M. O. Rabin*
D. Scott†

Finite Automata and Their Decision Problems‡

Abstract: Finite automata are considered in this paper as instruments for classifying finite tapes. Each one-tape automaton defines a set of tapes, a two-tape automaton defines a set of pairs of tapes, et cetera. The structure of the defined sets is studied. Various generalizations of the notion of an automaton are introduced and their relation to the classical automata is determined. Some decision problems concerning automata are shown to be solvable by effective algorithms; others turn out to be unsolvable by algorithms.

Introduction

Turing machines are widely considered to be the abstract prototype of digital computers; workers in the field, however, have felt more and more that the notion of a Turing machine is too general to serve as an accurate model of actual computers. It is well known that even for simple calculations it is impossible to give an *a priori* upper bound on the amount of tape a Turing machine will need for any given computation. It is precisely this feature that renders Turing's concept unrealistic.

In the last few years the idea of a *finite automaton* has appeared in the literature. These are machines having only a finite number of internal states that can be used for memory and computation. The restriction of finiteness appears to give a better approximation to the idea of a physical machine. Of course, such machines cannot do as much as Turing machines, but the advantage of being able to compute an arbitrary general recursive function is questionable, since very few of these functions come up in practical applications.

Many equivalent forms of the idea of finite automata have been published. One of the first of these was the definition of "nerve-nets" given by McCulloch and Pitts.[3] The theory of nerve-nets has been developed by authors too numerous to mention. We have been particularly influenced, however, by the work of S. C. Kleene[2] who proved an important theorem characterizing the possible action of such devices (this is the notion of "regular event" in Kleene's terminology). J. R. Myhill, in some unpublished work, has given a new treatment of Kleene's results and this has been the actual point of departure for the investigations presented in this report. We have not, however, adopted Myhill's use of directed graphs as a method of viewing automata but have retained throughout a machine-like formalism that permits direct comparison with Turing machines. A neat form of the definition of automata has been used by Burks and Wang[1] and by E. F. Moore,[4] and our point of view is closer to theirs than it is to the formalism of nerve-nets. However, we have adopted an even simpler form of the definition by doing away with a complicated output function and having our machines simply give "yes" or "no" answers. This was also used by Myhill, but our generalizations to the "nondeterministic," "two-way," and "many-tape" machines seem to be new.

In Sections 1-6 the definition of the one-tape, one-way automaton is given and its theory fully developed. These machines are considered as "black boxes" having only a finite number of internal states and reacting to their environment in a deterministic fashion.

We center our discussions around the application of automata as devices for defining sets of tapes by giving "yes" or "no" answers to individual tapes fed into them. To each automaton there corresponds the set of those tapes "accepted" by the automaton; such sets will be referred to as *definable sets*. The structure of these sets of tapes, the various operations which we can perform on these sets, and the relationships between automata and defined sets are the broad topics of this paper.

After defining and explaining the basic notions we give, continuing work by Nerode,[5] Myhill, and Shepherdson,[7] an intrinsic mathematical characterization of definable sets. This characterization turns out to be a useful tool for both proving that certain sets are definable by an automaton and for proving that certain other sets are not.

In Section 4 we discuss decision problems concerning automata. We consider the three problems of deciding whether an automaton accepts any tapes, whether it ac-

*Now at the Department of Mathematics, Hebrew University in Jerusalem.

†Now at the Department of Mathematics, University of Chicago.

‡The bulk of this work was done while the authors were associated with the IBM Research Center during the summer of 1957.

cepts an infinite number of different tapes, and whether two automata accept precisely the same tapes. All three problems are shown to be solvable by effective algorithms.

In Chapter II we consider possible generalizations of the notion of an automaton. A *nondeterministic automaton* has, at each stage of its operation, several choices of possible actions. This versatility enables us to construct very powerful automata using only a small number of internal states. Nondeterministic automata, however, turn out to be equivalent to the usual automata. This fact is utilized for showing quickly that certain sets are definable by automata.

Using nondeterministic automata, a previously given construction of the direct product of automata (Definition 7), and the mathematical characterization of definable sets, we give short proofs for various well-known closure properties of the class of definable sets (e.g., the definable sets form a Boolean algebra). Furthermore we include, for the sake of completeness, a formulation of Kleene's theorem about regular events.

In trying to define automata which are closer to the ideal of the Turing machine, while preserving the important feature of using only a preassigned amount of tape, another generalization suggests itself. We relax the condition that the automaton always move in one direction and allow the machine to travel back and forth. In this way we arrive at the idea of a *two-way* automaton. In Section 7 we consider the problem of comparing one-way with two-way automata, a study that can be construed as an investigation into the nature of memory of finite automata. A one-way machine can be imagined as having simply a keyboard representing the symbols of the alphabet and as having the sequence from the tape fed in by successively punching the keys. Thus no permanent record of the tape is required for the operation of the machine. A two-way automaton, on the other hand, does need a permanent, actual tape on which it can run back and forth in trying to compute the answer. Surprisingly enough, it turns out that despite the ability of backwards reference, two-way automata are no more powerful than one-way automata. In terms of machine memory this means that all information relevant to a computation which an automaton can gather by backward reference can always be handled by a finite memory in a one-way machine.

In Chapter III we study multitape machines. These automata can read symbols on several different tapes, and we adopt the convention that a machine will read for a while on one tape, then change control and read on another tape, and so on. Thus, with a two-tape machine, a set of pairs of tapes is defined, or we can say a binary relation between tapes is defined. Using again the powerful tool of nondeterministic automata, we establish a relationship between two-tape automata and one-tape automata. Namely, the domain and range of a relation defined by a two-tape automaton are sets of tapes definable by one-tape automata. From this follows the fact that, unlike the sets definable by one-tape automata, the relations definable by two-tape automata do not form a Boolean algebra. The problems whether a two-tape automaton accepts any pair of tapes and whether it accepts an infinite number of pairs are shown to be solvable by effective algorithms.

We conclude with a brief discussion of two-way, two-tape automata. Here even the problem whether an automaton accepts any tapes at all is not solvable by an effective algorithm. Furthermore a reduction of two-way automata to one-way automata is not possible. All in all, there is a marked difference between the properties of one-tape automata and those of two-tape automata. The study of the latter is yet far from completion.

Chapter I. One-tape, one-way automata

• *1. The intuitive model and basic definitions*

An automaton will be considered as a black box of which questions can be asked and from which a "yes" or "no" answer is obtained. The number of questions that can be asked will be infinite, and for simplicity a question is interpreted as any arbitrary finite sequence of symbols from a finite alphabet given in advance. An easy way to imagine the act of asking the question of the automaton is to think of the black box as having the separate symbols on a typewriter keyboard. Then the machine is turned on and the question is typed in; after an "end of question" button is pressed, a light indicates a "yes" or "no" answer. Other good images of how the automaton could appear physically would use punched cards. Suppose that we punch just one symbol or code number for a symbol to a card; then a question is simply a stack of cards. The automaton is asked a question by having the stack read in a card at a time in the usual way.

For the purposes of this paper, we shall not use either of the above images but rather think of the questions as given on one-dimensional tapes. The machine will be endowed with a reading head which can read one square of the tape (i.e., one symbol) at a time, and then it can advance the tape one unit and read, say, the next square to the right. We assume the machine stops when it runs out of tape. So much for the external character of an automaton.

The internal workings of an automaton will not be analyzed too deeply. We are not concerned with how the machine is built but with what it can do. The definition of the internal structure must be general enough to cover all conceivable machines, but it need not involve itself with problems of circuitry. The simple method of obtaining generality without unnecessary detail is to use the concept of *internal states*. No matter how many wires or tubes or relays the machine contains, its operation is determined by stable states of the machine at discrete time intervals. An actual existing machine may have billions of such internal states, but the number is not important from the theoretical standpoint—only the fact that it is finite.

As a further simplifying device, we need not consider all the intermediate states that the machine passes through but only those directly preceding the reading of a symbol. That is, the machine first reads a symbol or square on the tape, then it may pass through several states before it is ready to read the next symbol. To be able to mimic the action of the automaton, we need not remember all these intermediate states but only the last one it goes into before it reads the next square. In fact, if we make a table of all the transitions from a state and a symbol to a new state, then the whole action of the machine is essentially described.

Finally, to get the answer from the machine, we need only distinguish between those states in which the "yes" light is on and those states in which the "no" light is on when the end of the question is reached. Again, for simplicity, it is assumed that all states are in one category or the other but not in both. Thus the whole machine is described when a class of designated states corresponding to the "yes" answers is given. It remains now to give a precise mathematical form to these ideas.

First a finite alphabet Σ is given and fixed for the rest of the discussion. The actual number of symbols in the alphabet is not important. It is only important that all the automata considered use the same alphabet so that different machines can be compared. For illustration we shall often think of Σ as containing only the two symbols 0 and 1. By a *tape* we shall understand any finite sequence of symbols from Σ. We also include the empty tape with no symbols to be denoted by Λ. The class of all tapes is denoted by T. If x and y are tapes in T, then xy denotes the tape obtained by splicing x and y together or by juxtaposing or concatenating the two sequences. In other words, if

$$x=\sigma_0\sigma_1 \ldots \sigma_{n-1}$$

and

$$y=\tau_0\tau_1 \ldots \tau_{n-1},$$

then

$$xy=\sigma_0\sigma_1 \ldots \sigma_{n-1}\tau_0\tau_1 \ldots \tau_{n-1},$$

where the σ's and τ's are in Σ. We assume as obvious the two laws

$$\Lambda x = x\Lambda = x,$$

and

$$x(yz)=(xy)z,$$

for all x, y, z in T. In mathematical terminology, T together with the operation of juxtaposition forms the *free semigroup* (with unit) generated by Σ.

We shall often have occasion to cut tapes into pieces. For example, let

$$x=\sigma_0\sigma_1 \ldots \sigma_{n-1};$$

the σ's are in Σ and n is referred to as the length of the tape x. We adopt the following notation

$${}_kx_l=\sigma_k\sigma_{k+1} \ldots \sigma_{l-1},$$

where $k \leq l \leq n$. In other words ${}_kx_l$ is a section of x running from the $(k+1)^{\text{st}}$ symbol of x through the l^{th} symbol. Clearly, the length of ${}_kx_l$ is $l-k$. We will agree that if $k=l$, then ${}_kx_l=\Lambda$, the tape of length 0. As a useful property of the notation, we have

$$x={}_0x_k\ {}_kx_n,$$

where $k \leq n$, or more generally

$${}_kx_m={}_kx_l\ {}_lx_m,$$

where $k \leq l \leq m \leq n$.

We shall refer to such tapes as ${}_0x_k$ as the *initial section* or *initial portion* of x of length k.

The obvious notation x^n for $xxx \ldots x$ multiplied together n times will also be used with the convention that $x^0=\Lambda$.

Having explained all the notations for the tapes that will be fed into the machines, we turn now to the formal definition of an automaton.

Definition 1. *A* (finite) automaton *over the alphabet Σ is a system $\mathfrak{A}=(S,M,s_0,F)$, where S is a finite non-empty set* (*the* internal states *of* $\mathfrak{A}$), *M is a function defined on the Cartesian product $S \times \Sigma$ of all pairs of states and symbols with values in S* (*the table of* transitions *or* moves *of* $\mathfrak{A}$), *s_0 is an element of S* (*the* initial state *of* $\mathfrak{A}$), *and F is a subset of S* (*the* designated final states *of* $\mathfrak{A}$).

Let $\mathfrak{A}$ be an automaton. First of all the function M can be extended from $S \times \Sigma$ to $S \times T$ in a very natural way by a definition by recursion as follows:

$M(s,\Lambda)=s$, for s in S;

$M(s,x\sigma)=M(M(s,x),\sigma)$, for s in S, x in T, and σ in Σ.

The meaning of $M(s,x)$ is very simple: *it is that state of the machine obtained by beginning in state s and reading through the whole tape x symbol by symbol, changing states according to the given table of moves.* It should be at once apparent from the definition of the extension of M just given that we have the following useful property:

$M(s,xy)=M(M(s,x),y)$, for all s in S and x,y in T.

We may now easily define the set of those tapes which cause the automaton to give a "yes" answer.

Definition 2. *The set of tapes* accepted *or* defined *by the automaton $\mathfrak{A}$, in symbols $T(\mathfrak{A})$, is the collection of all tapes x in T such that $M(s_0,x)$ is in F.*

Definition 3. *The class of all* definable sets of tapes, *in symbols $\mathcal{T}$, is the collection of all sets of the form $T(\mathfrak{A})$ for some automaton $\mathfrak{A}$.*

The meaning of acceptance can be made clearer by a diagram. Let $x=\sigma_0 \ldots \sigma_{n-1}$. For each $k \leq n$, let

$$s_k=M(s_0,{}_0x_k),$$

so that for $k>0$ we have

$$s_k=M(s_{k-1},\sigma_{k-1}).$$

The condition that x be in $T(\mathfrak{A})$ is that s_n be in F. Each s_k is the state of the machine $\mathfrak{A}$ after reaching the

k^{th} symbol in the tape x. Thus if we write down the following diagram:

$$\begin{array}{ccccccccccc} & \sigma_0 & & \sigma_1 & & \sigma_2 & & \cdots & & \sigma_{n-1} & \\ s_0 & & s_1 & & s_2 & & s_3 \cdots s_{n-1} & & & & s_n, \end{array}$$

we have a complete picture of the motion of the machine $\mathfrak{A}$ across the tape x. It is very important to notice in this picture that there is exactly one more internal state than there are symbols on the tape, a fact that will be used several times in Section 4.

• 2. *A mathematical characterization of definable sets*

An automaton can be a very complicated object, and it is not clear exactly how complicated the sets definable by automata can become. In order to understand the nature of these definable sets, we will develop in this section a mathematically simple and completely intrinsic characterization of these sets, which shows exactly the effect of considering machines with only a finite number of internal states. This "finiteness" condition is certainly the main feature of our study.

Actually two different characterizations will be given, but they share a common feature of involving equivalence relations over the set T of all tapes. The reader is assumed familiar with the notion of an equivalence relation and equivalence classes.

Definition 4. *An equivalence relation R over the set T of tapes is* right invariant *if whenever xRy, then $xzRyz$ for all z in T.*

Clearly there is an analogous definition of *left-invariant* equivalence relations.

Definition 5. *An equivalence relation over the set T is a* congruence relation *if it is both right and left invariant.*

If R is a congruence relation then the formulas xRz and yRw always imply $xyRzw$. In consequence, if $[x]$ is the equivalence class containing x, and $[y]$ is the equivalence class containing y, then we can define unambiguously the product of the two equivalence classes by the equation

$$[x][y]=[xy].$$

In mathematical terms, the set of equivalence classes is said to be the *quotient semigroup* of T under the congruence relation R and is called a *homomorphic image* of T. There are many distinct homomorphic images of T, but we shall be most interested in those that are finite. Somewhat more generally we shall make use of equivalence relations satisfying the following definition.

Definition 6. *An equivalence relation over T is of* finite index *if there are only finitely many equivalence classes under the relation.*

With these definitions, we may now state the first result on characterizing definable sets. This theorem is due to J. R. Myhill and is published with his kind permission.

Theorem 1. (Myhill) *Let U be a set of tapes. The following three conditions are equivalent:*

(i) *U is in $\mathcal{T}$;*

(ii) *U is the union of some of the equivalence classes of a congruence relation over T of finite index;*

(iii) *the explicit congruence relation $\equiv$ defined by the condition that for all x,y in T, $x\equiv y$ if and only if for all z,w in T, whenever zxw is in U, then zyw is in U, and conversely, is a congruence relation of finite index.*

Proof: Assume (i) and in particular that $U=T(\mathfrak{A})$ for a suitable automaton $\mathfrak{A}$. Define a relation R by the condition that xRy if and only if $M(s,x)=M(s,y)$ for all s in S. Clearly R is an equivalence relation, but it is also a congruence relation. For assume that xRy and z is any tape in T. Then

$$\begin{aligned} M(s,xz) &= M(M(s,x),z) \\ &= M(M(s,y),z) \\ &= M(s,yz), \qquad \text{for all } s \text{ in } S. \end{aligned}$$

Thus R is right invariant. Likewise

$$\begin{aligned} M(s,zx) &= M(M(s,z),x) \\ &= M(M(s,z),y) \\ &= M(s,zy), \qquad \text{for all } s \text{ in } S, \end{aligned}$$

and R is shown to be left invariant.

That R is of finite index is a consequence of the fact that if x is a fixed tape and r is the number of internal states of $\mathfrak{A}$, then the expression $M(s,x)$ can assume at most r different values. Thus the number of equivalence classes is at most r^r.

Finally if x is in $T(\mathfrak{A})$ and xRy, then $M(s_0,x)=M(s_0,y)$ so that y is in $T(\mathfrak{A})$ also. This remark shows that $U=T(\mathfrak{A})$ is in fact the union of the equivalence class under R of those tapes in U. We have thus shown that (i) implies (ii).

Assume next that statement (ii) holds, and let R now stand for any congruence relation satisfying the conditions mentioned in (ii). Consider the specific relation $\equiv$ defined in (iii) in terms of U. Let x and y be any tapes such that xRy. Suppose that zxw is in U. Now R is a congruence relation, so that $zxwRzyw$. On the other hand U is a union of equivalence classes. Thus zyw must also be in U. This argument actually shows that if xRy, then $x\equiv y$. In other words, $\equiv$ is a relation making fewer distinctions than the relation R. That $\equiv$ is a congruence relation is a trivial consequence of its definition, so if R is of finite index, then $\equiv$ must necessarily be of finite index too. Hence, (ii) implies (iii).

Finally, assume that (iii) holds. We must define an automaton $\mathfrak{A}$ such that $U=T(\mathfrak{A})$. To this end, let S be the set of equivalence classes under the congruence relation $\equiv$. Define the function M by the formula:

$$M([x],\sigma)=[x\sigma],$$

where the square brackets indicate the formation of equivalence classes. Notice we need only the fact that $\equiv$ is right invariant to see that the definition of M is unambiguous. Further, let $s_0=[\Lambda]$, and finally let F be the set of all $[x]$ where x is in U. It should be obvious that U is indeed a union of equivalence classes under $\equiv$. A simple inductive argument shows that if M is extended in the way indicated in Section 1 to the set $S\times T$, then $M([x],y)=[xy]$ for all x,y in T. Thus we see at once that $M(s_0,x)=M([\Lambda],x)=[x]$ is in F if and only if x

is in U; in other words $U=T(\mathfrak{A})$, as was to be shown. Hence, (iii) implies (i), and the proof of Theorem 1 is complete.

The main trouble with Theorem 1 is that the number of equivalence classes under the relation $\equiv$ can become very large as is indicated in the proof that (i) implies (ii). To be more economical and to stay closer to the simpler automata defining the set U, one should use only right-invariant equivalence relations rather than demanding congruence relations. The following theorem is formulated in an exactly parallel fashion to Theorem 1 and is essentially a simplification of a theorem by A. Nerode,[5] who used a somewhat more involved notion of automaton than that adopted here. The principle is very useful and was employed by J. C. Shepherdson[7] in a proof of the main theorem of Section 7, as is explained there.

Theorem 2. (Nerode) *Let U be a set of tapes. The following three conditions are equivalent:*

(i) *U is in $\mathcal{T}$;*

(ii) *U is the union of some of the equivalence classes of a right-invariant equivalence relation over T of finite index;*

(iii) *the explicit right-invariant equivalence relation E defined by the condition that for all x,y in T, xEy if and only if for all z in T, whenever xz is in U, then yz is in U, and conversely, is an equivalence relation of finite index.*

The proof need not be given in detail because it can be copied almost word for word from the proof of Theorem 1. It should only be mentioned that the relation R in the proof that (i) implies (ii) has the simpler definition:

xRy *if and only if* $M(s_0,x)=M(s_0,y)$.

This implies that the number of equivalence classes for R is at most the number of internal states of $\mathfrak{A}$. This remark and an analysis of the full proof leads directly to the following corollary.

Corollary 2.1. *If U is in $\mathcal{T}$, then the number of equivalence classes under the relation E is the least number of internal states of any automaton defining U.*

In other words, the relation E leads at once to the most economical automaton defining U. This remark is is also due to Nerode.

As a simple application of Theorem 1, we shall show that the set U of all tapes of the form 0^n10^n for $n=0,1,2,\ldots$ is not definable by any automaton. Suppose to the contrary that U is in $\mathcal{T}$. Consider the relation $\equiv$ of Theorem 1 (iii). This relation would have to be of finite index, so that for some integers $n\neq m$ we would have $0^n\equiv 0^m$. It follows at once that $0^n10^m\equiv 0^n10^n$, and hence that 0^n10^m is in U, which is impossible. Thus U cannot be in $\mathcal{T}$.

• 3. *Closure properties of the class of definable sets*

Using the theorems just given in the preceding section, we can derive very simply some facts about the class $\mathcal{T}$. It turns out that $\mathcal{T}$ can be actually characterized by its closure properties under some natural operations on sets of tapes, but the discussion of this fact will be delayed to Section 6. Sometimes it is easier to use Theorems 1 and 2 and sometimes it is easier to give direct constructions of machines. In this section we shall indicate how the Boolean operations can be done in both ways. First, however, we prove two theorems that seem to be easier by the indirect method.

Theorem 3. *If x is in T, then $\{x\}$, the set consisting only of x, is in $\mathcal{T}$.*

Proof: Clearly an automaton can be built which recognizes one and only one tape given in advance; however, Theorem 2 is easier to apply. The relation E defined in Theorem 2 (iii) in terms of $U=\{x\}$ simply means that yEz if and only if whenever y and z are initial segments of the tape x, then $y=z$. Thus E has one equivalence class for each initial segment of x and one extra equivalence class for all the rest of the tapes. Obviously E then is of finite index, which completes the proof.

If x is any tape, then it can be turned end-for-end and written backwards. Let x^* stand for the result of writing x backwards so that if $x=\sigma_0\sigma_1\ldots\sigma_{n-1}$, then $x^*=\sigma_{n-1}\sigma_{n-2}\ldots\sigma_0$. Clearly we have the rules:

$\sigma^*=\sigma$, for σ in Σ,

$\Lambda^*=\Lambda$,

$x^{**}=x$,

and

$(xy)^*=y^*x^*$.

In case U is any set of tapes, U^* will denote the set of all x^* where x is in U.

The motion of an automaton, according to the definitions of Section 1, is always from left to right. Thus from the original definition, the following result is a little surprising.

Theorem 4. *If U is in $\mathcal{T}$, then U^* is in $\mathcal{T}$.*

Proof: The content of the theorem is that if a set of tapes is definable, then so is the set obtained by writing all the defined tapes backwards. The direct construction of a machine defining U^* from a given machine defining U is rather lengthy, but Theorem 1 makes the result almost obvious. Let $\equiv$ be the relation defined in terms U from Theorem 1 (iii) and let $\equiv^*$ be the analogous relation for U^*. Assume that $x\equiv^*y$. If zx^*w is in U, then $(zx^*w)^*$ is in U^*. But $(zx^*w)^*=w^*xz^*$. Hence, w^*yz^* is in U^* also; however, $w^*yz^*=(zy^*w)^*$, and so zy^*w is in U. This shows that $x^*\equiv y^*$. Since $U^{**}=U$, this argument with U and U^* interchanged is also valid, and we have proved that $x\equiv^*y$ if and only if $x^*\equiv y^*$, for all x,y in T. Clearly then, if $\equiv$ is of finite index, then $\equiv^*$ must be also of finite index with the same number of equivalence classes, which completes the proof.

Theorem 5. *The class $\mathcal{T}$ is a Boolean algebra of sets.*

Proof: That the class $\mathcal{T}$ is closed under complements is the most obvious fact, even from the original definition. For if $U=T(\mathfrak{A})$ where $\mathfrak{A}=(S,M,s_0,F)$, then $T-U=T(\mathfrak{B})$, where $\mathfrak{B}=(S,M,s_0,S-F)$. One need only prove in addition that $\mathcal{T}$ is closed under intersections. Suppose that U_1 and U_2 are in $\mathcal{T}$. By Theorem 2, let R_1 and R_2 be two right-invariant equivalence relations of finite index such that U_i is a union of equivalence classes under R_i for $i=1,2$. Consider the equivalence relation $R_3 = R_1 \cap R_2$, in other words xR_3y if and only if xR_1y and xR_2y. R_3 is, of course, right invariant. Every equivalence class under R_3 is an intersection of equivalence classes under R_1 and R_2. Hence, the number of equivalence classes for R_3 is at most the product of the numbers for R_1 and R_2. We see, then, that R_3 is of finite index. Now $U_1 \cap U_2$ is simply a union of intersections of the two kinds of equivalence classes, so that $U_1 \cap U_2$ is a union of equivalence classes under R_3, which shows that $U_1 \cap U_2$ is in $\mathcal{T}$ by Theorem 2. The proof is complete.

Corollary 5.1. *The class $\mathcal{T}$ contains all finite sets of tapes.*

This is a direct consequence of Theorems 3 and 5.

The proof of Theorem 5 may seem too abstract. To make it more direct, we show next how to form at once a machine defining the intersection.

Definition 7. *Let $\mathfrak{A} = (S,M,s_0,F)$ and $\mathfrak{B} = (T,N,t_0,G)$ be two automata. The* direct product *$\mathfrak{A} \times \mathfrak{B}$ is that automaton $(S\times T, M\times N, (s_0,t_0), F\times G)$ where $S\times T$ and $F\times G$ are the Cartesian products of sets, (s_0,t_0) is the ordered pair of s_0 and t_0, and the function $M\times N$ on $(S\times T)\times\Sigma$ is defined by the formula*

$$(M\times N)((s,t),\sigma) = (M(s,\sigma), N(t,\sigma))$$

for all s in S, t in T, and σ in Σ.

Theorem 6. *If $\mathfrak{A}$ and $\mathfrak{B}$ are automata, then*

$$T(\mathfrak{A}\times\mathfrak{B}) = T(\mathfrak{A}) \cap T(\mathfrak{B}).$$

Proof: An obvious inductive argument shows that for all tapes x we have $(M\times N)((s,t),x) = (M(s,x),N(t,x))$ for all s in S and t in T. Now x is in $T(\mathfrak{A}\times\mathfrak{B})$ if and only if $(M\times N)((s_0,t_0),x) = (M(s_0,x),N(t_0,x))$ is in $F\times G$. This in turn is equivalent to the conjunctions of conditions that $M(s_0,x)$ is in F and $N(t_0,x)$ is in G; in other words, x is in $T(\mathfrak{A}) \cap T(\mathfrak{B})$, as was to be shown.

• 4. *The emptiness problem*

Suppose someone gave you an automaton $\mathfrak{A} = (S,M,s_0,F)$ without telling you what it was supposed to do. The gift might turn out to be an elaborate practical joke, and $T(\mathfrak{A})$ could very well be empty. Now a person would not want to spend the rest of his life feeding all the infinite number of possible tapes into the machine if all the answers are going to be the same. Thus one would like to know an upper bound on the number of tapes that need be tried to determine whether the machine is of any use. Such an upper bound is supplied by the next theorem.

Theorem 7. *Let $\mathfrak{A}$ be an automaton. Then $T(\mathfrak{A})$ is not empty if and only if $\mathfrak{A}$ accepts some tape of length less than the number of internal states of $\mathfrak{A}$.*

Proof: We need only establish the implication from left to right. Assume that $T(\mathfrak{A})$ is not empty and indeed that x is a tape in $T(\mathfrak{A})$ of minimal length. Let n be the length of x and let r be the number of internal states of $\mathfrak{A}$. By way of contradiction, assume that $r \leq n$. It follows at once that there must exist integers $k < l \leq n$ such that

$$M(s_0, {}_0x_k) = M(s_0, {}_0x_l),$$

where ${}_0x_k$ and ${}_0x_l$ are the initial segments of x of length k and l. Consider the tape $x' = {}_0x_k\ {}_lx_n$ which is shorter than x. We have

$$\begin{aligned} M(s_0,x') &= M(s_0, {}_0x_k\ {}_lx_n) \\ &= M(M(s_0,{}_0x_k), {}_lx_n) \\ &= M(M(s_0,{}_0x_l), {}_lx_n) \\ &= M(s_0, {}_0x_l\ {}_lx_n) \\ &= M(s_0,x) \end{aligned}$$

because $x = {}_0x_l\ {}_lx_n$. Hence x' must be in $T(\mathfrak{A})$ also, which contradicts the minimum conditions on x and proves that $n < r$.

Corollary 7.1. *Given a finite automaton $\mathfrak{A}$ there is an effective procedure whereby in a finite number of steps it can be decided whether $T(\mathfrak{A})$ is empty.*

The corollary is an immediate consequence of the fact that Theorem 7 shows that only a finite number of tapes that need be tried, and any one tape can be run effectively through a machine once the table of moves has been given. It is also possible to give a simple necessary and sufficient condition of a similar nature for $T(\mathfrak{A})$ to be infinite. We precede that result by a lemma.

Lemma 8. *Let $\mathfrak{A}$ be an automaton with r internal states. Let x be a tape in $T(\mathfrak{A})$ of length n. If $r \leq n$, then there exist tapes y,z,w such that $x = yzw$, $z \neq \Lambda$, and all the tapes yz^mw are in $T(\mathfrak{A})$ for $m = 0,1,2\ldots.$*

Proof: As in Theorem 7, there must exist integers $k \leq l \leq n$ such that

$$M(s_0,{}_0x_k) = M(s_0,{}_0x_l).$$

Let $y = {}_0x_k$, $z = {}_kx_l$, $w = {}_lx_n$. Since $k < l$, we see that $z \neq \Lambda$. Clearly $x = yzw$, and $yz = {}_0x_l$, hence $M(s_0,y) = M(s_0,yz)$. It follows then at once by induction that $M(s_0,y) = M(s_0,yz^m)$. Whence, we derive

$$\begin{aligned} M(s_0,x) &= M(s_0,yzw) \\ &= M(M(s_0,yz),w) \\ &= M(M(s_0,yz^m),w) \\ &= M(s_0,yz^mw). \end{aligned}$$

Thus all the tapes yz^mw are also in $T(\mathfrak{A})$.

Theorem 9. *Let $\mathfrak{A}$ be an automaton with r internal states. Then $T(\mathfrak{A})$ is infinite if and only if it contains a tape of length n with $r \leq n \leq 2r$.*

Proof: The implication from right to left is a direct consequence of Lemma 8. Assume that $T(\mathfrak{A})$ is infinite. The alphabet Σ is finite, and so $T(\mathfrak{A})$ must contain tapes of length greater than any integer. Let x be a tape in $T(\mathfrak{A})$ of length $n \geq r$. As in the other two proofs, there must exist integers $k < l \leq n$ such that

$$M(s_0,{}_0x_k) = M(s_0,{}_0x_l).$$

Now take a new tape x which is of minimal length of any tape in $T(\mathfrak{A})$ for which integers $k < l$ exist satisfying the above equation. Assume further that l is the least such integer $\leq n =$ the length of x. We no longer know that $n \geqq r$. Thus if $i<j<l$, then

$$M(s_0, {}_0x_i) \neq M(s_0, {}_0x_j).$$

Since there are at most r values for the function M to assume, this proves that $l \leqq r$. Further, if $l \leqq i<j \leqq n$, then

$$M(s_0, {}_0x_i) \neq M(s_0, {}_0x_j),$$

since otherwise the tape $x' = {}_0x_i\,{}_jx_n$ would be a shorter tape than x satisfying the given conditions on x. Counting the number of indices between l and n, we see that $n-l+1 \leqq r$. Adding l to both sides and applying the previous inequality, we find $n+1 \leqq 2r$, or better, $n<2r$. If $r \leqq n$, then the proof would be complete; however, this may not be the case. Assume that $n<r$. Let $y={}_0x_k$, $z={}_kx_l$, $w={}_lx_n$. We have $x \neq \Lambda$, and all tapes yz^mw are in $T(\mathfrak{A})$. Let m be the least integer such that

$$r \leqq k+m(l-k)+(n-l).$$

Clearly $m \neq 0$, since $k+(n-l)<n<r$. If

$2r \leqq k+m(l-k)+(n-l)$, then

$$r \leqq k+(m-l)(l-k)+(n-l),$$

because $l-k \leqq n<r$. But this is impossible because m was chosen as the least such integer. Hence

$$k+m(l-k)+(n-l)<2r$$

and the number on the left is the length of yz^mw, which proves that there is some tape in $T(\mathfrak{A})$ of the indicated length.

Corollary 9.1. *Given a finite automaton $\mathfrak{A}$, there is an effective procedure whereby in a finite number of steps it can be decided whether $T(\mathfrak{A})$ is infinite.*

Corollary 9.2. *Let $\mathfrak{A}$ be a finite automaton with r in-internal states, and let the alphabet Σ have $q>1$ symbols. Then if $T(\mathfrak{A})$ is finite, it can have at most*

$$\sum_{k<r} q^k = \frac{q^r-1}{q-1} \text{ tapes.}$$

Notice also that Lemma 8 gives another proof that the set of tapes of the form 0^n10^n is not definable by any finite automaton.

Finally we shall treat in this section the question of deciding whether two automata define the same set of tapes.

Definition 8. *Two automata $\mathfrak{A}$ and $\mathfrak{B}$ are* equivalent *if $T(\mathfrak{A})=T(\mathfrak{B})$.*

Theorem 10. *Two automata $\mathfrak{A}$ and $\mathfrak{B}$ are not equivalent if and only if there is a tape x of length less than the product of the number of internal states of $\mathfrak{A}$ by that of $\mathfrak{B}$ which is accepted by one machine but not by the other.*

Proof: Let $\mathfrak{A}'$ be the machine having the same internal states as $\mathfrak{A}$ and defining the complement of $T(\mathfrak{A})$ as in the proof of Theorem 5. Similarly for $\mathfrak{B}$. $\mathfrak{A}$ and $\mathfrak{B}$ are not equivalent if and only if one of the sets $T(\mathfrak{A} \times \mathfrak{B}')$, $T(\mathfrak{A}' \times \mathfrak{B})$ is not empty. The theorem follows now directly from Theorem 7, Theorem 6 and Definition 7.

Corollary 10.1. *Given two finite automata $\mathfrak{A}$ and $\mathfrak{B}$, there is an effective procedure whereby in a finite number of steps it can be decided whether $\mathfrak{A}$ and $\mathfrak{B}$ are equivalent.*

All the results of this section are quite evident from the literature, e.g., Burks-Wang,[1] Section 2.2. Only Theorem 10 and its corollary are a little stronger than the corresponding results there because of a wider definition of equivalence of automata. These results are nonetheless included for completeness, since the general approach here is rather different.

Chapter II. Reductions to one-way automata

• 5. Nondeterministic operation

The automata used throughout Chapter I were strictly deterministic in their tape-reading action, which was uniquely determined by the table of moves, since there was one and only one way the machine would change its state in any particular situation. Requiring all machines to be of this form can lead to rather cumbersome details, in view of the large number of internal states needed even for some relatively elementary operations. In this section we introduce the notion of a *nondeterministic* automaton and show that any set of tapes defined by such a machine could also be defined by an ordinary automaton. The main advantage of these machines is the small number of internal states that they require in many cases and the ease in which specific machines can be described. Several examples of their use will be found in Section 6.

Definition 9. *A* nondeterministic (finite) automaton *over the alphabet Σ is a system $\mathfrak{A}=(S,M,S_0,F)$ where S is a finite set, M is a function of $S \times \Sigma$ with values in the set of all subsets of S, and S_0 and F are subsets of S.*

A nondeterministic automaton is not a probabilistic machine but rather a machine with many choices in its moves. At each stage of its motion across a tape it will be at liberty to choose one of several new internal states. Of course, some sequence of choices will lead either to impossible situations from which no moves are possible or to final states not in the designated class F. We disregard all such failures, however, and agree to let the machine accept a tape if there is at least one winning combination of choices of states leading to a designated final state. The next definition makes this convention precise.

Definition 10. *Let $\mathfrak{A}$ be a nondeterministic automaton. The set $T(\mathfrak{A})$ of tapes accepted by $\mathfrak{A}$ is the collection of all tapes $x=\sigma_0\sigma_1 \ldots \sigma_{n-1}$ for which there exists a sequence $s_0, s_1, \ldots, s_n$ of internal states of $\mathfrak{A}$ such that*

(i) *s_0 is in S_0;*

(ii) *s_i is in $M(s_{i-1}, \sigma_{i-1})$, for $i=1,2,\ldots,n$;*

(iii) *s_n is in F.*

It is readily seen that if $\mathfrak{A}$ is a nondeterministic machine such that $M(s,\sigma)$ consists of exactly one internal state for each s in S and σ in Σ, then $\mathfrak{A}$ is really the same as an ordinary automaton, and $T(\mathfrak{A})$ will contain the expected tapes. Thus ordinary automata are special cases of nondeterministic automata, and we shall freely identify the ordinary machines with their counterparts.

One might imagine at first sight that these new machines are more general than the ordinary ones, but this is not the case. We shall give a direct construction of an ordinary automaton, defining exactly the same set of tapes as a given nondeterministic machine.

Definition 11. *Let $\mathfrak{A}=(S,M,S_0,F)$ be a nondeterministic automaton. $\mathfrak{D}(\mathfrak{A})$ is the system (T,N,t_0,G) where T is the set of all subsets of S, N is a function on $T\times\Sigma$ such that $N(t,\sigma)$ is the union of the sets $M(s,\sigma)$ for s in t, $t_0=S_0$, and G is the set of all subsets of S containing at least one member of F.*

Clearly $\mathfrak{D}(\mathfrak{A})$ is an ordinary automaton, but it is actually equivalent to $\mathfrak{A}$.

Theorem 11. *If $\mathfrak{A}$ is a nondeterministic automaton,* then $T(\mathfrak{A})=T(\mathfrak{D}(\mathfrak{A}))$.

Proof: Assume first that a tape $x=\sigma_0\sigma_1\ldots\sigma_{n-1}$ is in $T(\mathfrak{A})$ and let $s_0,s_1,\ldots,s_n$ be a sequence of internal states satisfying the conditions of Definition 10. We show by induction that for $k\leqq n$, s_k is in $N(t_{0,0}x_k)$. For $k=0$, $N(t_{0,0}x_k)=N(t_0,\Lambda)=t_0=S_0$ and we were given that s_0 is in S_0. Assume the result for $k-1$. By definition, $N(t_{0,0}x_k)=N(N(t_{0,0}x_{k-1}),\sigma_{k-1})$. But we have assumed s_{k-1} is in $N(t_{0,0}x_{k-1})$ so that from the definition of N we have $M(s_{k-1},\sigma_{k-1})\subset N(t_{0,0}x_k)$. However, s_k is in $M(s_{k-1},\sigma_{k-1})$, and so the result is established. In particular s_n is in $N(t_{0,0}x_n)=N(t_0,x)$, and since s_n is in F, we have $N(t_0,x)$ in G, which proves that x is in $T(\mathfrak{D}(\mathfrak{A}))$. Hence, we have shown that

$$T(\mathfrak{A})\subset T(\mathfrak{D}(\mathfrak{A})).$$

Assume next that a tape $x=\sigma_0\sigma_1\ldots\sigma_{n-1}$ is in $T(\mathfrak{D}(\mathfrak{A}))$. Let for each $k\leqq n$, $t_k=N(t_{0,0}x_k)$. We shall work backwards. First, we know that t_n is in G. Let then s_n be any internal state of $\mathfrak{A}$ such that s_n is in t_n and s_n is in F. Since s_n is in

$$t_n=N(t_{0,0}x_n)=N(t_{n-1},\sigma_{n-1}),$$

we have from the definition of N that s_n is in $M(s_{n-1},\sigma_{n-1})$ for some s_{n-1} in t_{n-1}. But

$$t_{n-1}=N(t_{0,0}x_{n-1})=N(t_{n-2},\sigma_{n-2}),$$

so that s_{n-1} is in $M(s_{n-2},\sigma_{n-2})$ for some s_{n-2} in t_{n-2}. Continuing in this way we may obtain a sequence, $s_n,s_{n-1},s_{n-2},\ldots,s_0$ such that s_k is in t_k; s_k is in $M(s_{k-1},\sigma_{k-1})$, for $k>0$; and s_n is in F. Since $t_0=S_0$, we also have s_0 in S_0, which proves that x is in $T(\mathfrak{A})$. Thus, $T(\mathfrak{D}(\mathfrak{A}))\subset T(\mathfrak{A})$, which completes the proof.

This theorem has many interesting consequences. For example, it shows that any automaton with several initial states can be replaced by an equivalent automaton with but one initial state. It would seem that the notions of final state and initial state should be dual in some sense. But one must be careful, because, as the reader may easily show for himself, with the alphabet $\Sigma=\{0,1\}$ the set of all tapes of the form 0^n or 1^n cannot be defined by any nondeterministic automaton with but one designated final state. The correct notion of duality between initial and final states is connected with the reversal of right and left, as indicated in the next definition and theorem.

Definition 12. *Let $\mathfrak{A}=(S,M,S_0,F)$ be a nondeterministic automaton. The* dual *of $\mathfrak{A}$ is the machine $\mathfrak{A}^*=(S,M^*,F,S_0)$ where the function M^* is defined by the condition*

s' is in $M^*(s,\sigma)$ if and only if s is in $M(s',\sigma)$.

Notice that we have at once the equation $\mathfrak{A}^{**}=\mathfrak{A}$. The relation between the sets defined by an automaton and its dual is as follows.

Theorem 12. *If $\mathfrak{A}$ is a nondeterministic automaton, then $T(\mathfrak{A}^*)=T(\mathfrak{A})^*$.*

Proof: In view of the equality $\mathfrak{A}^{**}=\mathfrak{A}$, we need only show $T(\mathfrak{A}^*)\subset T(\mathfrak{A})^*$. Let $x=\sigma_0\sigma_1\ldots\sigma_{n-1}$ be a tape in $T(\mathfrak{A}^*)$; we must show that x^* is in $T(\mathfrak{A})$. Let $s_0,s_1,\ldots,s_n$ be the sequence of internal states of $\mathfrak{A}^*$ such that s_0 is in F, s_n is in S_0 and s_k is in $M^*(s_{k-1},\sigma_{k-1})$ for $k=1,2,\ldots,n$. Define a new sequence $s'_0,s'_1,\ldots,s'_n$ by the equation $s'_k=s_{n-k}$ for $k\leqq n$. Obviously, s'_0 is in S_0 and s'_n is in F. Further, for $k>0$ and $k\leqq n$, $s'_{k-1}=s_{n-k+1}$ is in $M^*(s_{n-k},\sigma_{n-k})$, or in other words, $s_{n-k}=s'_k$ is in $M(s'_{k-1},\sigma_{n-k})$. Now defining a new sequence of symbols $\sigma'_0\sigma'_1\ldots\sigma'_{n-1}$ by the formula $\sigma'_k=\sigma_{n-k-1}$, we see that $\sigma'_{k-1}=\sigma_{n-k}$ and $\sigma'_0\sigma'_1\ldots\sigma'_{n-1}=x^*$. Thus, x^* is in $T(\mathfrak{A})$ as was to be proved.

It should be noted that Theorem 12 together with Theorem 11 yields a direct construction and proof for Theorem 4 of Section 3 which was first proved by the indirect method of Theorem 1. In the next section we make heavy use of the direct constructions supplied by the nondeterministic machines to obtain results not easily apparent from the mathematical characterizations of Theorems 1 and 2.

6. *Further closure properties*

Simplifying a result due originally to Kleene, Myhill in unpublished work has shown that the class $\mathcal{T}$ can be characterized as the least class of sets of tapes containing the finite sets and closed under some simple operations on sets of tapes. We indicate here a different proof using the method developed in the preceding section.

First of all, we need to define the operations on sets of tapes. Let U and V be two sets of tapes. By the *complex product* UV of U and V we understand the collection of all tapes of the form xy with x in U and y in V. Clearly the product of sets satisfies the associative law:

$$(UV)W=U(VW).$$

This leads to the introduction of finite exponents where we define $U^n=UU\ldots U(n \text{ times})$ with the convention than $U^0=\{\Lambda\}$. Finally, if U is a set of tapes we can form the *closure* of U, in symbols $cl(U)$, which is the least set V containing U, having Λ as an element, and such that whenever x, y are in V then xy is in V. Another definition is given by the equation

$$cl(U)=U^0\cup U^1\cup U^2\cup U^3\ldots,$$

where the infinite union extends all over finite exponents. We may prove at once about these operations that the class $\mathcal{T}$ is closed under them.

Theorem 13. *The class $\mathcal{T}$ is closed under the formation of complex products and closures of sets in $\mathcal{T}$.*

Proof: Assume first that U, V are in $\mathcal{T}$. Let $U=T(\mathfrak{A})$ and $V=T(\mathfrak{B})$ where $\mathfrak{A}$ and $\mathfrak{B}$ are ordinary automata with $\mathfrak{A}=(S,M,s_0,F)$ and $\mathfrak{B}=(T,N,t_0,G)$. We need only find a nondeterministic machine $\mathfrak{C}$ such that $UV=T(\mathfrak{C})$. We may assume that the sets S and T have no elements in common, and then equate $\mathfrak{C}=(S\cup T,P,\{s_0\},G)$ where the function P is defined as follows:

$P(s,\sigma)=\{M(s,\sigma)\}$, if s is in $S-F$;

$P(s,\sigma)=\{M(s,\sigma),N(T_0,\sigma)\}$, if s is in F;

$P(t,\sigma)=N(t,\sigma)$, if t is in T.

The straightforward proof that $\mathfrak{C}$ has the desired property is left to the reader.

Next, we must show why $Hcl(U)$ is in $\mathcal{T}$. We construct a machine $\mathfrak{D}$ such that $cl(U)=T(\mathfrak{D})$, where $\mathfrak{D}$ is allowed to be nondeterministic. Simply let $\mathfrak{D}=(S,Q,s_0,F)$, where the function Q is defined as follows:

$Q(s,\sigma)=\{M(s,\sigma)\}$, if s is in $S-F$;

$Q(s,\sigma)=\{M(s,\sigma),M(s_0,\sigma)\}$, if s is in F.

The easy completion of the proof is left to the reader.

Theorem 14. (Kleene-Myhill). *The class $\mathcal{T}$ is the least class of sets of tapes containing the finite sets and closed under the formation of unions, complex products, and closures of sets.*

The full proof of Theorem 14 will not be given. Instead we give a brief account of the method of proof needed. Let $\mathcal{U}$ be the least class closed under the operations mentioned in the theorem. That $\mathcal{U}\subset\mathcal{T}$ is the content of Theorems 5, 5.1, and 13. To prove that $\mathcal{T}\subset\mathcal{U}$, consider each set in $\mathcal{T}$ to be of the form $T(\mathfrak{A})$, where $\mathfrak{A}$ is nondeterministic, and proceed by a kind of induction on $\mathfrak{A}$. In more precise terms, define the *weight* of $\mathfrak{A}$, in symbols $|\mathfrak{A}|$, to be the sum of all the cardinal numbers of the sets $M(s,\sigma)$ for all s in S and σ in Σ. Then by assuming that $T(\mathfrak{B})$ is in $\mathcal{U}$ for all $\mathfrak{B}$ with $|\mathfrak{B}|<|\mathfrak{A}|$, one can prove that $T(\mathfrak{A})$ is also in $\mathcal{U}$. The details, however, are tiring.

This discussion completes our survey of the closure properties of the class of definable sets begun in Section 3, and the authors are not aware of any other interesting operations on sets that can be effected by constructions of automata that we have not already indicated. The remainder of this paper will be therefore devoted to generalizations of the notion of an automaton.

7. *Two-way automata*

Trying to further generalize the notion of an automaton, we consider automata which are not confined to a strict forward motion across their tapes. This leads to the following definition, which is a direct extension of Definition 1.

Definition 13. *Let $L=\{-1,0,+1\}$. A* two-way (finite) automation *over a finite alphabet Σ is a system $\mathfrak{A}=(S,M,s_0,F)$ where S is a finite non-empty set (the set of* internal states *of $\mathfrak{A}$), M is a function from $S\times\Sigma$ into $L\times S$ (the table of* moves *of $\mathfrak{A}$), s_0 is an element of S (the* initial state *of $\mathfrak{A}$), and F is a subset of S (the set of* designated final states *of $\mathfrak{A}$).*

A two-way automaton $\mathfrak{A}$ operates as follows: When given a tape, i.e., a finite linear sequence of squares each containing a single symbol of the alphabet Σ, $\mathfrak{A}$ is set in internal state s_0 scanning the first (leftmost) square of the tape. At each stage of the machine's operation, if the internal state is s, the scanned symbol is σ, and $M(s,\sigma)=(p,s')$, where p is one of $-1,0,1$, then $\mathfrak{A}$ will move one square to the left, stay where it is, or move one square to the right, according as $p=-1,0,1$; furthermore, $\mathfrak{A}$ will enter internal state s'. The operation described just now is called an *atomic step* of $\mathfrak{A}$. After completion of an atomic step, $\mathfrak{A}$ is again in a certain internal state scanning a certain symbol, and a new atomic step is performed, and so on.

If, when operating in this way on a given tape, $\mathfrak{A}$ will eventually *get off* the tape on the right side and at that time be in a state in F, then we shall say that the tape is *accepted* by $\mathfrak{A}$. The formal definition is as follows:

Definition 14. *The set $T(\mathfrak{A})$ of tapes* accepted *by the two-way automaton $\mathfrak{A}$ is the set of all sequences $\sigma_0\ldots\sigma_{n-1}$ of symbols from the alphabet Σ for which there exist an integer $m>0$, a sequence of integers $p_0,\ldots,p_m$, and a sequence $s_0,\ldots,s_m$ of internal states of $\mathfrak{A}$ such that*

(i) $p_0=0$ *and s_0 is the initial state of $\mathfrak{A}$;*
(ii) $0\leq p_i<n$ *for $i=0,\ldots,m-1$;*
(iii) $p_m=n$ *and s_m is in F;*
(iv) $(p_i-p_{i-1},s_i)=M(s_{i-1},\sigma_{p_{i-1}})$ *for $i=1,\ldots,m$.*

In the above definition the sequence $p_0,\ldots,p_m$ should be interpreted as the sequence of positions of the machine $\mathfrak{A}$ on the tape; thus, p_i-p_{i-1} indicates the change in position of the machine from time $i-1$ to time i. Condition (ii), for example, means that the machine does not run off the tape before the computation has been completed.

In analogy with Definition 3 we shall say that a set P of tapes is *definable by a two-way automaton* if there exists some two-way automaton $\mathfrak{A}$ such that $T(\mathfrak{A})=P$.

To avoid confusion we shall, from now on, refer to the automata discussed in Sections 1-6 as *one-way automata.*

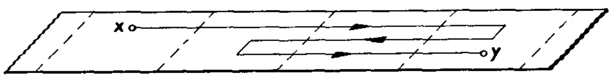
Let us consider an example of a two-way machine illustrating the complicated fashion in which such a machine can operate on a given tape. Let $\mathfrak{A}$, $\mathfrak{B}$, and $\mathfrak{C}$, be three one-way automata over the same alphabet Σ. We combine these automata into a single two-way automaton $\mathfrak{D}$ having the following flow diagram. Given a tape t the automaton $\mathfrak{A}$ (which we imagine as being a part of $\mathfrak{D}$) starts reading it on the left end and proceeds from left to right until a designated final state of $\mathfrak{A}$ is reached; when this happens $\mathfrak{D}$ goes into the initial state of $\mathfrak{B}$ and starts reading the tape from right to left until a designated final state of $\mathfrak{B}$ is reached; when this happens $\mathfrak{D}$ switches into the initial state of $\mathfrak{A}$ and again starts moving from left to right, and so on; all this time automaton

$\mathfrak{C}$ is reading the tape symbols as they come in (i.e., in the sequence in which they are being scanned by $\mathfrak{D}$) and t is accepted by $\mathfrak{D}$ only if $\mathfrak{D}$ ever gets off the right-hand end of t and at that time $\mathfrak{C}$ is in one of its final designated states. It seems to be quite difficult to determine the kind of set of tapes defined by $\mathfrak{D}$. It turns out, to our surprise, that the following theorem holds.

Theorem 15. *For every two-way automaton $\mathfrak{A}$ there exists a one-way automaton $\overline{\mathfrak{A}}$ such that $T(\mathfrak{A}) = T(\overline{\mathfrak{A}})$. Furthermore, $\overline{\mathfrak{A}}$ can be obtained effectively from $\mathfrak{A}$.*

*Outline of Proof:** By definition, a *Z-motion* of $\mathfrak{A}$ on a tape t consists of $\mathfrak{A}$ moving across a square x in a certain direction up to a square y, changing direction at y and moving back towards x, changing again direction before passing x and moving up to y; a Z-motion thus contains exactly two changes of direction. While operating on a tape t a two-way automation will in general perform a complicated succession of forward and backward motions before accepting or rejecting t. In particular, $\mathfrak{A}$ will go through a great number of Z-motions.

In a given Z-motion in the diagram,



the internal state s' in which $\mathfrak{A}$ re-enters y is a function of the state s in which $\mathfrak{A}$ originally entered y and the portion of the tape from x to y. If it were possible to compute this new state s', without actually having to move back, then we could substitute for $\mathfrak{A}$ a new automaton which, instead of turning back at y, would simply go directly from state s into state s' and thus the Z-motion would be eliminated. It turns out that the computation of s' from s is indeed possible because the set $R(s,s')$ ($L(s,s')$) of tapes such that when $\mathfrak{A}$ starts on the right-(left) hand end it will go through a simple loop (i.e., move directly to some square, change direction there, and go straight back to where it started) and arrive back in state s', is definable by a one-way automaton. Combining $\mathfrak{A}$ with these one-way automata it is possible to define a new *derived automaton* $\mathfrak{A}'$ which on any given tape t performs fewer Z-motions than $\mathfrak{A}$ does and such that $T(\mathfrak{A}') = T(\mathfrak{A})$. We then show that, by repeating this derivation operation a sufficient number of times, a one-way automaton is obtained which defines the same set as $\mathfrak{A}$. This depends on the fact that there is a bound, common to all tapes t accepted by $\mathfrak{A}$, on the number of times $\mathfrak{A}$ goes through any square of t; this bound being the number of internal states of $\mathfrak{A}$.

Corollary 15.1. *The equivalence problem for two-way automata is effectively solvable.*

Proof: Given two two-way automata $\mathfrak{A}$ and $\mathfrak{B}$, to decide whether $T(\mathfrak{A}) = T(\mathfrak{B})$ construct one-way automata $\overline{\mathfrak{A}}$ and $\overline{\mathfrak{B}}$ such that $T(\overline{\mathfrak{A}}) = T(\mathfrak{A})$ and $T(\overline{\mathfrak{B}}) = T(\mathfrak{B})$; by the previous theorem this can be done effectively. Apply now to $\overline{\mathfrak{A}}$ and $\overline{\mathfrak{B}}$ the procedure given in Corollary 10.1.

*The result, with its original proof, was presented to the Summer Institute of Symbolic Logic in 1957 at Cornell University. Subsequently J. C. Shepherdson communicated to us a very elegant proof which also appears in this Journal.[7] In view of this we confine ourselves here to sketching the main ideas of our proof.

Chapter III. *Multitape automata*

• 8. *Description and definitions*

We turn now to the study of multitape machines, fixing our attention, without any real loss of generality, on the two-tape case. We can picture the two-tape machine $\mathfrak{A}$ as having two scanning heads reading a pair (t_0,t_1) of tapes. We adopt the convention that the machine will read for a while on one tape, then change control and read for a while on the other tape, and so on until one of the tapes is exhausted. When this happens $\mathfrak{A}$ stops and the pair (t_1,t_2) is accepted if and only if $\mathfrak{A}$ is in a designated final state. Thus, with a two-tape automaton, a set of pairs of tapes is defined, or we can say a binary relation between tapes is defined.

To make two-way automata more versatile we afford them with the ability to anticipate the end of the tape. This arrangement consists in augmenting the alphabet Σ with an *end-marker* ε and always feeding into the automaton pairs of the form $(t_0\varepsilon, t_1\varepsilon)$; here t_1 and t_2 do not contain ε, the latter being merely a technical symbol.

In order to indicate the change of control from one tape to the other we use the device of dividing the states of the machine into two classes: the first class contains those states in which the first tape is being read, while the second class has to do with the second tape. These remarks should serve as sufficient background for the following formal definition.

Definition 15. *A* two-tape, one-way automaton *over an alphabet Σ is a system $\mathfrak{A} = (S,M,s_0,F,C_0,C_1)$ where (S,M,s_0,F) is an ordinary automaton; except that M is a function from $S \times (\Sigma \cup \{\varepsilon\})$ into S, and where the sets C_0,C_1 form a partition of S, i.e., $C_0 \cap C_1 = \phi$ and $C_0 \cap C_1 = S$.*

Thus a two-tape machine is just an ordinary automaton having an additional structure to determine which tape is to be read.

To be able to define explicitly when a pair of tapes is accepted by an automaton, the following notation involving the partition of the set of states is needed.

Let $\mathfrak{A} = (S,M,s_0,F,C_0,C_1)$ be a two-tape automaton and let $s_0,s_1,\ldots,s_n$ be a sequence of states (where s_0 is the initial state). Then there is a unique pair of *associated* sequences of integers $k_0,\ldots,k_n;l_0,\ldots,l_n$ such that:

(i) k_i is 0 or 1 according as s_i is in C_0 or C_1;

(ii) l_i is the number of indices $j<i$ such that s_j is in C_{k_i}.

Definition 16. *The set of all pairs of tapes accepted by a two-tape automaton $\mathfrak{A}$, in symbols $T_2(\mathfrak{A})$, is the set of all pairs (t_1,t_2) on the alphabet Σ such that for*

$$(t_1\varepsilon, t_2\varepsilon) = (\sigma_{00}\sigma_{01}\cdots\sigma_{0(m-1)}, \sigma_{10}\sigma_{11}\cdots\sigma_{1(n-1)})$$

there is a (unique) sequence of states $s_0,s_1,\ldots,s_p$ and associated sequences of integers $k_0,\ldots,k_p;\ l_0,\ldots,l_p$ such that

(i) s_0 *is the initial state of* $\mathfrak{A}$;
(ii) $s_i = M(\sigma_{k_{i-1}} l_{i-1} s_{i-1})$ *for* $i=1, \ldots, p$;
(iii) *if* $k_{p-1}=0$ *then* $l_{p-1}=m-1$ *and if* $k_{p-1}=1$ *then* $l_{p-1}=n-1$;
(iv) s_p *is in* F.

In the above definition we are, of course, assuming that if $k_i=0$, then $l_i<m$ and if $k_i=1$, then $l_i<n$ otherwise condition (ii) would be meaningless.

• 9. *Relation to one-tape automata*

Two-tape automata behave in a fashion almost identical with that of one-tape automata, the only difference being that they operate on two tapes. It is therefore natural to try to establish relationships between the sets of pairs of tapes definable by two-tape machines and the sets of tapes definable by one-tape automata.

Theorem 16. *Let* $\mathfrak{A}=(S,M,s_0,F,C_0,C_1)$ *be a two-tape automaton. The set of all tapes* t_1 *for which there exists some tape* t_2 *such that* (t_1,t_2) *is in* $T_2(\mathfrak{A})$ *(i.e., the* domain *of the relation defined by* $\mathfrak{A}$*) is definable by a one-tape automaton. An automaton defining this set can in fact be constructed effectively from* $\mathfrak{A}$.

Proof: The idea underlying the proof is that on the first component of any pair of tapes ($\mathfrak{A}$) operates like a nondeterministic one-tape machine. Once we are able to define the one-tape, nondeterministic automaton accepting precisely the tapes t_1 for which (t_1,t_2) is in $T_2(\mathfrak{A})$ for some t_2 the proof is completed by Theorem 11.

To shorten the argument we shall consider a slightly simplified version of the notion of two-tape automata; namely, in Definitions 15 and 16 we disregard the end symbol ε and the special role it plays (it is possible to extend the proof to cover the general case). A pair (t_1,t_2) is thus fed directly into $\mathfrak{A}$ and is said to be accepted if and when $\mathfrak{A}$ gets off one of the tapes in a designated final state of $\mathfrak{A}$.

Let s'' be in C_0 and s' be in C_1. A tape t on the alphabet Σ is called a (s',s'') *transition tape* if $\mathfrak{A}$, when started on t in s' will go through states in C_1 until it gets off t in s''. For every pair (s',s'') for which there exists some transition tape let $t(s',s'')$ denote a shortest one. The length of $t(s',s'')$ is clearly less than the number of states in C_1 so that all shortest transition tapes, and hence all pairs of states possessing a transition tape, can be effectively found.

A state s' in C_1 will be called a *finalizing state* if there exists a tape $t(s')$ such that $\mathfrak{A}$, when started on $t(s')$ in s', will go in states of C_1 to the end of $t(s')$ and get off the tape in a designated final state of $\mathfrak{A}$.

Define now a nondeterministic one-tape automaton $\mathfrak{B}$ as follows. Let f be some new element not in S, the set of states of $\mathfrak{B}$ is $C_0 \cup \{f\}$. The table N of moves of $\mathfrak{B}$ is defined by

(i) $N(f,\sigma)=\{f\}$;
(ii) $N(s,\sigma)=\{M(s,\sigma)\}$, if $M(s,\sigma)$ is in C_0;
(iii) $N(s,\sigma)=\{f\}$, if $M(s,\sigma)$ is a finalizing state in C_1;
(iv) $N(s,\sigma)=$the set of all s'' where there exists a transition tape $t(M(s,\sigma),s'')$, otherwise.

The set of initial states of $\mathfrak{B}$ is $\{s_0\}$ if s_0 is in C_0 and is the set of s'' for which there exists a tape $t(s_0,s'')$, if s_0 is in C_1. The set of designated final states of $\mathfrak{B}$ $(C_0 \cap F) \cup \{f\}$.

It is left for the reader to verify that the set of all tapes accepted by $\mathfrak{B}$ is precisely the domain of the relation $T_2(\mathfrak{A})$; we recall at this point the simplified definition of acceptance used in the proof. This completes our proof.

Corollary 16.1. *There are effective procedures whereby, given a two-tape automaton* $\mathfrak{A}$, *it can be decided in a finite number of steps whether* $T_2(\mathfrak{A})$ *is empty and whether* $T_2(\mathfrak{A})$ *is infinite.*

Proof: Construct the one-tape automata $\mathfrak{B}$ and $\mathfrak{C}$ defining the domain and range of the relation $T_2(\mathfrak{A})$. The set $T_2(\mathfrak{A})$ is empty if and only if $T(\mathfrak{B})$ is empty. The set $T_2(\mathfrak{A})$ is infinite if and only if at least one of $T(\mathfrak{B})$ and $T(\mathfrak{B})$ is infinite. Now apply Corollaries 7.1 and 9.1.

Corollary 16.2. *If* $T_2(\mathfrak{A})$ *contains only pairs of the form* (t,t) *(i.e.,* $\mathfrak{A}$ *defines a* diagonal relation*) then the set of all tapes* t *for which* (t,t) *is in* $T_2(\mathfrak{A})$ *is definable by a one-tape automaton.*

• 10. *Impossibility of Boolean operations*

Whereas the class of sets definable by one-tape automata is closed under the Boolean operations (Theorem 5), when we come to sets of pairs definable by two-tape automata the situation is markedly different.

Theorem 17. *The class of all sets definable by two-tape automata is (i) closed under complementation; (ii) is not closed under intersection and union.**

Proof: (i) Let $\mathfrak{A}=(S,M,s_0,F,C_0,C_1)$. The complement, with respect to the set of all pairs of tapes on Σ, of $T_2(\mathfrak{A})$, is $T_2((S,M,s_0,S-F,C_0,C_1))$. (ii) Let $\Sigma=\{0,1\}$ and use the notation 0^n to denote the tape containing n zeroes. The sets $U=\{(0^n10^m, 0^k10^n), n,m,k=1,2,\ldots\}$ and $V=\{(t,t), t$ runs through all tapes$\}$ are definable by two-tape automata. Now $B \cap D=\{(0^n10^n, 0^n10^n), n=1,2,\ldots\}$. If this set were definable by a two-tape automaton then, by Corollary 16.2, the set $\{0^n10^n, n=1,2,\ldots\}$ would be definable by a one-tape automaton, which is impossible. That the class of definable sets is not closed with respect to unions now follows from the identity $\mathfrak{A} \cap \mathfrak{B}=T-[(T-\mathfrak{A}) \cup (T-\mathfrak{B})]$ and (i).

• 11. *Unsolvability of the intersection problem*

We have shown that the emptiness problem for two-tape automata is effectively solvable. It will now turn out that a similar elementary problem is not solvable. As a preparation for this result concerning automata we must recall a theorem of E. Post.[6]

The *correspondence problem* is the following: Given two equally long ordered lists $a_1,a_2,\ldots,a_n$ and $b_1,b_2,\ldots,b_n$ of tapes on the alphabet Σ, to decide whether there exist a sequence of indices $i_1,i_2,\ldots,i_k$, where

---

*J. C. Shepherdson informed us in a letter about a different simple example for the fact that the class of definable relations is not closed under intersections.

$1 \leqq i_j \leqq n$, such that

$$a_{i_1}\, a_{i_2} \ldots a_{i_k} = b_{i_1}\, b_{i_2} \ldots b_{i_k}.$$

E. Post proved that the correspondence problem (for an alphabet with more than one letter) is not effectively solvable.

Theorem 18. *The problem whether for two finite two-tape automata* $\mathfrak{A}_1$ *and* $\mathfrak{A}_2$ *we have* $T_2(\mathfrak{A}_1) \cap T_2(\mathfrak{A}_2) = \phi$ *(the empty set) is not effectively solvable.*

Proof: Corresponding to every sequence, $a_1, a_2, \ldots, a_n$ of words on our alphabet Σ construct a set $P(a_1, a_2, \ldots, a_n)$ of pairs of tapes as follows: We may assume that 0,1 are in Σ; if i is an integer let $\mathbf{i}$ be the tape consisting of i symbols 1 followed by a single 0. Now (t_1, t_2) is in $P(a_1, a_2, \ldots, a_n)$ if and only if for some k

(i) $t_1 = a_{i_1}\, a_{i_2} \ldots a_{i_k}$

(ii) $t_2 = \mathbf{i}_1 \mathbf{i}_2 \ldots \mathbf{i}_k$, where $i_j \leqq n$.

It is not hard to construct a two-tape automaton $\mathfrak{A}(a_1, a_2, \ldots, a_n) = \mathfrak{A}$ such that $T_2(\mathfrak{A}) = P(a_1, a_2, \ldots, a_n)$. Namely, to check whether a pair (t_1, t_2) satisfies conditions (*i*) and (*ii*), $\mathfrak{A}$ will start on t_2 and count the number of symbols 1 until the first 0 is met, let this number be i_1. The machine then switches to t_1 and checks whether this tape begins with a_{i_1}; if it does not, then (t_1, t_2) is not accepted. If t_1 does begin with a_{i_1}, then after reading through a_{i_1} the machine switches back to t_2, and the whole process is repeated. If at any time a symbol other than 0 or 1 is found on t_2, or if t_2 contains a run of more than n symbols 1 or more than one symbol 0, then the pair (t_1, t_2) is not accepted. These remarks sufficiently indicate the construction of $\mathfrak{A}$ and we shall not go into further detail.

Given two sequences of words $S_1 = (a_1, a_2, \ldots, a_n)$ and $S_2 = (b_1, b_2, \ldots, b_n)$ then $P(a_1, a_2, \ldots, a_n) \cap P(b_1, b_2, \ldots, b_n) \neq \phi$ if and only if the Post correspondence problem of S_1 and S_2 has a solution. Since the correspondence problem is not effectively solvable it follows that the problem whether

$$T_2(\mathfrak{A}(a_1, \ldots, a_n)) \cap T_2(\mathfrak{A}(b_1, \ldots, b_n)) \neq \phi$$

is not effectively solvable.

12. *Two-way, two-tape automata*

Turning now to *two-way, two-tape automata* we find that all hope of any constructive decision processes is lost. It is even impossible to decide, by a constructive decision method applicable to all automata, whether a two-way, two-tape machine accepts any tapes. To prove this formally it is, of course, necessary to give the explicit definition of a two-way machine. We shall not give the details here, since they are long and not very much different from the formal definitions needed for two-way, one-tape automata. The main point is that, as with the two-way, one-tape automaton, the table of moves of a two-way, two-tape automaton sometimes requires the machine to back up from the scanned square. However, an outline of the proof should clarify the method.

It was shown above that there is no constructive decision method for deciding whether two two-tape, one-way machines $\mathfrak{A}_1$ and $\mathfrak{A}_2$ both accept a common pair of tapes, that is, whether $T_2(\mathfrak{A}_1) \cap T_2(\mathfrak{A}_2) \neq \phi$. From the construction of the two-tape machines it follows that if h is a new symbol not in the alphabet Σ, then there is a one-one correspondence between all two-tape, one-way machines $\mathfrak{A}$ over Σ and certain two-tape one-way machines $\mathfrak{A}'$ over $\Sigma \cup \{h\}$ such that a pair of tapes (t_1, t_2) is in $T_2(\mathfrak{A})$ if and only if (ht_1h, ht_2h) is in $T_2(\mathfrak{A}')$. In words, we simply put a marker at the ends of the tapes, and all accepted tapes must be of this form. Let now $\mathfrak{A}_1$ and $\mathfrak{A}_2$ be any two two-tape machines. The corresponding machines over $\Sigma \cup \{h\}$ are $\mathfrak{A}'_1$ and $\mathfrak{A}'_2$. Now because $\mathfrak{A}'_1$ and $\mathfrak{A}'_2$ only accept tapes with markers at the ends, they can be glued together into a two-way machine $\mathfrak{B}$ such that $T_2(\mathfrak{B}) = T_2(\mathfrak{A}'_1) \cap T_2(\mathfrak{A}'_2)$. The two-way motion of $\mathfrak{B}$ is obvious: first run through the tapes in the style of $\mathfrak{A}_1$ to see if the pair is accepted, and then, after hitting the markers at the right end, run backwards until the left markers are hit, at which time the motion is again reversed, and the machine is started over, running in the style of $\mathfrak{A}_2$.

The outline of the construction given above shows that every intersection problem about one-way machines $\mathfrak{A}_1$ and $\mathfrak{A}_2$ is equivalent to the intersection problem about machines $\mathfrak{A}'_1$ and $\mathfrak{A}'_2$, which in turn is equivalent to the emptiness problem for a two-way machine $\mathfrak{B}$. Since there is an effective method for showing these equivalences, and since there is no effective solution of the intersection problem for one-way machines, we have proved the following.

Theorem 19. *There is no effective method of deciding whether the set of tapes definable by a two-tape, two-way automaton is empty or not.*

An argument similar to the above one will show that the class of sets of pairs of tapes definable by two-way, two-tape automata is closed under Boolean operations. In view of Theorem 17, this implies that there are sets definable by two-way automata which are not definable by any one-way automaton; thus no analogue to Theorem 15 holds.

References

1. A. W. Burks and Hao Wang, "The logic of automata," *Journal of the Association for Computing Machinery,* **4,** 193-218 and 279-297 (1957).
2. S. C. Kleene, "Representation of events in nerve nets and finite automata," *Automata Studies,* Princeton, pp. 3-41, (1956).
3. W. S. McCulloch and E. Pitts, "A logical calculus of the ideas imminent in nervous activity," *Bulletin of Mathematical Biophysics,* **5,** 115-133 (1943).
4. E. F. Moore, "Gedanken-experiments on sequential machines," *Automata Studies,* Princeton, pp. 129-153 (1956).
5. A. Nerode, "Linear automaton transformations," *Proceedings of the American Mathematical Society,* **9,** 541-544 (1958).
6. E. Post, "A variant of a recursively unsolvable problem," *Bulletin of the American Mathematical Society,* **52,** 264-268 (1946).
7. J. C. Shepherdson, "The reduction of two-way automata to one-way automata," *IBM Journal,* **3,** 198-200 (1959).

Revised manuscript received August 8, 1958