" ℓ for which $g(\ell) = x$.

We have associated with each letter ℓ_1 in the alphabet A a word $\sigma(\ell_1)$, which will be called <u>the treeword of</u> ℓ_1. The treeword of ℓ_1 denoted:

$$\sigma(\ell) = \ell_1 \ell_2 \ldots \ell_{i-1} \ell_i \ldots \ell_h$$

is defined by the following program:

(a) ℓ_1 is its first letter.

(b) for $i = 2,3,\ldots,h$
ℓ_1 is the "first letter" such that $g(\ell_i) = d(\ell_{i-1})$
and $d(\ell_i) \neq g(\ell_j)$ for $j<i$.

(c) if (b) is not valid $\ell_i = \bar{\ell}$, where ℓ designates the rightmost letter for which the conjugate letter $\bar{\ell}$ is not yet written.

(d) if (c) is not valid the writing of $\sigma(\ell_1)$ is terminated.

(e) $\sigma(\ell_1)$ has the maximum number of letters allowed by (b) and (c).

The definition itself of the treeword $\sigma(\ell_1)$ affirms its existence, its finiteness, its uniqueness and the evenness of h.

The essential property of the word $\sigma(\ell_1)$ is that each of its letters ℓ_i can be "written" by the automaton placed at the node $x = g(\ell_i)$ in the graph G, which has retained in memory only the letters of $\sigma(\ell_1)$ having x for right or left end. This automaton has no need of knowing all the parts of $\sigma(\ell_1)$ already written; in order to apply (c) for example, it does not read the word already written backwards, but keeps in its memory only the last letter ℓ written having end x, and for which the conjugate $\bar{\ell}$ has not been written.

It can be shown that the treeword $\sigma(\ell_1)$ has the following additional properties:

(1) $h = 2n-2$ if G is connected and has n nodes.

(2) $g(\sigma(\ell_1)) = d(\sigma(\ell_1)) = g(\ell_1)$.

(3) if ℓ is written in $\sigma(\ell_1)$, ℓ and $\bar{\ell}$ are written there one time exactly.

(4) $\forall x \varepsilon X$, $\exists \ell_i$ in $\sigma(\ell_1)$: $d(\ell_i) = x$, if G is connected.

We reference G. Tarry [8][9], Tremaux, and P. R. [7].

If G is not connected properties 1 and 4 are clearly valid for the component connected to ℓ_1, that is to say for the maximal connected subgraph of G for which ℓ_1 is one letter.

Property 4 for words $\sigma(\ell_1)$ assures us that that unique word, written starting from ℓ_1, activates <u>all the nodes</u> of the component connected to ℓ_1; in addition $\sigma(\ell_1)$ has the virtue of being a word of minimum length with this property.

In the case of the graph with 4 nodes in Figure (2), and for $\ell_1 = \bar{a}$, we have for $\sigma(\ell_1)$ the following word of 6 letters:

$$\sigma(\bar{a}) = \bar{a}c\bar{c}\bar{b}ba$$

In summary, we proceed to define for each graph G an injection

$$\sigma: A \rightarrow L$$

which will be called the "treeword function" of G, associated with a local order of edges at each node of G.

In a connected network of automata, the word $\sigma(\ell_1)$ will correspond to a "Firing Squad" in line, where a given automaton A_i might appear several times; each automaton will appear there in any case at least once.

<u>Realization of treewords by a network of automata.</u>

We now specify the operations executed by an automaton $A_i = (E, d, \vec{M})$ which "attached" to node i of graph G, calls off the letters ℓ of $\sigma(\ell_1)$ with left end i.

A treeword, if we recall its characteristic properties, is defined completely by its local properties. The word $\sigma(\ell_1)$ can therefore be realized in pieces by the automata A_i. The local ordering of the edges at i is by convention that which leads, thanks to the injection ϕ_i, to the ordering Δ, the set of input addresses of A_i.

By the <u>right</u> (resp. left) <u>edge</u> of a letter ℓ of G is meant the input address associated with the right (resp. left) end of ℓ.

We show that the contribution on the part of automaton A_i to the word $\sigma(\ell_1)$ amounts, for each time t, to a function

$$m_i(t) : \Delta \to \{0, X, *, +, .\}$$

that is to say a word of d letters written with the following symbols:

0: free input address for A_i (not the end of an edge)

X: input address of A_i, the right edge of the first letter ℓ with right end i written in $\sigma(\ell_1)$ at time t, but only in the case where $\bar{\ell}$ is not yet written in the part of $\sigma(\ell_1)$ written by time t

*: input address of A_i, formerly marked X, the right edge and left edge of two conjugate letters which appear in the part of $\sigma(\ell_1)$ already written by time t

+: input of A_i, which is not in the category X and which is the left edge of a letter which appears in the part of $\sigma(\ell_1)$ already written by time t

.: input of A_i, the left edge of a letter which does not appear in the part of $\sigma(\ell_1)$ already written by time t

The word $m_i(t)$ is sufficiently specified to define all sequences of two letters $\ell_p \ell_{p+1}$ such that $i = d(\ell_p) = g(\ell_{p+1})$, for the following reasons:

ℓ_{p+1} is deduced from ℓ_p by instructions (b) and (c) of the program for treewords
after ℓ_{p+1} the next letter of $\sigma(\ell_1)$ of right end i, if there is one, is necessarily $\bar{\ell}_{p+1}$.

For example, if the 9 input addresses of A_i (case d = 9) are at time t in the state:

```
0 0 + + X . 0 0 .
1 2 3 4 5 6 7 8 9
```

and if one denotes by 3,4,5,6,9 the letters of G which have for right edge the input addresses of A_i of the same name, we affirm, in the part of $\sigma(\ell_1)$ written by time t, the existence of the configuration:

$$5\bar{3}\ldots 3\bar{4}\ldots 4\bar{6}\ldots 6\bar{9}\ldots 9\bar{5}$$

and that there does not appear another sequence $\ell_p \ell_{p+1}$ such that $i = d(\ell_p) = g(\ell_{p+1})$. The word $m_i(t)$ is by definition a word of d letters written in the alphabet {0,X,*,+,.}, subject to the following rules:

X and * appear at most one time and not together.
+ cannot precede ..

This word is the state of A_i = (E,d,M), the automaton charged with the realization of $\sigma(\ell_1)$.

Let k(d) denote the number of words of this variety, that is, the size of E. Clearly $k(d)<5^d$.

The next-state function M indicates:

either the absence of transition
or the replacement of the first letter . by the letter +
or the replacement of the letter X by the letter *.

The replacements correspond to the writing of ℓ_{p+1} in $\sigma(\ell_1)$, then that of ℓ_p is signaled by a neighbor of A_i in G. Thus the automaton A_i, situated higher, passes successively through the following six distinct states:

0 0 . . . . 0 0 .	(state of rest)
0 0 + . X . 0 0 .	
0 0 + + X . 0 0 .	
0 0 + + X + 0 0 .	
0 0 + + X + 0 0 +	
0 0 + + * + 0 0 +	(final state)
1 2 3 4 5 6 7 8 9	

An automaton A chosen arbitrarily calls off, for the name of ℓ_1, the first letter of the word $\sigma(\ell_1)$, its first input address which is not free. A is the only one to never use either X or *. When A replaces the last letter of its word-state by the letter +, it calls off the last letter of $\sigma(\ell_1)$.

In conclusion 5^d states suffice for the automata (E,d,M) to be capable, we recall, when they are connected in networks of degree d, of realizing a treeword of the network upon starting with an arbitrary one among them. This operation requires, for a connected network of n automata, a time 2n-2, since the treeword has then 2n-2 letters.

IV. CONSTRUCTION OF THE AUTOMATON (E,d,S) SOLUTION TO THE GENERAL PROBLEM POSED

Assume a connected network $\tilde{R}$ of n identical automata (E,d,S), $A_1, A_2, \ldots, A_n$.

To the edge u of the graph G associated with $\tilde{R}$ correspond two conjugate letters ℓ and $\bar{\ell}$ of the alphabet $\tilde{A}$ of G. One of the two letters, ℓ for example, has for left end the node i of G, and for right end the node j of G. We say then that the letter ℓ has for left edge (i,k), that is to say, the input address k of A_i, and for right edge (j,r), that is to say, the input address r of A_j.

We construct (E,d,S) in the following way:

(1) To each of the input addresses of (E,d,S) is associated an automaton (E,2,S) with q states, defined in paragraph II.

(2) The initial excitation of $\hat{A} = A_p$ consists of putting the automaton (E,2,S) of the first input address (p,k) of $\hat{A}$, the end of one edge of G, in state $\hat{M}$. The letter of G having for left edge (p,k) is designated by ℓ_1, for $\sigma(\ell_1)$ the treeword of G associated with ℓ_1.

(3) Each letter ℓ_i of $\sigma(\ell_1)$ is represented by its left edge. $\sigma(\ell_1)$ is therefore, by virtue of (1) and (3), a line of 2n-2 automata (E,2,S).

(4) In (E,d,S) there is an automaton component of the type $(E,d,\tilde{M})$ defined in paragraph III. This automaton is charged with the realization of $\sigma(\ell_1)$, that is, with the putting into line the 2n-2 automata of type (E,2,S).

(5) (E,d,S) is said to be at rest if its d components (E,2,S) and its component $(E,d,\tilde{M})$ are at rest. (E,d,S) is said to be in state F if at least one of its components (E,2,S) is in state F.

The automaton (E,d,S) so constructed is a solution to the problem posed because all the automata A_i of the connected network $\tilde{R}$ have at least one component (E,2,S) in $\sigma(\ell_1)$; whenever the (E,2,S) of $\sigma(\ell_1)$ are synchronized, all the A_i's are synchronized. It goes without saying that if $\tilde{R}$ is not connected, only the automata of the component connected to $\hat{A}$ are synchronized.

<u>Number of states</u>. (E,d,S) has d components with q states and a (d+1)-th component with k(d) states, which grants it in all $q^d x k(d)$ states. The number of states of (E,d,S) is therefore bounded from above by the number $8^d x 5^d$.

<u>Time-to-fire</u>. The time-to-fire for n automata arbitrarily connected in a connected network is given by:

$$\theta(n) = 2(2n-2)-2 = 4n-6.$$

In effect the realization of the word $\sigma(\ell_1)$ and the sending of the first signal down the line of automata constituted by $\sigma(\ell_1)$ are done simultaneously.

Figure (2) shows a network of 4 automata of degree 3 for which the components (E,2,S) are of the type described in paragraph II, and the history of their being put into synchrony in 10 units of time.

<u>Conclusion</u>. The placing into synchrony of n automata of arbitrary degree, on the initiative of an arbitrary one among them, needs, if the automata are arbitrarily connected, no more time than twice the time required for firing the "Firing Squad" of n soldiers on the initiative of a soldier at the end of the line. It requires on the contrary less time for one very special class of connection schemes, which allows the definition of ordered layers of automata (see the beginning of paragraph III); in this special case only, the time-to-fire is a function of d, which we notice is decreasing.

For the general case of arbitrarily connectable automata, the number of states, independent of n, is by contrast an exponential function in d. One should see this growth of the number of states as the price paid for the freedom permitted each in the way of being connected to the others by its d connections. We notice also that no automaton of the network has a privileged structure; each automaton, when given the chance, can become the general. The essential point is that there not exist two distinct initiatives of this kind.

The nature of the graph formed by arbitrary connections, and in particular the number of edges, plays no role in the calculation, since the automata cooperate in remembering only one tree of this graph. The automata once synchronized can forget everything in their environment which is external to the tree.

Once synchronized, our automata can also reject the clock which tells them their time, that is to say synchronizes their transitions. It suffices in effect to agree that the components $(E,d,\tilde{M})$ continuously <u>circulate</u> a mark on the cyclic word $\sigma(\ell_1)$ without ceasing, so that each round of this circulation counts as one unit of time. That which will then be called the synchronization of the transitions of the automata is the fact that an automaton will never have two transitions in advance of another. Our synchronized automata themselves constitute a clock.

REFERENCES

1. Balzer, R. M. "Studies Concerning Minimal Time Solutions to the Firing Squad Synchronization Problem", doctoral thesis, Carnegie Institute of Technology, Pittsburgh, 1966.

2. Berge, C. "Théorie des Graphes et ses Applications", Dunod, 1958.

3. Cole, S. N. "Real Time Computation by Iterative Arrays of Finite-State Machines", doctoral thesis, Harvard University, Mass., 1964.

4. Goto, E. "A Minimum Time Solution of the Firing Squad Problem" dittoed course notes for Applied Mathematics 298, Harvard University, 1962, pp. 52-59, (cited in Reference 5).

5. Moore, E. F. "The Firing Squad Synchronization Problem" in "Sequential Machines, Selected Papers", Addison Wesley, Mass., 1964, pp. 213-214.

6. Moore, E. F. "The Shortest Path through a Maze", Proceedings of International Symposium in the Theory of Switching, part II, Harvard University Press, Cambridge, 1959, pp. 285-292.

7. Rosenstiehl, P. "Codage des Graphes" - (to appear).

8. Tarry, G. "Parcours d'un labyrinthe rentrant", Association Francaise pour l'Avancement des Sciences, 1886, pp. 49-53.

9. Tarry, G. "Le problème des labyrinthes", Nouvelles Annales de Mathématiques XIV, 1895, pp. 187-190.

10. Waksman, A. "An Optimum Solution to the Firing Squad Synchronization Problem", Information and Control 9, 1966, pp. 66-78.

(n=2)

t		
t=1	$\hat{M}$	D
2	$\bar{M}$	M
3	F	F

(n=3)

t			
t=1	$\hat{M}$	D	D
2	$\bar{M}$	$\vec{R}$	M
3	M	$\hat{M}$	M
4	F	F	F

(n=4)

t				
t=1	$\hat{M}$	D	D	D
2	$\bar{M}$	$\vec{R}$	D	M
3	M	$\vec{1}$	$\vec{R}$	M
4	M	$\vec{2}$	$\overleftarrow{R}$	M
5	M	$\hat{M}$	$\hat{M}$	M
6	F	F	F	F

(n=5)

t					
t=1	$\hat{M}$	D	D	D	D
2	$\bar{M}$	$\vec{R}$	D	D	M
3	M	$\vec{1}$	$\vec{R}$	D	M
4	M	$\vec{2}$	D	$\vec{R}$	M
5	M	$\vec{3}$	D	$\overleftarrow{R}$	M
6	M	D	$\hat{M}$	D	M
7	M	$\hat{M}$	$\bar{M}$	$\hat{M}$	M
8	F	F	F	F	F

(n=6)

t						
t=1	$\hat{M}$	D	D	D	D	D
2	$\bar{M}$	$\vec{R}$	D	D	D	M
3	M	$\vec{1}$	$\vec{R}$	D	D	M
4	M	$\vec{2}$	D	$\vec{R}$	D	M
5	M	$\vec{3}$	D	D	$\vec{R}$	M
6	M	D	$\vec{1}$	D	$\overleftarrow{R}$	M
7	M	D	$\vec{2}$	$\overleftarrow{R}$	D	M
8	M	D	$\hat{M}$	$\hat{M}$	D	M
9	M	$\hat{M}$	$\bar{M}$	$\bar{M}$	$\hat{M}$	M
10	F	F	F	F	F	F

(n=7)

t							
t=1	$\hat{M}$	D	D	D	D	D	D
2	$\bar{M}$	$\vec{R}$	D	D	D	D	M
3	M	$\vec{1}$	$\vec{R}$	D	D	D	M
4	M	$\vec{2}$	D	$\vec{R}$	D	D	M
5	M	$\vec{3}$	D	D	$\vec{R}$	D	M
6	M	D	$\vec{1}$	D	D	$\vec{R}$	M
7	M	D	$\vec{2}$	D	D	$\overleftarrow{R}$	M
8	M	D	$\vec{3}$	D	$\overleftarrow{R}$	D	M
9	M	D	D	$\hat{M}$	D	D	M
10	M	D	$\overleftarrow{R}$	$\bar{M}$	$\vec{R}$	D	M
11	M	$\overleftarrow{R}$	$\overleftarrow{1}$	M	$\vec{1}$	$\vec{R}$	M
12	M	$\vec{R}$	$\overleftarrow{2}$	M	$\vec{2}$	$\overleftarrow{R}$	M
13	M	$\hat{M}$	$\hat{M}$	M	$\hat{M}$	$\hat{M}$	M
14	F	F	F	F	F	F	F

(n=8)

t								
t=1	$\hat{M}$	D	D	D	D	D	D	D
2	$\bar{M}$	$\vec{R}$	D	D	D	D	D	M
3	M	$\vec{1}$	$\vec{R}$	D	D	D	D	M
4	M	$\vec{2}$	D	$\vec{R}$	D	D	D	M
5	M	$\vec{3}$	D	D	$\vec{R}$	D	D	M
6	M	D	$\vec{1}$	D	D	$\vec{R}$	D	M
7	M	D	$\vec{2}$	D	D	D	$\vec{R}$	M
8	M	D	$\vec{3}$	D	D	D	$\overleftarrow{R}$	M
9	M	D	D	$\vec{1}$	D	$\overleftarrow{R}$	D	M
10	M	D	D	$\vec{2}$	$\overleftarrow{R}$	D	D	M
11	M	D	D	$\hat{M}$	$\hat{M}$	D	D	M
12	M	D	$\overleftarrow{R}$	$\bar{M}$	$\bar{M}$	$\vec{R}$	D	M
13	M	$\overleftarrow{R}$	$\overleftarrow{1}$	M	M	$\vec{1}$	$\vec{R}$	M
14	M	$\vec{R}$	$\overleftarrow{2}$	M	M	$\vec{2}$	$\overleftarrow{R}$	M
15	M	$\hat{M}$	$\hat{M}$	M	M	$\hat{M}$	$\hat{M}$	M
16	F	F	F	F	F	F	F	F

Figure 1. Histories of the "Firing Squads" of n soldiers for $n = 2,3,\ldots,8$.

t	A_1	A_2	A_3	A_4
	1 2 3	1 2 3	1 2 3	1 2 3
1	0 . 0	. . .	+ . .	. . .
	D D D	D D D	$\hat{M}$ D D	D D D
2	0 . 0	+ X .	+ . .	. . .
	D D D	$\vec{R}$ M D	$\bar{M}$ D D	D D D
3	0 X 0	+ X .	+ . .	. . .
	D $\vec{R}$ D	$\vec{1}$ M D	M D D	D D D
4	0 * 0	+ X +	+ . .	. . .
	D D D	$\vec{2}$ M $\vec{R}$	M D D	D D D
5	0 * 0	+ X +	+ . .	. . X
	D D D	$\vec{3}$ M D	M D D	D D $\vec{R}$
6	0 * 0	+ * +	+ . .	. . *
	D $\vec{1}$ D	D M D	M D D	D D $\overleftarrow{R}$
7	0 * 0	+ * +	+ . .	. . *
	D $\vec{2}$ D	D M $\overleftarrow{R}$	M D D	D D D
8	0 * 0	+ * +	+ . .	. . *
	D $\hat{M}$ D	D M $\hat{M}$	M D D	D D D
9	0 * 0	+ * +	+ . .	. . *
	D $\bar{M}$ D	$\hat{M}$ M $\bar{M}$	M D D	D D $\hat{M}$
10	0 * 0	+ * +	+ . .	. . *
	F F F	F F F	F F F	F F F

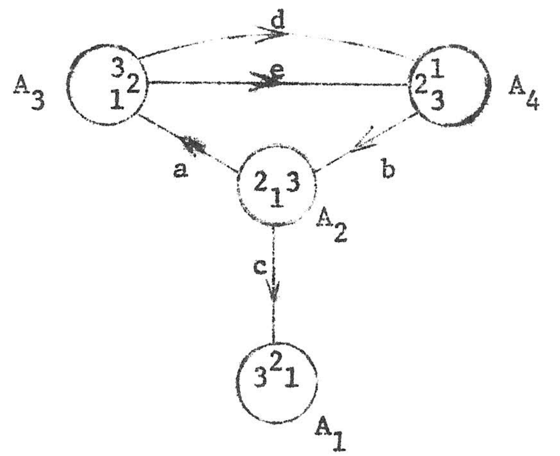

Figure 2. Network of four automata A_1, A_2, A_3, A_4 of degree 3 and the history of their being synchronized in 10 time steps when A_3 is excited. The automata components (E,2,S) utilized are of the type described in paragraph II.