

THE UNIVERSITY OF MICHIGAN
COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS
Department of Philosophy

Technical Note

COMMUTATIVE MACHINES

Richard Laing
Jesse B. Wright

ORA Projects 03105 and 04422

under contract with:

DEPARTMENT OF THE NAVY
OFFICE OF NAVAL RESEARCH
CONTRACT NO. Nonr 1224(21)
WASHINGTON, D.C.

and

OFFICE OF THE CHIEF SIGNAL OFFICER
RESEARCH AND DEVELOPMENT DIVISION
CONTRACT NO. DA-SIG-36-039-61-G4
WASHINGTON, D.C.

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

December 1962

COMMUTATIVE MACHINES*

Richard Laing and Jesse B. Wright

*This research was supported through Contract Nonr-1224(21) of the Office of Naval Research, and through Grant DA-SIG-36-039-61-G4 of the Office of the Chief Signal Officer.

COMMUTATIVE MACHINES

In this paper certain commutative automata and their behavior are discussed. We begin with characterizations of the class of machines under consideration. We then go on to show several associations between this class of machines and a set of words, and a set of points. We then show an association between a language for the set of words and the set of points. Some results about this language are presented, after which we show how machines can be constructed which realize the behavior expressed in this language. We conclude with results and comments on the characteristics of commutative machines and their transition systems.

I. BASIC TERMINOLOGY

A machine M is a quintuple $M = \langle S, s_0, fa, fb, T \rangle$ in which S is a (finite or infinite) set of states of M ; $s_0 \in S$ is the initial state of M ; fa, fb are the (direct) transition functions of M which take elements of S into elements of S ; T is a (finite or infinite) subset of S , called the terminal state(s) of M .

When for every $s \in S$, $s^{fafb} = s^{fbfa}$, then M is a commutative machine (c-machine).

We associate with each M a set of words. Let W be the set of all words in a and b (and including Λ , the null word). There is a useful association between the words in W and the states $s \in S$ of machines M . For any $w \in W$ and any $s \in S$ in a machine M , $w \approx s$ will mean that if the functions fa, fb are applied to M in the sequence indicated by the word w (reading from left to right) and starting in s_0 (and Λ applied to s_0 results in s_0), then the result will be s .

The behavior of a machine M is the set of all words w such that $w \approx s$ for some $s \in T$.

W is the free algebra, where $W = \langle W, \Lambda, fa, fb \rangle$ and where (as before) W is the set of all words in a and b , and Λ is the empty word, and where $w^{fa} = wa$, $w^{fb} = wb$, for all $w \in W$. Whenever we have $M = \langle S, s_0, fa, fb, T \rangle$ consider a mapping $H(w) : W \rightarrow S$. If $H(w) = s$, then s is the state into which M is taken by the input sequence w beginning in state s_0 . $H(w)$ is a homomorphism between W and M .

We now introduce the c -machine L which plays for commutative machines a role analogous to that of W for the class of all M . $L = \langle L, \lambda, fa, fb \rangle$ where L is the set of points in the plane with non-negative, integral coordinates (i.e., the lattice-points in the first quadrant) and where $\lambda \in L$ is the origin. For every $p \in L$, p^{fa} is to mean the point one unit to the right of p , and p^{fb} is to mean the point one unit above p . The functions fa, fb commute so that for every point $p \in L$, $p^{fafb} = p^{fbfa}$. L may be viewed as a commutative machine (without final states) where the $p \in L$ are the states of L .

Consider W and an M . If M is a c -machine then we can interpolate L between W and the c -machine M . Consider the mapping $P : W \rightarrow L$ which for any word w associates a corresponding lattice-point p ; $P(w) = p$, such that starting at λ and moving right one unit for each a in w and upward one unit for each b in w , we arrive at p . For any $p \in L$, $h(p) = s$ is to mean that if a c -machine is placed on $\lambda \in L$ while in state s , and moves right one unit under fa , and upward one unit under fb , then M will be in state s when resting on $p \in L$. (Alternatively, $h(p) = s$ can be interpreted " s is the state into which M is taken by a word w , where $P(w) = p$, with M started in s_0 .") Because the machine M is a c -machine, the point p it reaches in L is independent of the path it follows to p .

Consequently h is a well-defined, single-valued function. $H(w) = h(P(w))$.

Note that h is a homomorphism.

W , L , and an M , where M is a c -machine, are related in that if we apply the mapping from $W \rightarrow M$, it is the same as if we go from $W \rightarrow L \rightarrow M$. Thus the study of commutative machines is the same as studying all homomorphic images of L .

II. A LANGUAGE FOR DENOTING POINTS OF L .

The behavior of any machine is a set α of words w such that $w \approx s$ for some $s \in T$. In addition, if M is a c -machine then M has associated with its final states a set of points of L such that $h(p) \in T$. Regular expressions (see [1, 2]) have frequently been used to denote the set of words α which is the behavior of a finite machine. Under a different usage we can employ regular expressions to denote the behavior of a c -machine.

If β is a regular expression denoting a set of words, then let $c(\beta)$ be the commutative closure of the words of β . $c(\beta)$ is the behavior of a c -machine. If a c -machine M has a behavior $c(\beta)$ then $P(c(\beta))$ is the set of lattice points of L corresponding to the final states of M . Note that $P(\beta) = P(c(\beta))$. Regular expressions can thus be employed to denote the points of L which are associated with the final states of certain c -machines.

The use of regular expressions in denoting points of L is shown in the following figures.

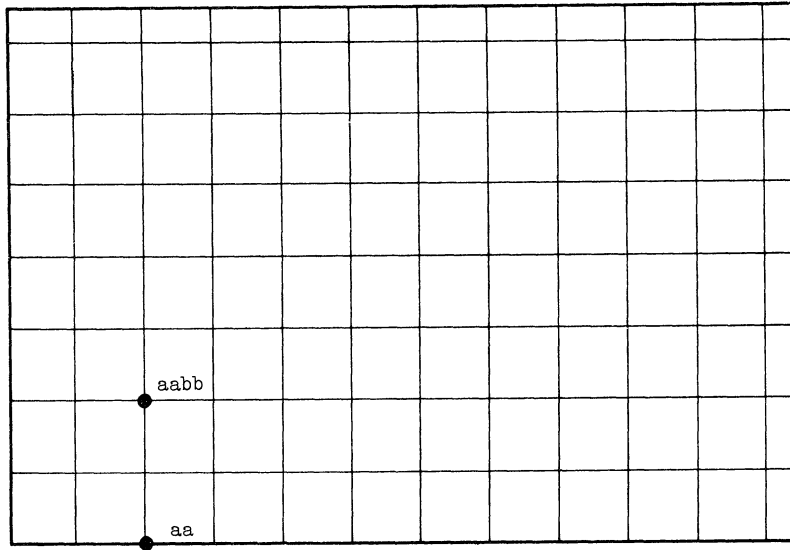


Figure 1

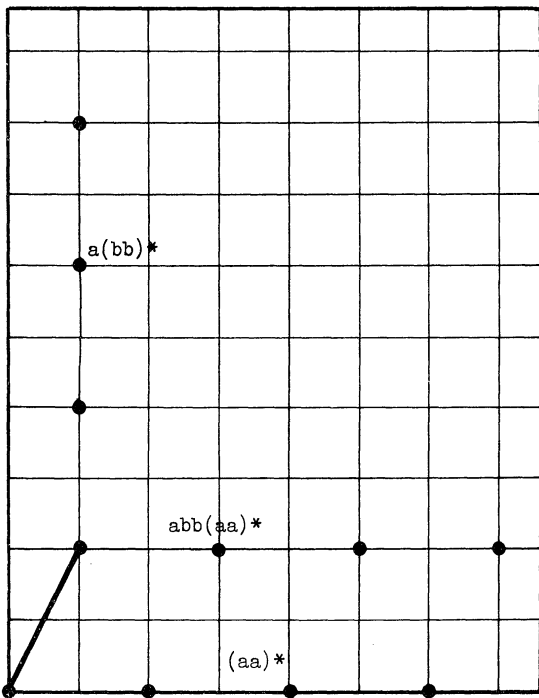


Figure 2

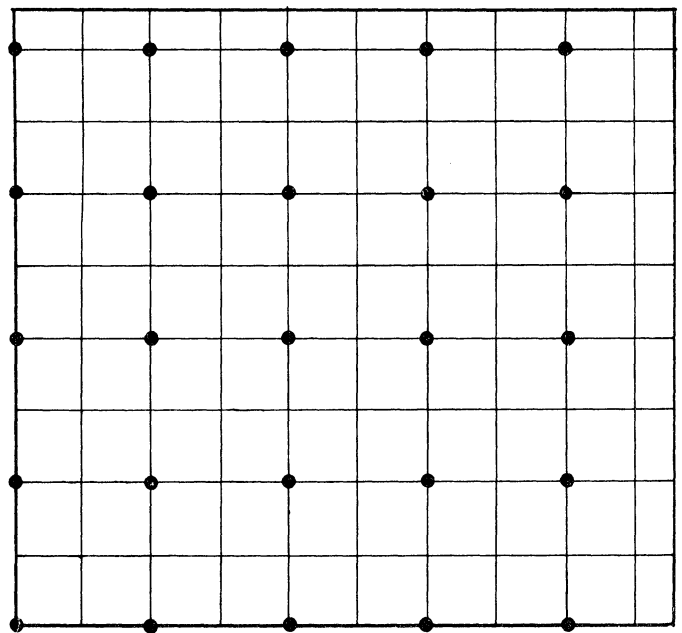


Figure 3. $\Lambda^{\sim}(aa)^*(bb)^*$.

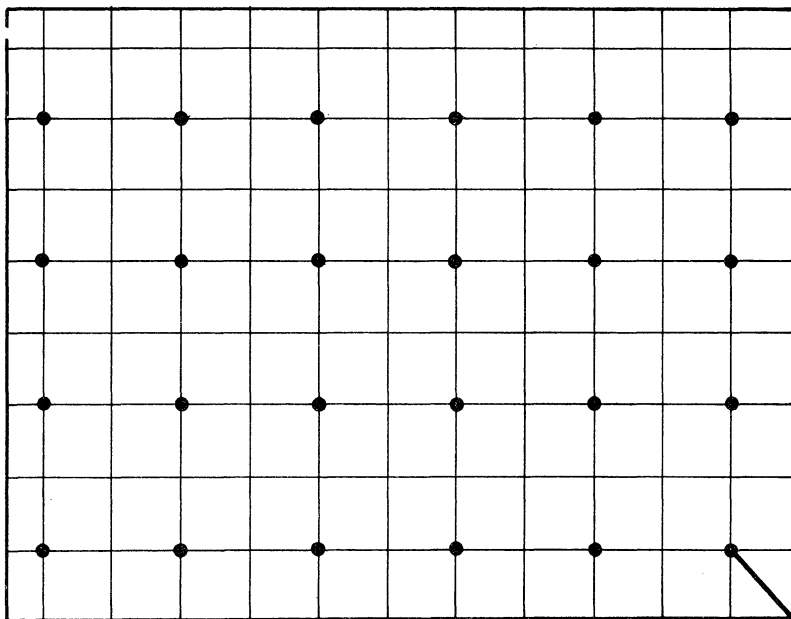


Figure 4. $ab(aa) *(bb) *$.

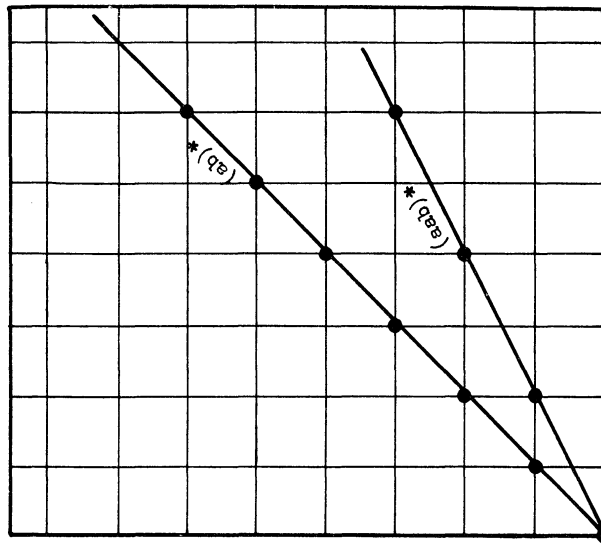


Figure 5

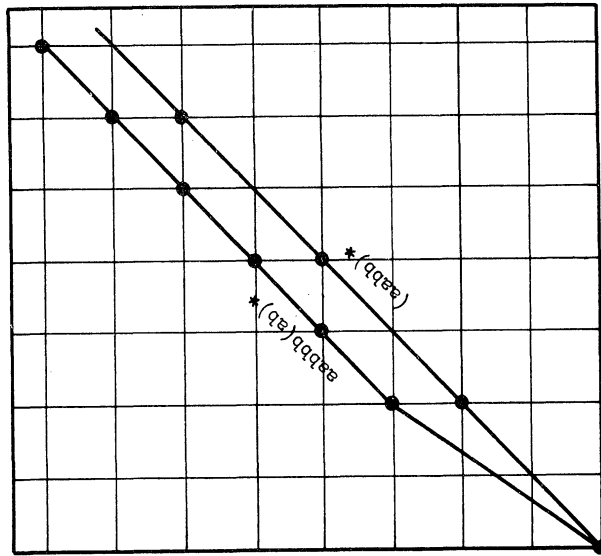


Figure 6

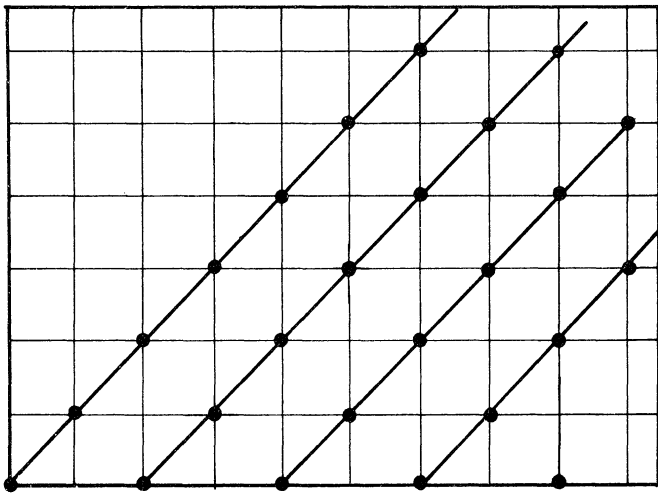


Figure 7. $(aa)^*(ab)^*$.

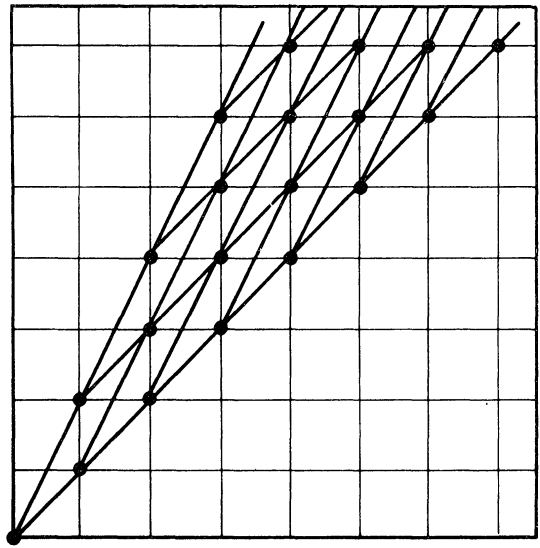


Figure 8. $(abb)^*(ab)^*$.

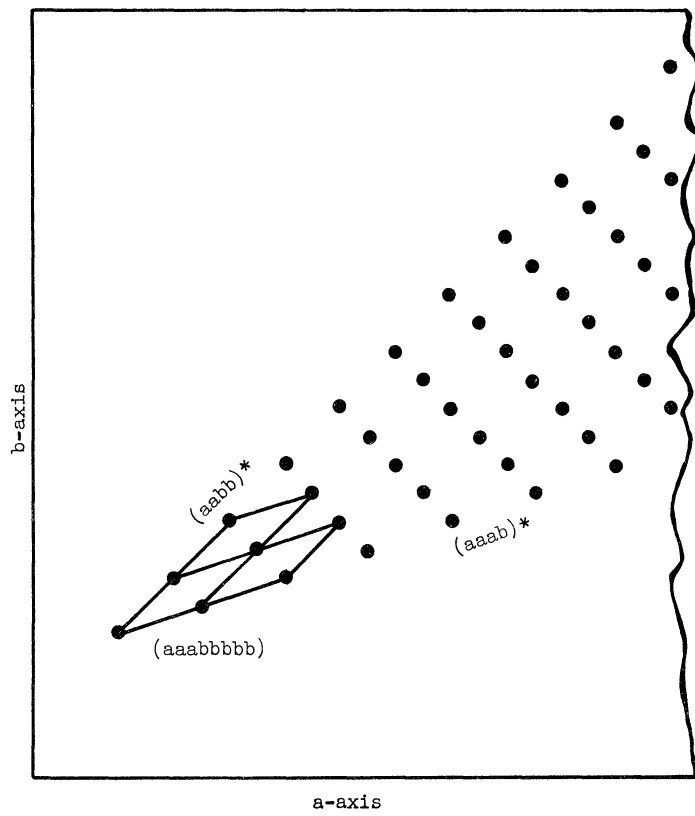


Figure 9. Fundamental set $(aaabbbb) \sim (aabb)^*(aaab)^*$

If a set of points P contained in L is denoted by a regular expression of the form $w_0 w_1^*$ or $w_0 w_1^* w_2^*$ (where w_0 may be Λ) then P is a fundamental set. (Also w_0 is called the origin of the fundamental set, and w_1^* , w_2^* are called the axes of the fundamental set.) If P is denoted by $w_0 w_1^* w_2^*$, where the points denoted by w_1 , w_2 along with the point λ , are not collinear, then P is a planar fundamental set. If P is denoted by $w_0 w_1^*$ and $w_1 \neq \Lambda$, then P is a linear fundamental set. (Thus the arrays of points shown in figures 3, 4, 7, 8, 9 are planar fundamental sets, and the points of figure 2, 5, 6 are linear fundamental sets.)

If the pattern of points of a planar fundamental set is extended beyond its axes over the whole first quadrant, then we will speak of such a set of points as the extended fundamental set of the given fundamental set.

When a fundamental set is described by an expression $w_1^* w_2^*$, where w_1 and w_2 are mixed words (in both a and b) then the lines of points w_1^* , w_2^* are oblique, and we will speak of such an expression as $w_1^* w_2^*$ as describing an oblique planar fundamental set (and similarly for linear fundamental sets). When an expression is of the form $w_1^* w_2^*$ where one of the two words w_1 , w_2 is in a 's alone and the other in b 's alone we will speak of the set as orthogonal.

Theorem: Every extended fundamental set derived from an oblique fundamental set is a union of orthogonal fundamental sets.

Two fundamental sets, $w_0 w_1^* w_2^*$ and $w_3 w_4^* w_5^*$ are said to be conjugate if $c(w_1^* w_2^*) = c(w_4^* w_5^*)$. Thus two fundamental sets are conjugate if, when they are given the same origin, they coincide. If two fundamental sets have their respective axes parallel they are called similar.

Theorem: If two fundamental sets are similar but not conjugate, they can be made conjugate by use of the Key Identity.

Lemma: Key Identity: Where x is a regular expression,
 $x^* = (\Lambda \cup x)^{k-1} (x^k)^*$.

The Key Identity allows infinite periodic sets of points in a line or in a planar array to be represented as a finite union of conjugate sets of the same kind. This changing of the period of the representation of an infinite set facilitates the comparison of infinite processes.

Theorem: Quasi-Normal Form. Where w_0, w_1, w_2 are any words of W , then any regular expression for denoting points of L can be put in the form of a finite union of constituents of the following kinds: $w_0, w_0^*, w_0 w_1^*, w_0 w_1^* w_2^*$.

The following identities are employed in obtaining this result.

Where x, y are regular expressions:

Universal Identities

1. $(x \cup y)^* = (x^* y^*)^*$
2. $x^* = (\Lambda \cup x)^{k-1} (x^k)^*$ (Key Identity)

Commutative Identities

1. $xy = yx$
2. $(x \cup y)^* = x^* y^*$
3. $(x^* y)^* = \Lambda \cup x^* y^* y$

This last identity allows the reduction of star-height of commutative regular expressions so that no star operates over a star.

Proof: [Note: $x^* = \Lambda \cup x \cup x x \cup \dots$]

$$\begin{aligned} (x^* y)^* &= \Lambda \cup (x^* y) \cup (x^* y)^2 \cup \dots \\ &= \Lambda \cup (x^* y) \cup (x^* x^* y y) \cup \dots \\ &= \Lambda \cup (x^* y) \cup (x^{*2} y^2) \cup \dots (x^{*n} y^n) \dots \end{aligned}$$

[Note: $x^{*2} = x^* x^* = (x \cup x)^* = x^*$ (and likewise for the nth power)]

$$\begin{aligned} &= \Lambda \cup x^* (y \cup y^2 \cup y^3 \dots y^n \dots) \\ &= \Lambda \cup x^* y^* y \end{aligned}$$

Thus, beginning with an arbitrary regular expression, and employing in particular the commutative identities 2 and 3 above, an expression is obtained composed of a union of constituents within which there are no unions and having no stars operating over starred expressions. Some of the resulting constituents may, however, contain long concatenations of starred words. Constituents of this sort may be brought into the proper form by use of the following result:

Lemma: Star-length Reduction. Any constituent of the form $w_0^* w_1^* w_2^*$ can be placed in the form $(\Lambda \cup w_0 \cup w_0^2, \cup \dots \cup w_0^k) \overbrace{w_1^* w_2^*}$.

Thus, beginning with a constituent composed of a concatenation of any length, of starred words, a portion of star-length three of the constituent can be reduced to star-length two. This process can be repeated, and, with applications of the distributive ^{law,} the whole expression reduced to the desired form.

Proof: Case 1: At least two of w_0^*, w_1^*, w_2^* lie on the same line. Similar linear fundamental sets are thus established, which by the Key Identity can be made conjugate, and the result is obtained.

Case 2: No two of w_0^* , w_1^* , w_2^* lie on the same line. Because of the commutativity identity we can assume without loss of generality that w_0^* is the line of points lying between w_1^* and w_2^* . $w_1^* w_2^*$ can be viewed as determining a planar fundamental set, and w_0^* can be viewed as providing an infinite set of relative origins for $w_1^* w_2^*$, i.e., establishing an infinite series of conjugate fundamental sets of the form $w_1^* w_2^*$.

Of two conjugate fundamental sets, F , F' , if the origin of F is contained in F' , then $F' \not\subseteq F$.

A linear fundamental set (an infinite linear periodic set of points) within a planar fundamental set, and having a point in common with the planar fundamental set, has an infinite periodic subset (a linear fundamental set) in common with the planar fundamental set (from Chinese Remainder Theorem).

Thus since w_0^* and $w_1^* w_2^*$ have the origin in common, an infinite periodic subset of w_0^* is contained in $w_1^* w_2^*$. In particular there exists a number such that w_0^* raised to that number is an element of $w_1^* w_2^*$. Let k be that number. As long as w_0^i , $i < k$, then the relative origin chosen from w_0^* does not satisfy the hypothesis that the origin of F' is in F . When i reaches k , at that point and beyond (because of the periodic nature of the process) every element w_0^i , for $i \geq k$ is an element of

$$(\Lambda \cup w_0^k) \cap w_1^* w_2^* .$$

Given any planar fundamental set F , there can be, in L , only a finite number of zero intersection fundamental sets conjugate to F . Given any linear fundamental set, there can be on a line in L only a finite number of zero-intersection fundamental sets conjugate to the given set. The number, ℓ , of

possible conjugate fundamental sets of a given kind F , (including F itself) under the above conditions will be called the index of F . Note that in the Star-length Reduction the index of $w_1^* w_2^*$ is a bound on k .

As will be seen in the section on construction, certain classes of commutative machines realizing behaviors expressed as the points of fundamental sets will prove to have minimal numbers of states related to the index of the associated fundamental sets.

Lemma: Given two regular expressions E and E' in quasi-normal form, denoting points of L , it can effectively be decided whether $E = E'$, i.e., whether E and E' denote the same points of L . Detailed proof will not be given.

Proof is obtained by first applying techniques for determining coincidence in the constituents within E and within E' . After this redundancy in E and E' separately has been eliminated, E and E' are examined together for coincidence and difference. Important tools in facilitating the making of these comparisons are results in elementary number theory such as the Euclidean Algorithm, the Chinese Remainder Theorem, and such intermediate results as the Key Identity and the fact (already mentioned) that if a planar fundamental set and a linear fundamental set lying wholly between the axes of the planar fundamental set have a point in common, then they have a linear fundamental set in common, and a regular expression for this set can be obtained.

Note also that if E is any regular expression in quasi-normal form, with constituents $e_0, e_1, e_2, \dots, e_n$, then $c(E) = c(e_0) \cup c(e_1) \cup \dots \cup c(e_n)$.

Theorem: Given any two regular expressions, E and E' it can be decided whether $c(E) = c(E')$. This follows from the last Lemma, and from the earlier result that any regular expression can be put into quasi-normal form.

III. CONSTRUCTION OF COMMUTATIVE MACHINES

We have shown how regular expressions can be employed to denote points of L , our lattice-point model. These points of L in turn are employed to denote all the legitimate paths to the points. The words which are associated with these paths are the behavior of the commutative machines under consideration.

We will consider in some detail the construction of machines which realize the behavior denoted by certain expressions in quasi-normal form. Some of these machines will prove to be finite state devices, and some of them will be infinite state devices.

The use of the lattice-point model L as a guide to the constructing of state-diagrams for both finite and infinite machines is central to our discussion. Any commutative event can be represented in a set of lattice points α of L . The process of producing a machine M consists in merging points of L relative to α and α' (complement of α) to form states of M . Consider an operator μ (merge) which is to be applied to sets of points of L

Definition: $\mu(\alpha)$ is a c-machine.

Definition: Let $x \sim y$ mean x and y can be merged; then $x \sim y$ if and only if for all u 's, $xu \in \alpha$ is equivalent to $yu \in \alpha$, where x, y, u are elements of L . \sim is thus defined.

Definition: $\mu(\alpha) = L \text{ (mod } \sim)$ (the quotient of the free algebra L where we factor out \sim) with the \tilde{x} designated an output if and only if $\tilde{x} \in \alpha$ (\tilde{x} is the set of all points related to x by \sim).

Lemma: \sim is an equivalence relation on L and is a right congruence.

(It is in fact a bi-congruence.) Thus: the two input operations of L induce operations on the equivalence classes by choosing representatives and the result is independent of the choice of these representatives. Every equivalence class is either entirely in or entirely out of α .

Since we wish to take the equivalence classes as states we must show the transitions between these states. From the Lemma it is apparent that the next state under an a -transition for example, can be obtained by taking a representative of \tilde{x} (x itself perhaps) and finding the point one unit to the right. The \sim -class of this point is the state sought under an a -transition. Similarly for a b -transition.

Theorem: The machine $\mu(\alpha)$ is the minimal machine having behavior α .

Certain points cannot be merged. Consider α and any merging relation \sim for which $L \pmod{\sim}$ has behavior α . Given α , $x \sim y$ implies $x u \in \alpha$ if and only if $y u \in \alpha$. Thus: If a u can be found such that $x u \in \alpha$ and $y u \in \alpha'$, then x, y cannot be merged.

Note that if F is an oblique fundamental set and x, y are two points, then if the line through x, y is not parallel to F , we cannot merge x, y .

Regularity Criterion

If a set of words is non-regular, the set will not be "acceptable" by the finite machine. All finite sets of words are regular. An infinite set α is non-regular if there is an infinite set β with an infinite sub-set such that if you can find two elements (words) of β such that when you add a word to one of the elements then the resulting word is in α , but when you add it to the second element of the pair then the result is not in α .

If such a β cannot be found, then the set α is regular.

Corollary: The set of words associated with any oblique fundamental set is non-regular.

Corollary: The set of words associated with any oblique fundamental set is not acceptable by a finite automaton.

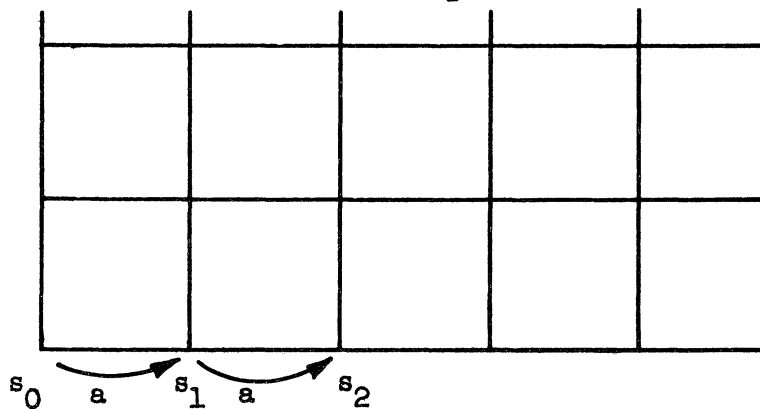
We now apply these results to construct machines realizing various behaviors. The behaviors first considered satisfy the Regularity Criterion above and thus prove to be realizable by finite machines.

Finite state machines.

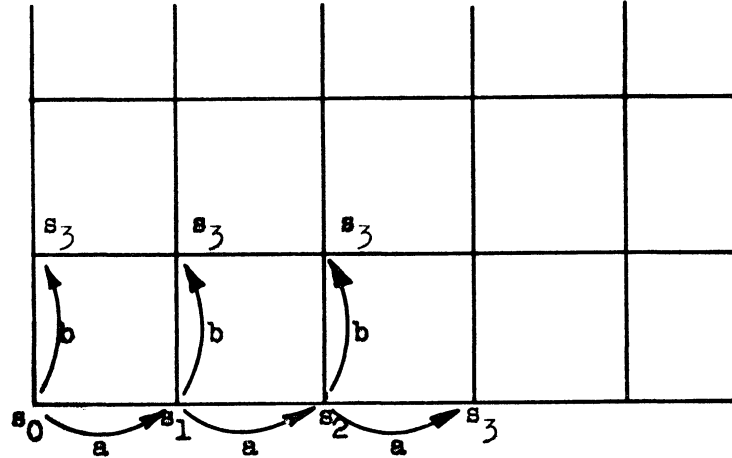
The state diagram for the behavior denoted by $c(w)$ where w is an unstarred word can be easily obtained. The method to be employed will be made apparent by use of example. Let us attempt to realize $c(aa)$ and also $c(aabb)$. (Figure 1 shows the points denoted by these expressions.)

Example 1: $c(aa)$: The commutative closure of aa is again aa . A state diagram is to be obtained of a machine which will "detect" or "accept" the word aa and no other words, in the sense that application of the sequence aa will place the machine in an $s \in T$, and no other word will place the machine in an $s \in T$.

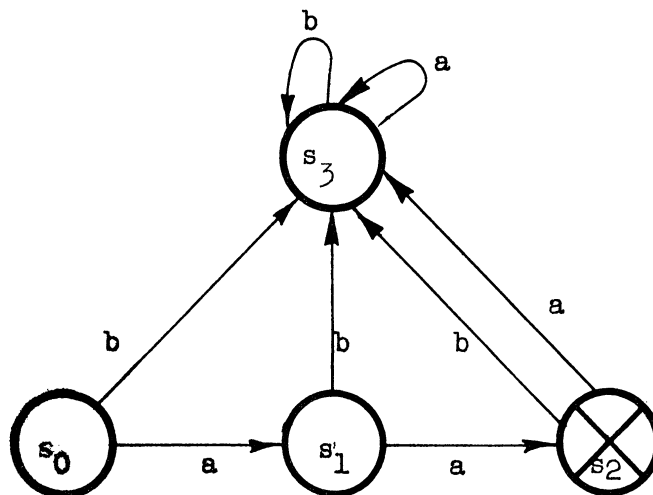
Using L as a guide, and identifying λ of L with the initial s_0 of a machine M , then $s_0 \xrightarrow{a} s_1$, $s_1 \xrightarrow{a} s_2$ (read " s_0 by a takes M to s_1 " etc.). This is the only acceptable path (word), and $s_2 \in T$.



No other words are to result in placing M in an $s \in T$. In this particular M , receipt of any b whatsoever, or any number of a 's greater than two, will take the machine into a "limbo" state: a state which is not an $s \in T$, and which the machine can never leave. In this case s_3 is a limbo state.



Thus all points of L other than s_0, s_1, s_2 are limbo states. Since all limbo states have the same behavior, viz. upon receipt of either a or b the machine remains in the limbo state, all limbo states may be merged into a single state. The state diagram for an M realizing $c(aa)$ is:



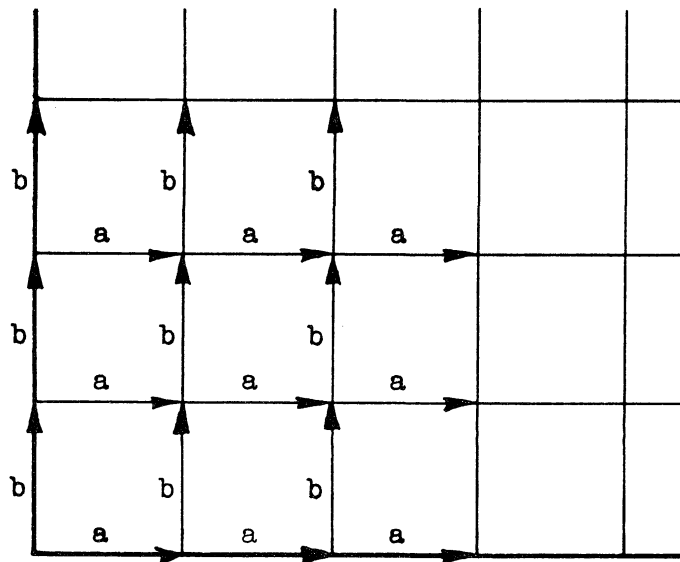
In order to indicate an $s \in T$ we will often mark the state with an x . Also, to secure neatness of state diagrams we will often abbreviate the transitions to limbo by $\rightarrow S_L$.

Thus the machine $M = \langle s_0, S, f_a, f_b, T \rangle$ realizing $C[(aa)]$ is:
 $s_0 \in S, S = \{s_0, s_1, s_2, s_3\} T = \{s_2\}$, and the transition functions f_a, f_b are:

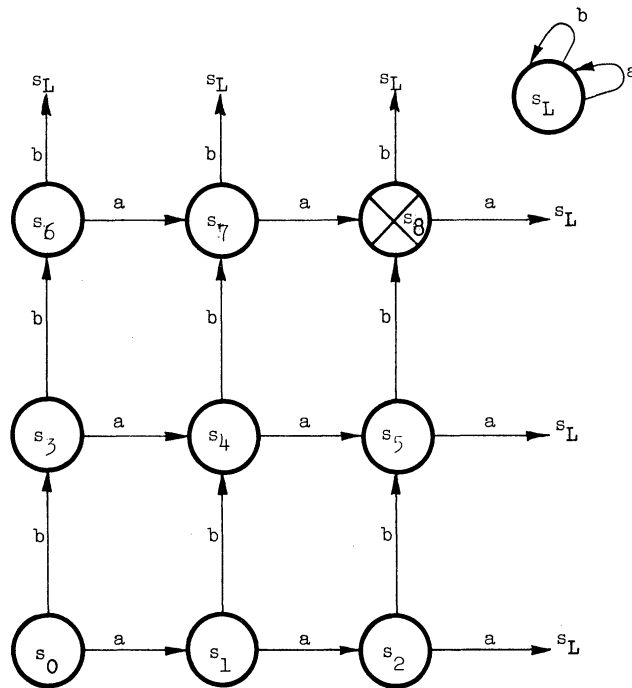
or (alternative notation)

$s_0 f_a = s_1$	$s_0 \xrightarrow{a} s_1$
$s_0 f_b = s_3$	$s_0 \xrightarrow{b} s_3$
$s_1 f_a = s_2$	$s_1 \xrightarrow{a} s_2$
$s_1 f_b = s_3$	$s_1 \xrightarrow{b} s_3$
$s_2 f_a = s_3$	$s_2 \xrightarrow{a} s_3$
$s_2 f_b = s_3$	$s_2 \xrightarrow{b} s_3$
$s_3 f_a = s_3$	$s_3 \xrightarrow{a} s_3$
$s_3 f_b = s_3$	$s_3 \xrightarrow{b} s_3$

Example 2: $c[(aabb)]$:



Lattice Model (aabb)

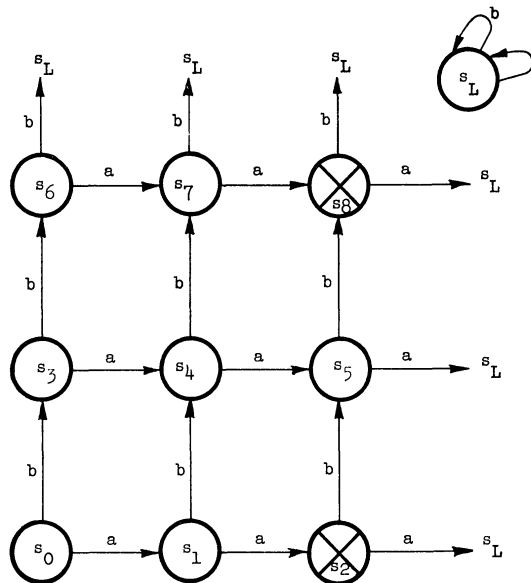


c-machine $c(aabb)$ ($s_8 \in T$)

This machine accepts aabb, abab, abba, bbaa, baba, baab, and no other words.

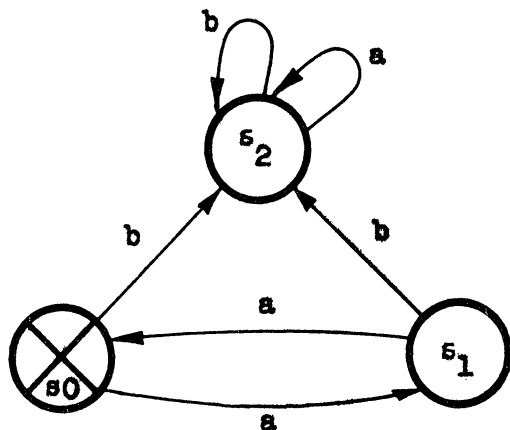
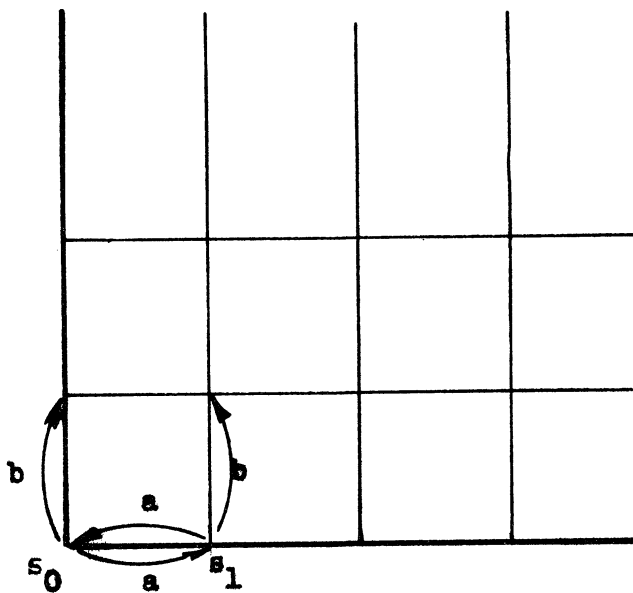
Example 3: $c(aa \cup aabb)$:

In realizing the behavior of a finite union of unstarred words, the resulting machines can always be at least partly combined.



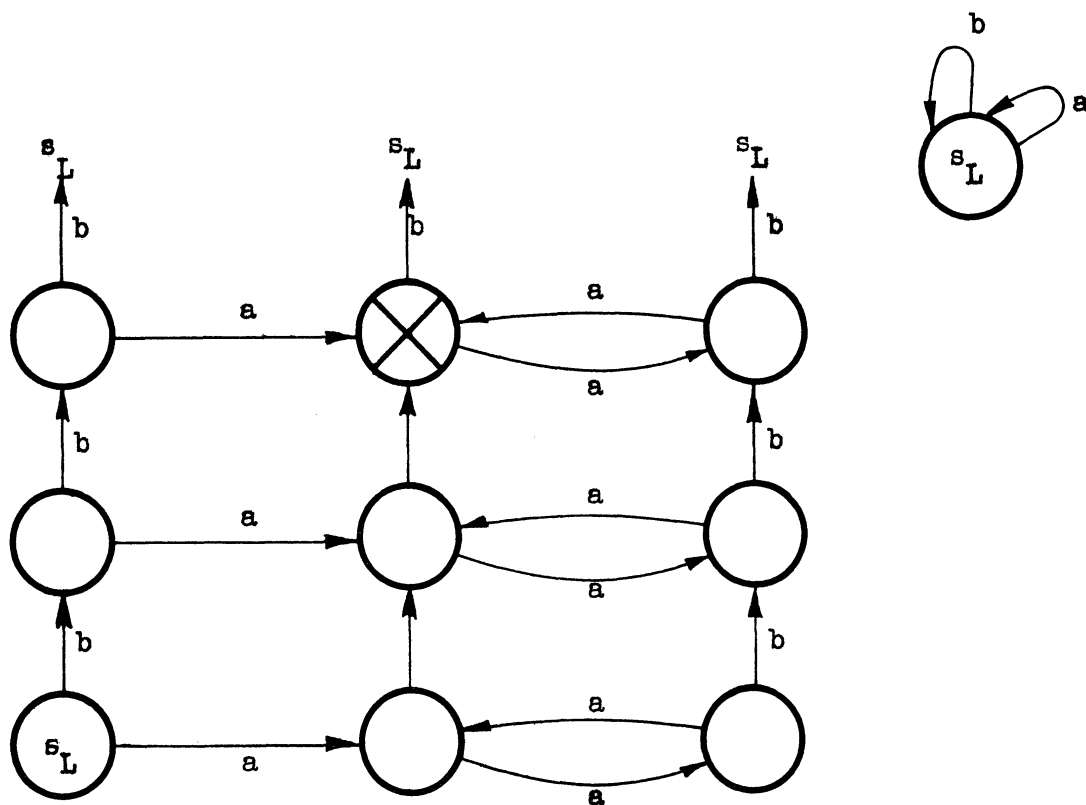
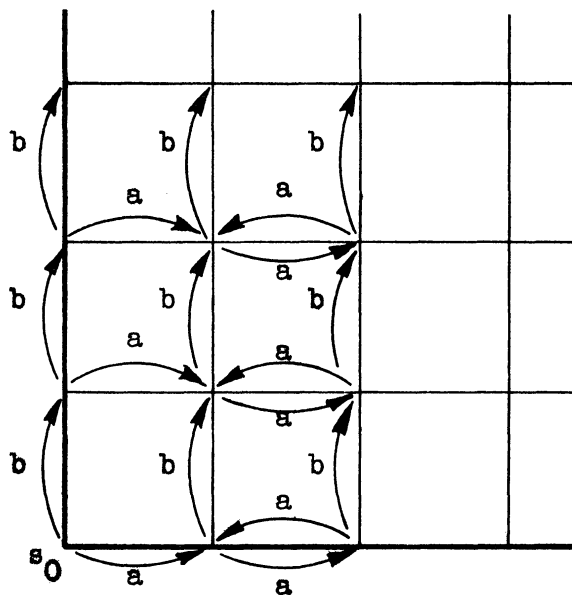
By distinguishing both s_2 and s_8 as elements of T we can realize $c(aa)$ and $c(aabb)$ in the "same" machine.

Example 4: $c(aa)^*$ (See Figure 2):

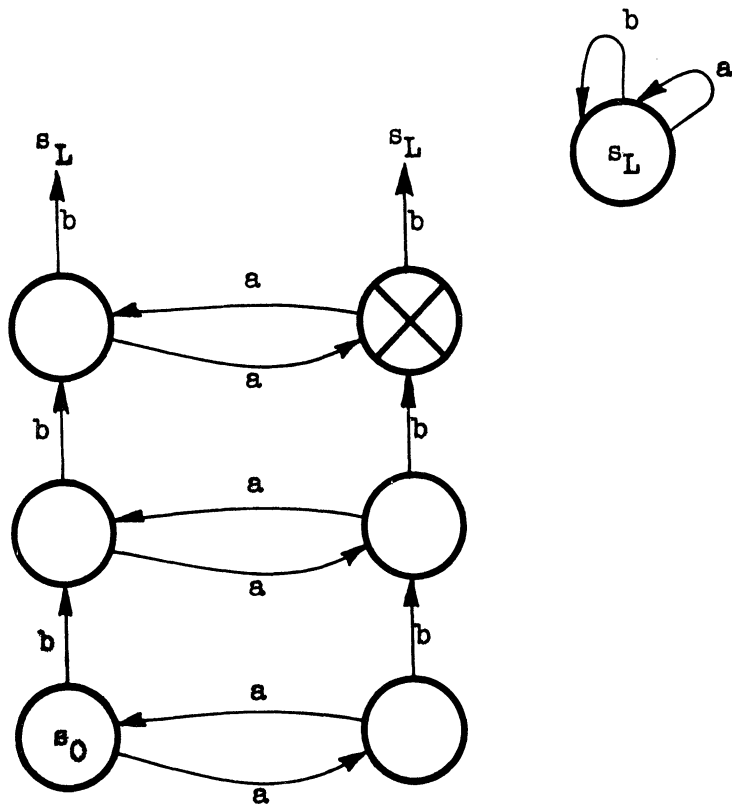


Note: (1) The index of $(aa)^*$ is 2. If the points (states) on the a-axis are numbered respectively 0, 1, 2, ... etc., then all the even numbered states can be merged into the single state s_0 , while all odd numbered states on the a-axis can be merged into the single state s_1 . (2) As before, the remaining states (points) of L are merged in the single limbo state s_2 . (3) Since Λ is an element of $c(aa)^*$ the machine, in s initially, also is "on", initially, under receipt of Λ .

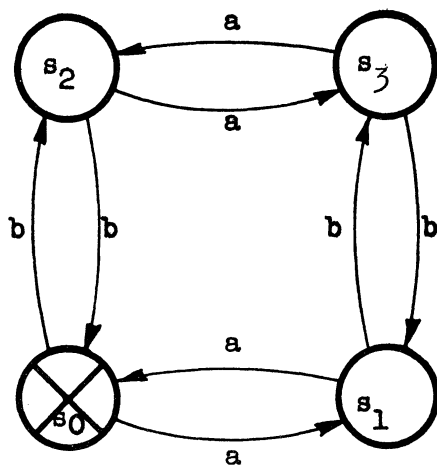
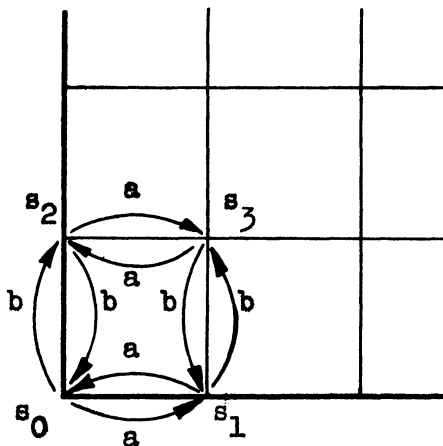
Example 5: $c(abb^{\wedge}(aa)^{*})$:



But note that parts of the above machine can be merged reducing the machine from 10 states to a 7 state minimal machine for $c(abb^{\wedge}(aa)^{*})$.



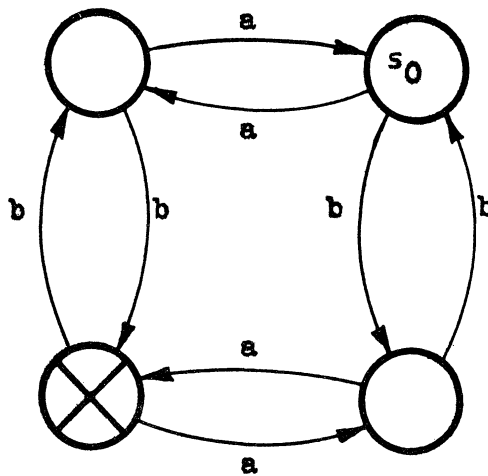
Example 6: $c((aa)^*(bb)^*)$ (See Figure 3):



There are four fundamental sets conjugate to $\Lambda^{\wedge}(aa)^*(bb)^*$:
 $\Lambda^{\wedge}(aa)^*(bb)^*$ itself, $a^{\wedge}(aa)^*(bb)^*$, $b^{\wedge}(aa)^*(bb)^*$, and $(ab)^{\wedge}(aa)^*(bb)^*$.
 All the points of L which are "output" points, viz, $\Lambda^{\wedge}(aa)^*(bb)^*$ are merged into $s_0 \in T$; all the points of L which we denoted by the other three fundamental sets are merged to form states s_1, s_2, s_3 respectively.

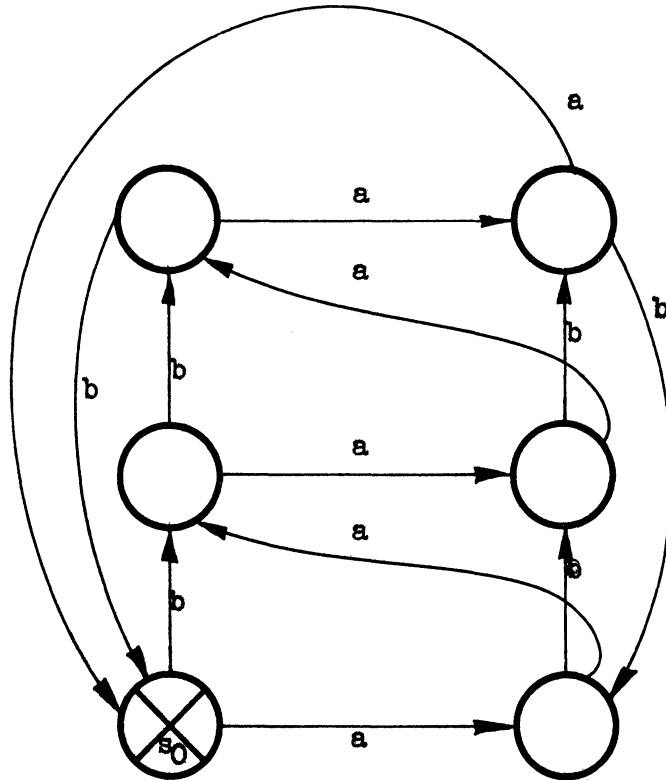
Note that the index, ℓ , of the fundamental set $(aa)^*(bb)^*$ is 4, and that this is the minimal number of states for the machine realizing this behavior. This is true in general of planar fundamental sets which denote the behavior of finite machines. For linear fundamental sets, because of the limbo state, the minimal machine realizing finite behavior is $\ell + 1$.

Example 7: $c(ab(aa)^*(bb)^*)$:



Compare Example 6. By designating a different initial state the behavior is modified without additional states. Note that by thus choosing alternative initial states the behavior denoted by each of the conjugate fundamental sets can be obtained.

Theorem: If a planar fundamental set is extended over the first quadrant, then the minimal c-machine having the behavior denoted by the set of points has a number of states equal to the index of the extended fundamental set. In some cases the minimal machines associated with extended fundamental sets may be "skewed" as in the following figure.



Example 9: $c((\Lambda \cup aabb \cup aaaab) \frown (aaaaaa)^*(bbb)^*)$

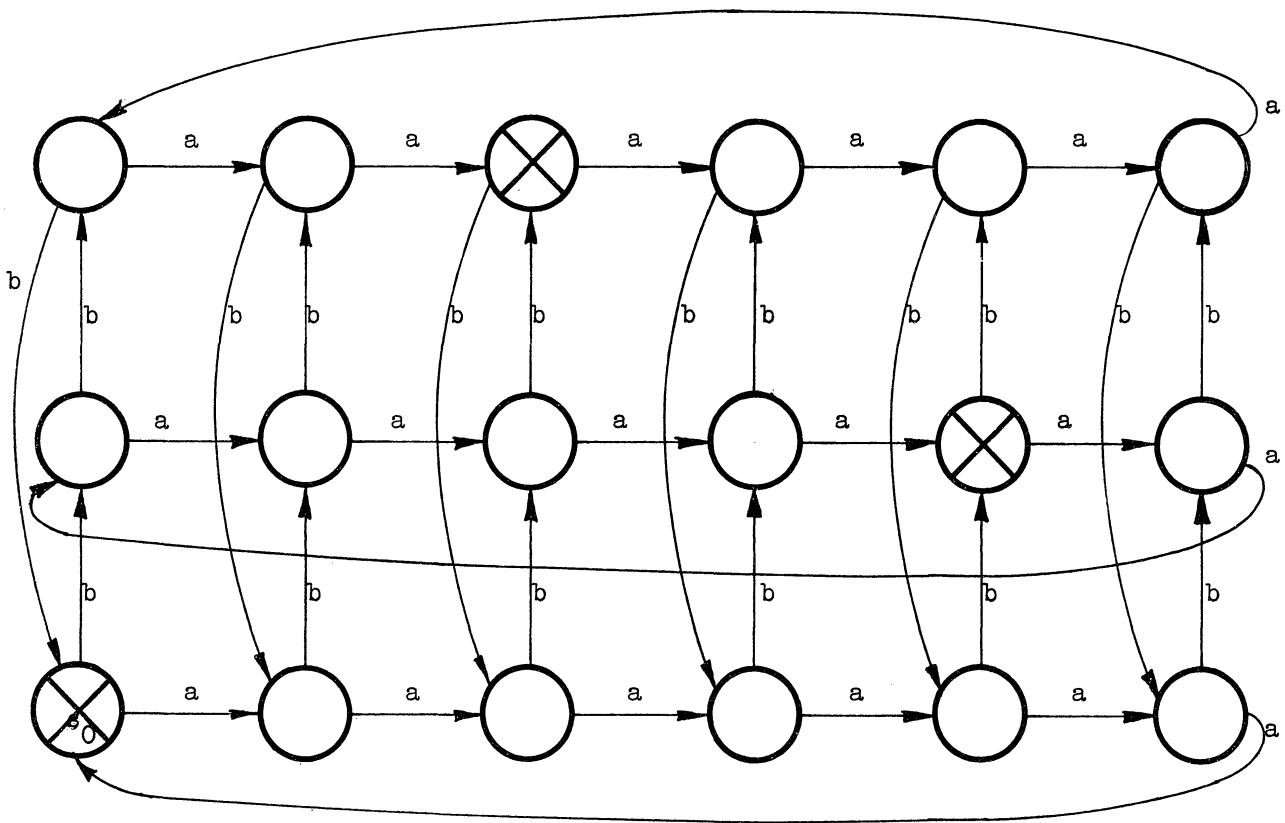
This machine realizes the behavior denoted by the oblique fundamental set $(aabb)^* \frown (aaaab)^*$ extended over the first quadrant. Recall from Section II:

Theorem: For every fundamental set with oblique axes, there is a union of fundamental sets with orthogonal axes (axes in a's alone and b's alone) such that this union is just the points of the oblique fundamental set extended over the first quadrant.

Corollary: For every minimal machine which is skewed, there is an "orthogonal" machine with the same behavior, (having usually several $s \in T$).

Thus the behavior of a minimal skewed machine can be realized by a direct product of counters.

The "orthogonal" (18-state) machine realizing the same behavior as the minimal (6-state) "skewed" machine of the last example, is shown below.



Since the machines of Examples 1-9 are all finite, there exist ordinary regular expressions [1] denoting their behavior, thus:

Comment: Where β is a regular expression in quasi-normal form, and where $c(\beta)$ denotes the behavior of a finite c-machine, then for every $c(\beta)$ there is a corresponding Kleene regular expression for the behavior of the machine.

For example, Lawrence Eggen has shown that an ordinary regular expression for $c((aa)^*(bb)^*)$ is $(aa \cup bb \cup (ab \cup ba)(aa \cup bb)^*(ab \cup ba))^*$.

An ordinary regular expression for the behavior of a c-machine can be obtained by the Analysis technique applied by Kleene. This technique, however, does not guarantee any minimality properties of the expression. In particular, the star-height of the resulting ordinary regular expression may not be minimal.

Comment: Where the behavior $c(\beta)$ requires an infinite c-machine there can be no ordinary regular expression for this behavior.

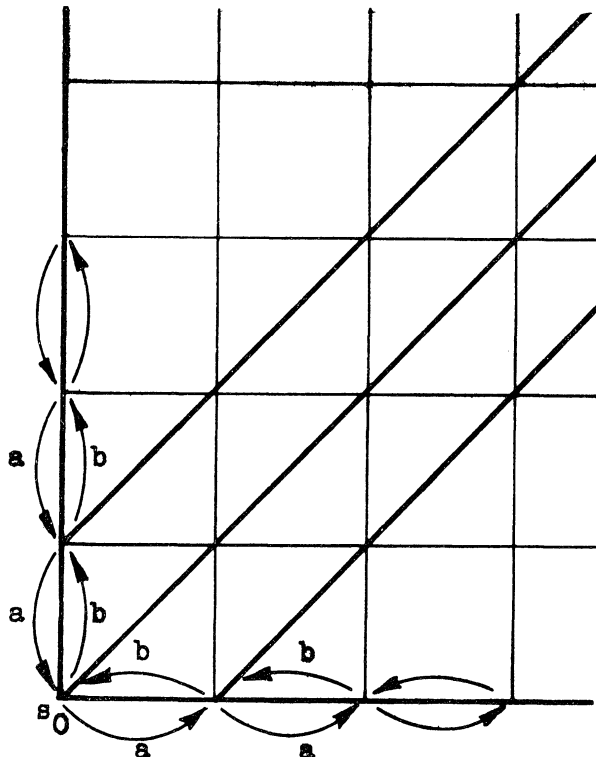
Comment: For any regular expression, it can be decided whether it denotes the behavior of a c-machine. (Using the Kleene Synthesis result, construct the machine realizing the behavior denoted by the regular expression. Minimize the resulting machine [3]. Test each of the states of this machine for the commutativity condition, $s_fafb = s_{fbfa}$.)

Infinite Commutative Machines

Some behaviors describable by employing the c -operator on regular expressions in quasi-normal form cannot be realized by finite machines.

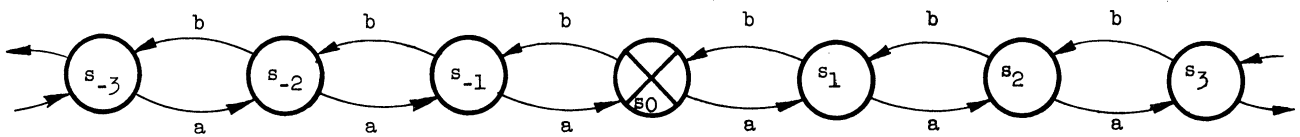
This will be indicated by means of the following examples: Consider the following set: $c(ab)^* \wedge a^*b^*$. a^*b^* is regular, and if $c(ab)^*$ is regular, then the intersection of the two sets should also be regular. The intersection consists of all sequences in equal numbers of a 's and b 's intersected with all sequences in which a 's precede b 's. That is to say, all sequences of the form $a^n b^n$, (where n ranges over all positive integers and zero). In order to detect the sequences of this set, the machine will be required to register the receipt of a^n and upon the receipt of b^n indicate acceptance of the sequence. But n can be indefinitely large, and indeed an n can always be found which will exceed the registering capacity of any given finite machine. Thus, to realize the behavior $c(ab)^*$ a potentially infinite machine is required, and $c(ab)^*$ is not regular.

Example 10: $c(ab)^*$:

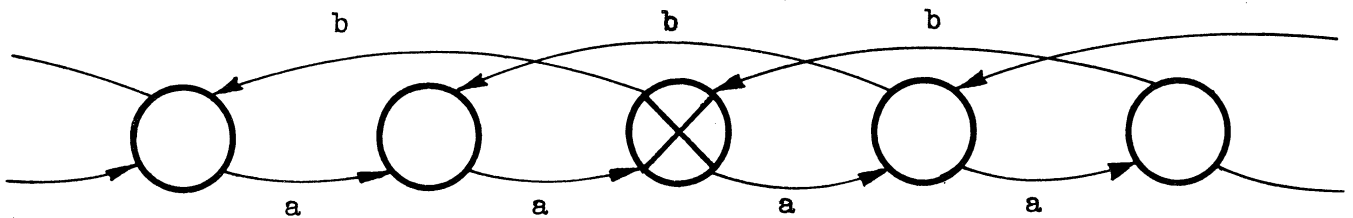


All of the points which lie on the line $(ab)^*$ can be merged into the single origin point (state) s_0 , which is also the output state. In addition, all of the points on lines parallel to $(ab)^*$ can be merged with their respective origins on the a and b axes.

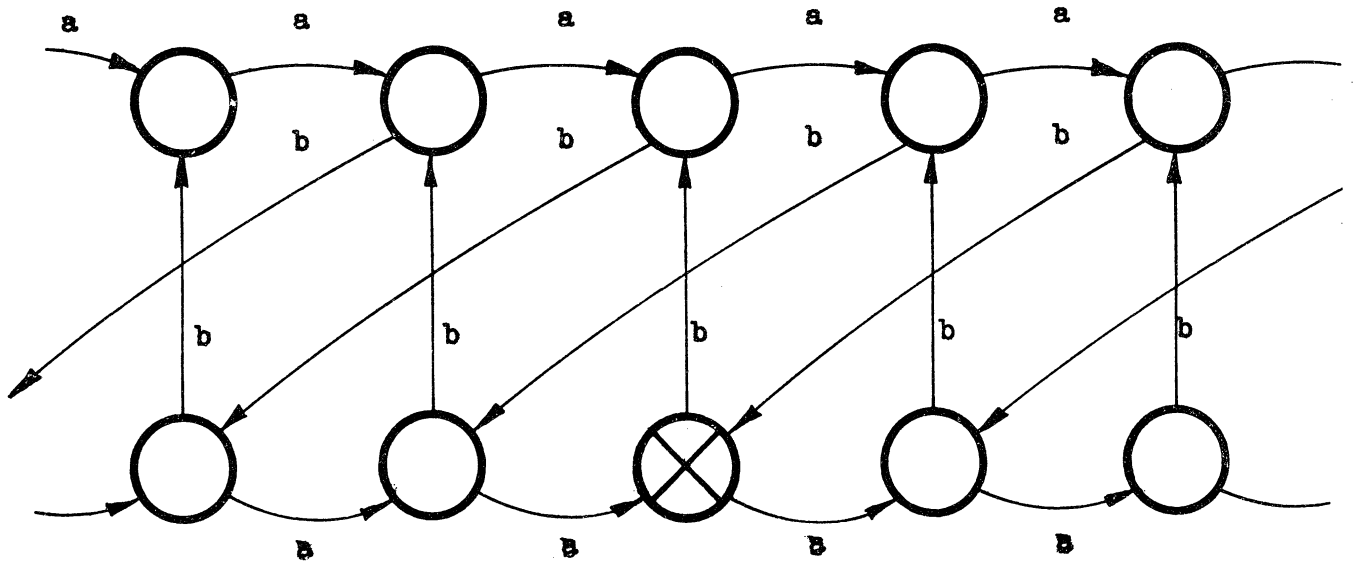
The state diagram for the infinite machine realizing $c(ab)^*$ can be set down in a linear fashion.



Example 11: $c(aab)^*$:

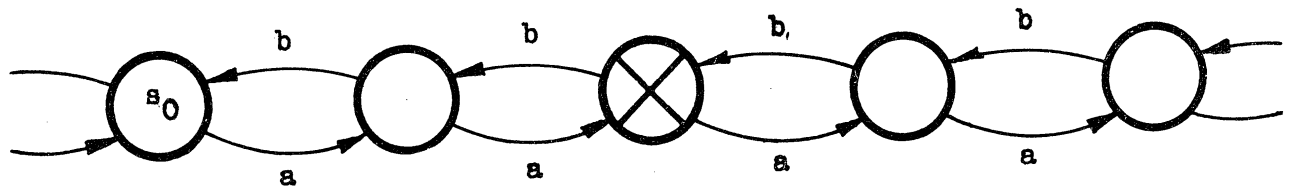


Example 12: $c(aabb)^*$



In this case alternate points on the main diagonal and on lines parallel to the main diagonal map into upper and lower rows of points (states). The number of rows of points is equal to the index of the linear fundamental set which denotes the behavior of this machine.

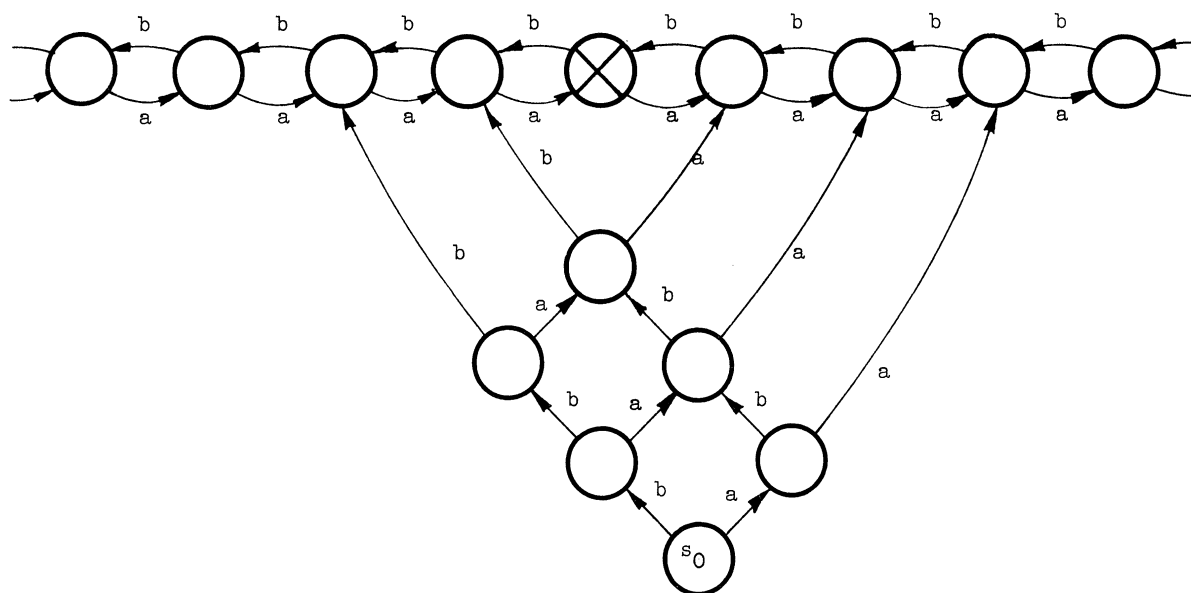
Example 13: $c(aa^{\sim}(ab)^*)$:



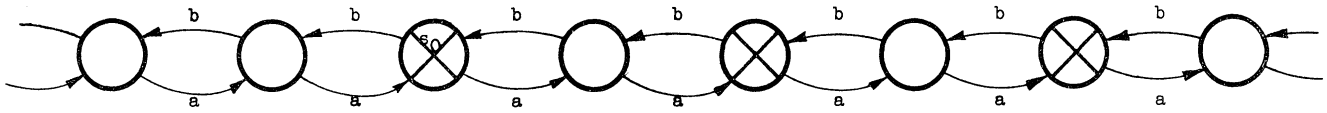
In this case the initial and output states are not identical. The prefixed word aa can readily be accommodated as can any prefix in a's and b's alone by displacing the initial state the requisite number of proper transitions from the output state.

Example 14: $c(aabb^{\sim}(ab)^*)$:

In order to realize this behavior, a "feeder" machine system must be employed. (See Example 8 for a feeder system to a finite machine.)



Example 15: $c((aa)^*(ab)^*) :$



This behavior can be realized by distinguishing an infinite number of states as output states: in this case alternate states to the right of s_0 . This technique can be applied to all cases where a pure word starred is concatenated with a mixed word starred.

Example 16: $c((aaabbbbb)^\wedge (aabb)^*(aaab)^*)$

The only "machine" that realizes this behavior is the machine M whose state diagram is L and whose output set T is those points of L denoted by $c((aaabbbbb)^\wedge (aabb)^*(aaab)^*)$.

Although points in lines parallel to oblique lines of points in α in L can be merged, since the α of this example is a system formed from two oblique lines of points, there are no lines of points parallel to both of the two non-parallel lines of points. Thus no points can be merged.

Comment: For every regular expression α there are machines realizing the behavior $c(\alpha)$.

Every regular expression can be put in the quasi-normal form, and in the foregoing examples we have shown how to realize the behavior denoted by any of the constituent types of the quasi-normal form.

IV. A CLASS OF TAPE MACHINES WITH INPUT

The finite and infinite machines described in the preceding section can be represented in the form of certain tape-machines with input (TM). These machines have two major components: a finite state reading head, and an indefinitely expandible tape upon which are a finite number of "marked" squares. The reading head receives input symbols and upon receipt of such symbols, the reading head will undergo transitions, and ^{may} move a unit distance right (R) or left (L) on the tape. These moves can be given by state diagrams or by a finite table (see example below). Whenever the reading head is in an $s \in T$, and is resting on a marked tape square, the machine has an output. In all cases under discussion here, the reading head will be a finite state automaton which is also commutative. Thus the reading heads of these tape machines with input are precisely the finite devices discussed in the preceding section of this paper.

For any finite commutative machine there is an associated tape-machine with input realizing the same behavior.

This follows trivially by setting any finite commutative machine on the marked square of a tape so that the additional condition "when M is in an $s \in T$ and M is on a marked square of tape there is an output" is satisfied.

Theorem: For any commutative behavior denoted by an expression of the form $w_0 w_1^*$ (where w_0 may be Λ) there is a tape-machine which realizes this behavior. If w_1 is a mixed word, then an infinite machine will be required.

There is a close relationship between the reading head required to realize this behavior and a class of finite commutative machines.

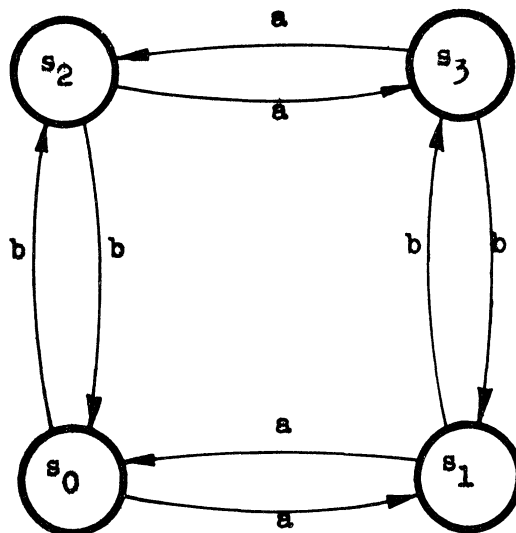
The transition structure of the reading head for a tape-machine with input which realizes the infinite behavior $c(a^k b^\ell)^*$ is identical to the transition structure for the finite c-machines realizing $c[(a^k)^*(b^\ell)^*]$.

To the transition structure so obtained must be appended moves (R,L) so that whenever an input sequence reaches a^k then R, and whenever an input sequence reached b^ℓ then L.

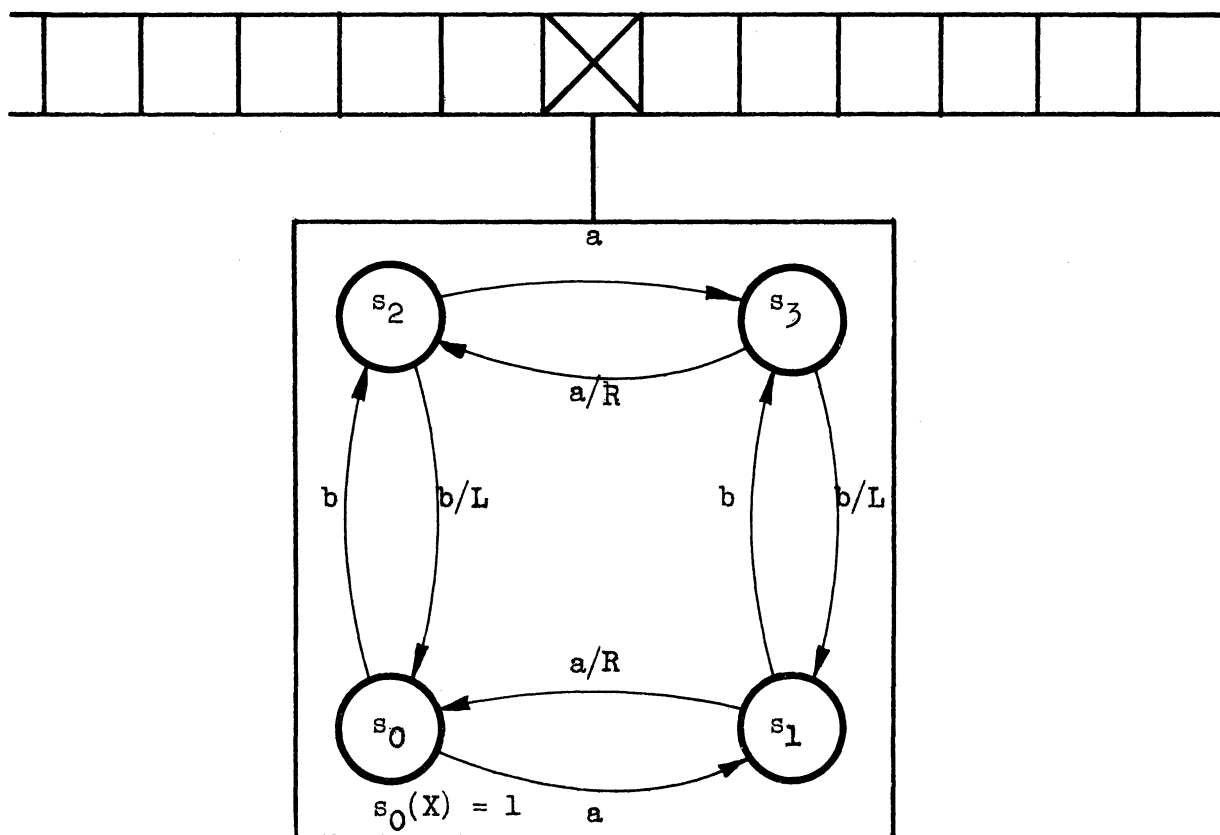
If, instead of unit moves of the tape, right or left, we allow moves of any fixed finite number of tape squares, then the minimal reading head will have a number of states equal to the index of the linear fundamental set denoting the behavior.

Example 17: $c(aabb)^*$ (See Example 11 for the realization of the infinite c-machine having this behavior.)

(1) The transition structure of the reading head is obtained by constructing a finite c-machine realizing $c(aa)^*(bb)^*$.



(2) To the above we append tape-move instructions R and L. We also add that the finite reading head is placed on the (single) marked square of a tape and is in s_0 ($s_0 \in T$) and is resting on the x-square, then there is an output ($s_0(x) = 1$). Thus the TM for $c(aabb)^*$ is:



The full description of the above tape machine with input may be presented in tabular form.

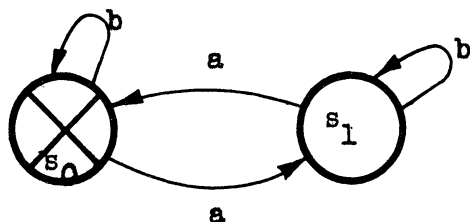
Present State	Input	Action	Next State
s_0	a	-	s_1
s_0	b	-	s_2
s_1	a	R	s_0
s_1	b	-	s_3
s_2	a	-	s_3
s_2	b	L	s_0
s_3	a	R	s_2
s_3	b	L	s_1

Output: $s_0(x) = 1$
Start: $s_0(x)$

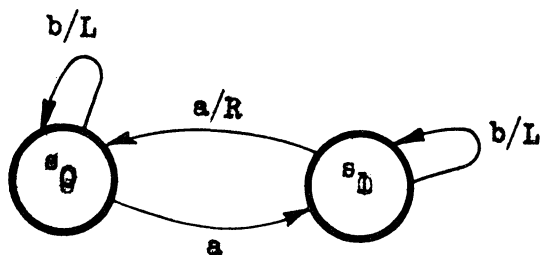
Note that if for example the above machine was started on tape square $x-1$ rather than on x ($M(x-1)$ instead of $M(x)$) the machine would realize the behavior $c(aa^{\sim}(aabb)^*)$ rather than $c(aabb)^*$. Feeder systems to the reading head may also be employed.

If we relax the restriction that only a finite number of squares of tape be marked, but require that the infinite markings be periodic, we can readily realize additional behaviors. If it is desired to realize a behavior such as $c((aa)^*(aabb)^*)$ then every tape square to the right of the start x square should also be marked x . If it is desired to realize a behavior such as $c((aaaa)^*(aabb)^*)$, then alternate tape squares to the right of the start should be marked.

Example 18: Some behaviors may not be realizable employing minimal state reading heads. Take for example $c((aaa)^*(aab)^*)$. $c((aa)^*b^*)$ is the behavior of the reading head, and this behavior can be realized by:



so that $c(aab)^*$ can be realized by a tape machine with reading head,



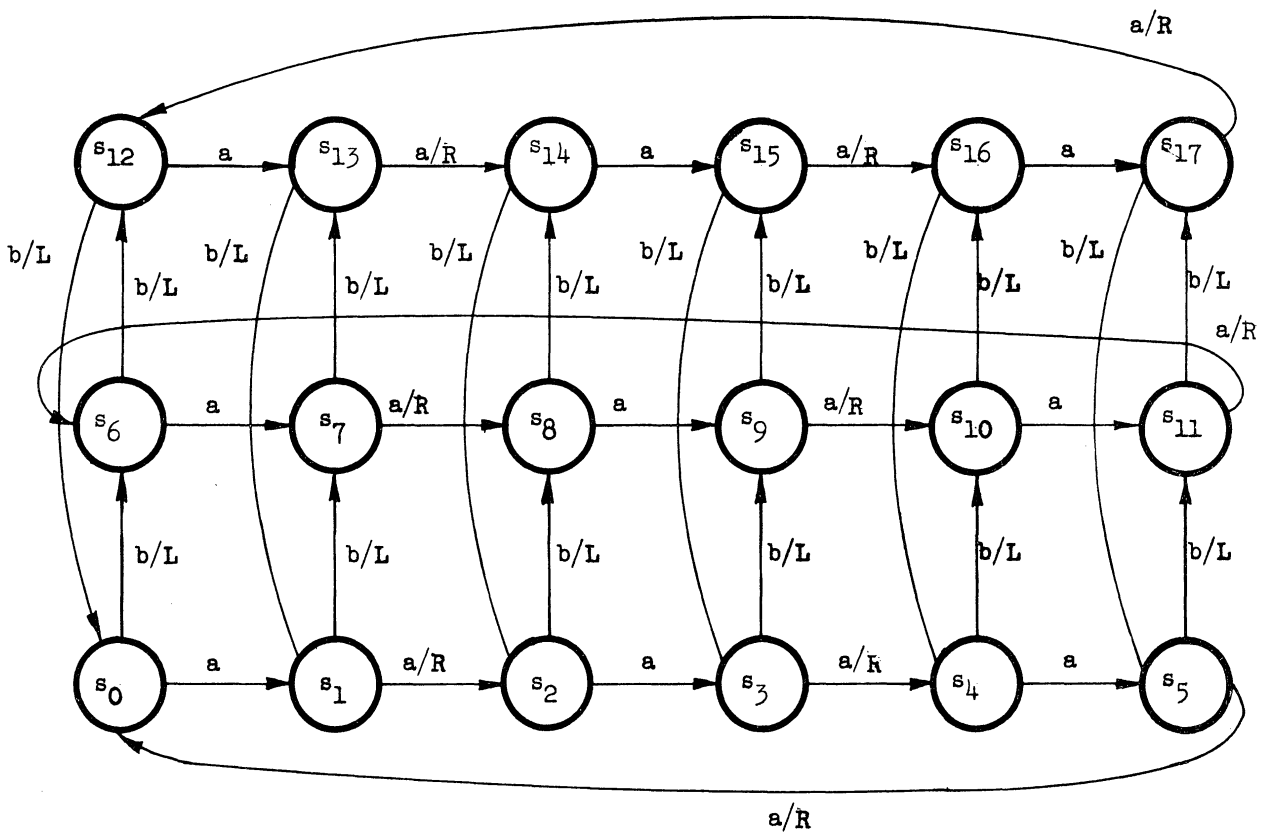
where the tape start square is x , and where $s_0(x) = 1$.

$c(aa)^*(aab)^*$ for example can be realized by marking all squares to the right of the start-square with x .

$c(a^*(aab)^*)$ can be realized by (1) marking all squares to the right of the start-square with x , and by making both $s_0(x) = 1$ and $s_1(x) = 1$.

$c((aaa)^*(aab)^*)$ cannot be realized employing this (2-state) minimal $c(aa)^*b^*$ reading head however. For this reading head has an insufficient number of internal states to distinguish aaa from a .

$c((aaa)^*(aab)^*)$ can be realized by the following tape machine with an 18-state reading head. The 18-state reading head is employed in order to obtain a common cycle with $(aaa)^*$ and is obtained by the technique of the Fundamental Identity Lemma of Section III.



$$\begin{aligned} s_0(x) &= 1 \\ s_3(x) &= 1 \\ s_8(x) &= 1 \end{aligned}$$

$$\begin{aligned} s_{11}(x) &= 1 \\ s_{13}(x) &= 1 \\ s_{16}(x) &= 1 \end{aligned}$$

The above machine is to start on the x square, and all squares to the right of this are also to be marked x.

Printing Machines

The use of a tape with a potentially infinite number of pre-marked squares may be found objectionable since under relaxed conditions this feature may permit the realizing of non-effective behaviors. Since, however, in the cases under discussion, the squares are to be marked in a periodic fashion, the marking procedures could be expressed as the behaviors of rather simple automata, viz. input-free finite automata. These automata would move the tape, (or move relative to the tape) and periodically print an x. Once the print machine had begun its action the reading head of the commutative tape machine proper could be set on the appropriate square ready to accept inputs.

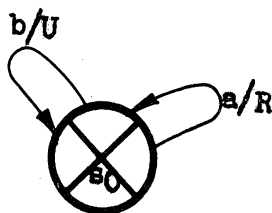
Both the move-print and the move-read actions can be combined in a finite automaton reading head.

Example 19: $cc((a^*)(ab)^*) :$

Present Head State	Tape State	Input	Action 1	Action 2	Next Head State
s_0	X	a	R	Px	s_0
s_0	X	b	L	-	s_0
s_0	-	a	R	-	s_0
s_0	-	b	L	-	s_0
Output: $s_0(x) = 1$					
Start: $s_0(x)$					

Upon receipt of a this machine will, when in s_0 and viewing an x , move right one unit and then print on x , (Px) .

The tape-machine concept can be further modified to produce a "machine" for realizing the behavior expressed by an oblique planar array of points in L , (e.g. $(aabb)^*(aab)^*$). The lattice-point model is viewed as an infinite two-dimensional array of tape squares. With each of the points of L denoting the behavior to be realized is associated an x -marked square of the array of tape. Since the behavioral properties of the systems are being carried in the array, the reading head can be of the simplest sort, viz. a one-state device which under receipt of an a moves right, and under receipt of a b moves up, and which is "on" whenever it is resting on a marked square.



V. COMMUTATIVE MACHINES AND THEIR TRANSITION ALGEBRAS

We now characterize in a more precise fashion the machines under discussion.

Lemma: The structure of $M = \langle S, s_0, f_a, f_b, T \rangle$ uniquely determines a semi-group operation (denoted by \circ) on S , and such that given

$$w_1 \approx s_1, w_2 \approx s_2 \text{ (for all } w_1, w_2 \in W, s_1, s_2 \in S) \text{ then}$$

$$s_1 \circ s_2 \approx w_1 w_2$$

Definition: A machine M is strongly connected if for every pair of states s, s' , there is some word w , such that $s \xrightarrow{w} s'$.

Definition: M is homogeneous if every machine obtained by choosing each $s \in S$ as initial state instead of s_0 , is isomorphic to M .

The isomorphism intended here is structural and does not take final states into account.

Definition: A machine M' will be said to be a sub-machine of M , if the states of M' are states of M , and the transitions of M' are the same for M but the initial state of M is not required to be a state of M' (contrary to conventional usage).

Definition: A machine is backwards-deterministic if its transition functions form a group.

"The Logic of Automata", A. W. Burks and H. Wang, JACM, 4, 2, p. 286.

Lemma: The following conditions (under the commutativity assumption) are equivalent:

- (1) M is strongly connected.
- (2) M is homogeneous.
- (3) The semigroup of M is a group.
- (4) M is backwards-deterministic.

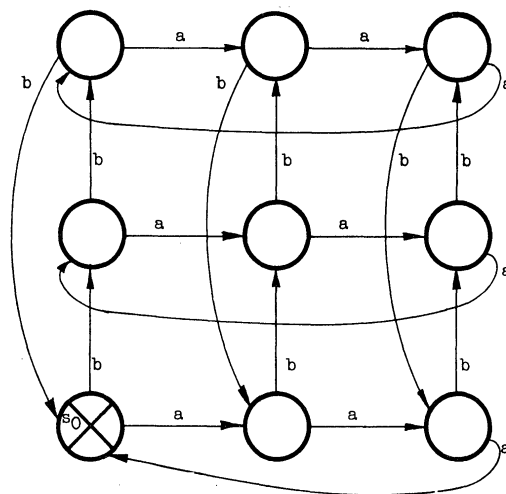
Lemma: Every (finite) commutative machine M has a unique homogeneous sub-machine.

That is, every M consists of a "feeder" system of states (possibly absent) and a system of states which is unique and homogeneous. (For explicit use of feeder systems to homogeneous machines see especially Examples 8 and 14 of Section IV. Note also that the homogeneous sub-machine need not contain a final state; see Example 3, where the "limbo" state forms the homogeneous sub-machine.)

Definition: A state s of a machine is a feeder state if there is no mixed word w such that $s \xrightarrow{w} s$.

Definition: A state s of a machine is a homogeneous state if there is a mixed word w such that $s \xrightarrow{w} s$.

It is of course possible that a fixed, finite number of a's alone or b's alone will, for an s , return M to the same s . This is typically the case in the sort of homogeneous finite machine shown below, which (for $s_0 \in T$) realizes the behavior $c((aaa)^*(bbb)^*)$



In this example both the a-cycle and the b-cycle are of finite order.

Homogeneous Machines

A homogeneous machine M is characterized by there being a mixed word w , such that $s_0 \xrightarrow{w} s_0$ (that is, such that the initial state of M is a homogeneous state).

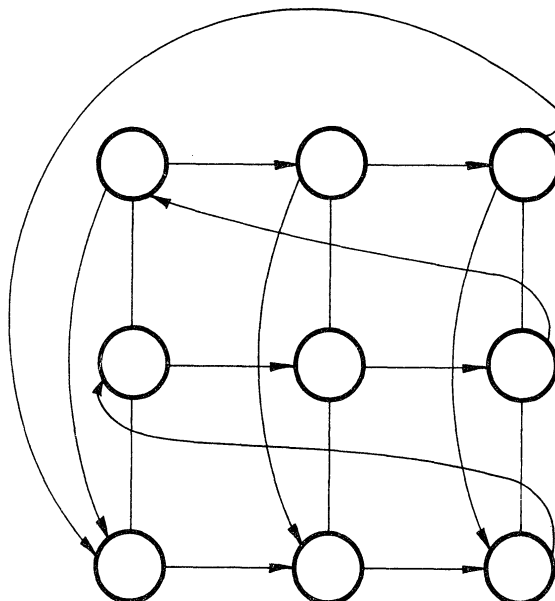
When M is finite and homogeneous there are positive integers i and j such that $s_0 \xrightarrow{a^i} s_0$ and $s_0 \xrightarrow{b^j} s_0$. Thus, there is a pure word (other than Λ) which is an "identity" word.

When M is infinite and homogeneous then there is a mixed word w such that $s_0 \xrightarrow{w} s_0$, (that is, there is an identity word which is a mixed word) and there is no pure word which is an identity word. The points of L associated with the identity words of an infinite homogeneous machine are collinear.

- Observations:
- (1) Every finite homogeneous machine can be embedded in torus.
 - (2) Every infinite homogeneous machine can be embedded in an infinite cylinder.

The finite homogeneous machine (realizing $c((aaa)^*(bbb)^*)$) of the last figure can be embedded in a torus. Note also the "variants" produced by "skewing," which, finite and homogeneous, can also be embedded in toruses.

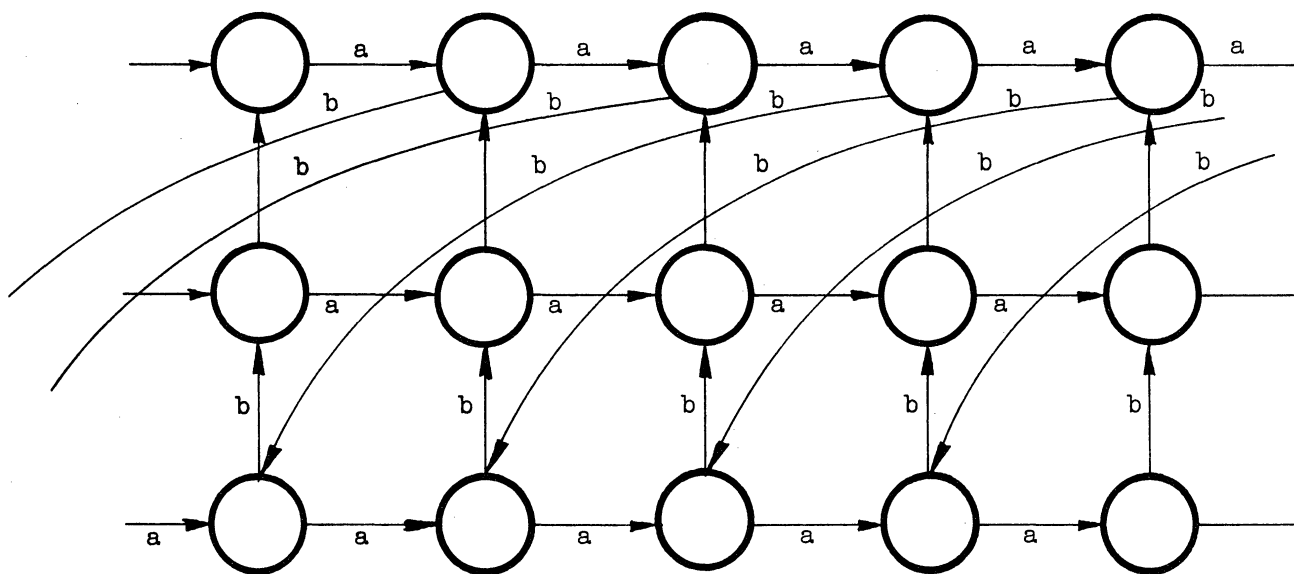
For example:



With any one state designated as both initial and final state, this system can realize the behavior $c((\Lambda \cup aaabb \cup aaaaaab) \overline{(aaaaaaaa)}^*(bbb)^*)$.

The infinite homogeneous machines which are embedable in infinite cylinders can conveniently be set down in an array in which a-transitions take the machine to states to the right, and b-transitions take the machine to states vertically above, until the top row is reached, at which point the next b-transition takes the machine to a state in the bottom row of the array. Repeated b-transitions will always take the machine to states in the bottom row which are to the left of previous states in the bottom row of the array.

For example, $c(aaabbb)^*$ (where because of the homogeneity property any state can be chosen as initial and final state);



Thus the machine under repeated a-transition goes to states to the right, and under repeated b-transitions spirals "backward" around the cylinder to the left. Note that if the b-returns from the top row of states do not go to the left then the infinite machine ceases to be strongly connected, and becomes instead a species of counter.

APPENDIX

Many of the results given in terms of an alphabet of two letters can be extended to expressions on larger alphabets.

Lemma: The star-length of constituents of a quasi-normal form can be made less than or equal to n , where n is the number of letters of the given alphabet.

This result was obtained by Y. Give'on.

Since the other properties employed in obtaining the quasi-normal form can be extended directly,

Theorem: There is a quasi-normal form for expressions on alphabets of n letters.

An association can be made between commutative regular expressions and the Presburger number theoretic logical language [4]. Let L_1, L_2, \dots, L_n be the lattice point models for languages of 2 letters, 3 letters, n letters, etc.

Theorem: For every regular expression E in n letters, there is a Presburger formula Q , having n free variables, which defines the same points of L_n .

For the case of expressions for the behavior of finite commutative automata, this was conjectured by M. P. Shützenberger, and was later demonstrated by C. C. Elgot and J. D. Rutledge [5].

There is a decision procedure for truth of sentences of the Presburger language. From this fact and the theorem above, the result of Section II that "for every two regular expressions E and E' it can be decided whether $c(E) = c(E')$ " follows directly.

Theorem: The set of regular expressions E such that $c(E)$ is regular is effectively decidable.

Given a regular expression E on two letters, there is a related Presburger formula $Q(x,y)$. An equivalence relation \approx on points is introduced:

$$(x_1, y_1) \approx (x_2, y_2) \iff \forall u, v [Q(x_1 + u, y_1 + v) \equiv Q(x_2 + u, y_2 + v)]$$

The above Presburger formula expresses the fact that x_1, y_1, x_2, y_2 are related by the induced congruence relation [6]. We now incorporate a finiteness condition for the number of congruence classes.

$$\exists z \forall x, y \exists x_1, y_1 [(x_1 < z, y_1 < z) \& ((x, y) \approx (x_1, y_1))]$$

This expression asserts that there exists a finite automaton having the behavior $c(E)$. By the Presburger decision procedure it can be determined whether such an expression as the above is true or false. Thus we have an effective procedure for deciding whether the event $c(E)$ is regular, for any regular expression E.

From the above two theorems it follows that,

Corollary: For any regular expression E, for which $c(E)$ is regular, we can effectively construct a finite commutative automaton whose behavior is $c(E)$.

REFERENCES

1. S. C. Kleene, "Representation of Events in Nerve Nets and Finite Automata," Automata Studies, (edited by C. E. Shannon and J. McCarthy), Princeton University Press, Princeton, New Jersey, (1956) 3-41.
2. Irving M. Copi, Calvin Elgot, and Jesse B. Wright, "Realization of Events by Logical Nets," J. Assoc. Comp. Mach., 5, (1958) 181-196.
3. E. F. Moore, "Gedanken-experiments on Sequential Machines," Automata Studies, (edited by C. E. Shannon and J. McCarthy), Princeton University Press, Princeton, New Jersey, (1956) 129-154.
4. M. Presburger, "Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt." Comptes-rendus du I Congres des Mathematiciens des Pays Slaves, Warsaw, (1930) 92-101 and 395.
5. C. C. Elgot and J. D. Rutledge, (Paper to appear on "Generalizations of the finite automaton model," Mathematical Sciences Department, International Business Machines Corporation, Yorktown Heights, New York).
6. J. R. Büchi, "Mathematical Theory of Automata: Notes for a Seminar" given by J. B. Wright and J. R. Büchi, Fall 1960, The University of Michigan

* * * *

DISTRIBUTION LIST

(One copy unless otherwise noted)

Assistant Secretary of Defense for Research and Engineering	2	Bureau of Ships Department of the Navy Washington 25, D. C. Attn: Code 607A, NTDS
Information Office Library Branch Pentagon Building Washington 25, D.C.		
Armed Services Technical Information Agency	10	Bureau of Naval Weapons Department of the Navy Washington 25, D.C. Attn: RAAV, Avionics Division
Arlington Hall Station Arlington 12, Virginia		
Chief of Naval Research	2	Bureau of Ships Department of the Navy Washington 25, D.C. Attn: Communications Br., Code 686
Department of the Navy Washington 25, D.C. Attn: Code 437, Infor. Syst. Br.		
National Security Agency Fort George G. Meade, Maryland Attn: R-4, Howard Campaigne		Naval Ordnance Laboratory White Oaks Silver Spring 19, Maryland Attn: Technical Library
Director, Naval Research Labs. Tech. Infor. Officer, Code 2000 Washington 25, D.C.	6	Massachusetts Institute of Technology Research Laboratory of Electronics Cambridge, Massachusetts Attn: Professor W. McCulloch
Commanding Officer Office of Naval Research Navy No. 100, Fleet Post Office New York, New York	10	David Taylor Model Basin Washington 7, D.C. Attn: Technical Library
Commanding Officer, ONR Br. Office 346 Broadway New York 13, New York		Naval Electronics Laboratory San Diego 52, California Attn: Technical Library
Commanding Officer, ONR Br. Office 495 Summer Street Boston 10, Massachusetts		University of Illinois Control Systems Laboratory Urbana, Illinois Attn: D. Alpert
Chief of Naval Operations OP-07T-12 Navy Department Washington 25, D.C.		University of Illinois Digital Computer Laboratory Urbana, Illinois Attn: Dr. J. E. Robertson

Technical Information Officer
U. S. Army Signal R&D Laboratory
Fort Monmouth, New Jersey
Attn: Data Equipment Branch

U. S. Naval Weapons Laboratory
Dahlgren, Virginia
Attn: Head, Computation Division
G. H. Gleissner

George Washington University
Washington, D.C.
Attn: Prof. N. Grisamore

Aberdeen Proving Ground, BRL
Aberdeen Proving Ground, Maryland
Attn: J. H. Giese, Chief Comput. Lab.

Office of Naval Research
Resident Representative
The University of Michigan
146-B Temporary Classroom Bldg.
Ann Arbor, Michigan

Commanding Officer
ONR, Branch Office
John Crerar Library Bldg.
86 East Randolph Street
Chicago 1, Illinois

Commanding Officer
ONR, Branch Office
1030 E. Green Street
Pasadena, California

Commanding Officer
ONR, Branch Office
1000 Geary Street
San Francisco 9, California

National Bureau of Standards
Washington 25, D.C.
Attn: Mr. R. D. Elbourn

Naval Ordnance Laboratory
Corona, California
Attn: H. H. Weider

Massachusetts Inst. of Technology
Cambridge, Massachusetts
Attn: D. W. Baumann, Dynamic
Analysis and Control Lab.

Burroughs Corporation
Research Center
Paoli, Pennsylvania
Attn: R. A. Tracey

National Bureau of Standards
Data Processing Systems Division
Room 239, Bldg. 10, Attn: A. K. Smilow
Washington 25, D. C.

Lockheed Missiles and Space Company
3251 Hanover Street
Palo Alto, California
Attn: W. F. Main

The University of Michigan
Ann Arbor, Michigan
Attn: Professor A. W. Burks,
Dept. of Philosophy

Census Bureau
Washington 25, D.C.
Attn: Office of Assistant Director
for Statistical Services,
Mr. J. L. McPherson

National Science Foundation
Program Dir. for Documentation Res.
Washington 25, D.C.
Attn: Helen L. Brownson

University of California
Los Angeles 24, California
Attn: Department of Engineering,
Professor Gerald Estrin

Columbia University
New York 27, New York
Attn: Department of Physics,
Professor L. Brillouin

University of Illinois
Champaign Urbana, Illinois
Attn: John R. Pasta

Naval Research Laboratory
Washington 25, D.C.
Attn: Security Systems
Code 5266, Mr. G. Abraham

Diamond Ordnance Fuze Laboratory
Connecticut Ave. & Van Ness St.
Washington 25, D.C.
Attn: ORDTL-012, E. W. Channel

Harvard University
Cambridge, Massachusetts
Attn: School of Applied Science
Dean Harvey Brook

The University of Chicago
Institute for Computer Research
Chicago 37, Illinois
Attn: Mr. Nicholas C. Metropolis

Wright Air Development Division
Electronic Technology Laboratory
Wright-Patterson AFB, Ohio
Attn: Lt. Col. L.M. Butsch, Jr.
ASRUEB

Laboratory for Electronics, Inc.
1079 Commonwealth Ave.
Boston 15, Massachusetts
Attn: Dr. H. Fuller

Stanford Research Institute
Computer Laboratory
Menlo Park, California
Attn: H. D. Crane

General Electric Co.
Schenectady 5, New York
Attn: Library, L.M.E. Dept.

Hunter College
New York 21, New York
Attn: Dean Mina Rees

The RAND Corp.
1700 Main St.
Santa Monica, California
Attn: Numerical Analysis Dept.
Willis H. Ware

General Electric Research Laboratory
P. O. Box 1088
Schenectady, New York
Attn: Information Studies Section
R. L. Shuey, Manager

Mr. Sidney Kaplan
1814 Glen Park Ave.
Silver Spring, Maryland

University of Pennsylvania
Institute of Co-operative Research
Philadelphia, Pennsylvania
Attn: Dr. John O'Conner

Stanford Research Institute
Menlo Park, California
Attn: Dr. Charles Rosen
Applied Physics Laboratory

Northeastern University
360 Huntington Ave.
Boston, Massachusetts
Attn: Prof. L. O. Dolansky

New York University
New York, New York
Attn: Dr. J. H. Mulligan, Jr.
Chairman of E. E. Dept.

Marquardt Aircraft Co.
16555 Saticoy St.
P. O. Box 2013, South Annex
Van Nuys, California
Attn: Dr. Basun Chang
Research Scientist

Texas Technological College
Lubbock, Texas
Attn: Paul G. Griffith
Dept. of Elec. Eng.

Dr. Stanley Winkler
IBM Corporation
Federal Systems Division
326 E. Montgomery Ave.
Rockville, Maryland

Post Office Department
Office of Research and Engineering
12th and Pennsylvania Ave.
Washington 25, D.C.
Attn: Mr. R. Kopp
Res. & Dev. Division

L. G. Hanscom Field
AF-CRL-CRRB
Bedford, Massachusetts
Attn: Dr. H. H. Zschirnt

Department of the Army
Office of the Chief of Res. & Dev.
Pentagon, Room 3D442
Washington 25, D.C.
Attn: Mr. L. H. Geiger

Bell Telephone Laboratories
Murray Hill Laboratory
Murray Hill, New Jersey
Attn: Dr. Edward F. Moore

National Biomedical Res. Found., Inc.
8600 16th St., Suite 310
Silver Spring, Maryland
Attn: Dr. R. S. Ledley

University of Pennsylvania
Moore School of Elec. Eng.
200 South 33rd St.
Philadelphia 4, Pennsylvania
Attn: Miss Anna Louise Campion

Division of Automatial Data
Processing /AOP/
Department of State
Washington 25, D.C.
Attn: F. P. Diblasi, 19A16

Auerbach Electronics Corp.
1634 Arch St.
Philadelphia 3, Pennsylvania

University of Pennsylvania
Mechanical Languages Projects
Moore School of Elec. Eng.
Philadelphia 4, Pennsylvania
Attn: Dr. Saul Gorn, Director

Applied Physics Laboratory
Johns Hopkins University
8621 Georgia Ave.
Silver Spring, Maryland
Attn: Document Library

Bureau of Supplies and Accounts, Chief
Navy Department
Washington, D.C.
Attn: Code W3

National Aeronautics & Space Admin.
Goddard Space Flight Center
Greenbelt, Maryland
Attn: Chief, Data Systems Div.
C.V.L. Smith

Federal Aviation Agency
Bureau of Research & Development
Washington 25, D.C.
Attn: RD-375, Mr. Harry Hayman

Mr. Donald F. Wilson
Naval Research Laboratory
Washington 25, D.C.
Attn: Code 5144

David Taylor Model Basin
Washington 7, D.C.
Attn: Aerodynamics Laboratory, Code 628
Miss Cravens

Lincoln Laboratory
Mass. Institute of Technology
Lexington 73, Massachusetts
Attn: Library

Dr. C. R. Porter
Psychology Department
Howard University
Washington 1, D.C.

Electronics Research Laboratory
University of California
Berkeley 4, California
Attn: Director

Institute for Defense Analysis
Communications Research Division
Von Neumann Hall
Princeton, New Jersey

Hebrew University
Jerusalem, Israel
Attn: Prof. Y. Bar-Hillel

National Physical Laboratory
Teddington, Middlesex
England
Attn: Dr. A. M. Uttley, Supt.
Autonomics Division

