Mappings of Languages by Two-Tape Devices

Seymour Ginsburg

System Development Corporation, Santa Monica, Calif.

and

Edwin H. Spanier

University of California at Berkeley, California

Abstract. Several devices with two input lines and one output line are introduced. These devices are viewed as transformations which operate on pairs of (Algol-like) languages. Among the results proved are the following: (i) a pair consisting of a language and a regular set is transformed into a language; (ii) let (V, W) be a pair consisting of a language and a regular set. Then the set of those words w_1, for which there exists a word w_2 in V so that (w_1, w_2) is mapped into W, is a language.

Introduction

The present study began with the observation that many operations, e.g., intersection and quotient, upon pairs of languages (equal context free or Algol-like languages) do not yield a language, whereas these same operations performed upon a pair consisting of a language and a regular set yield a language. In seeking some general reason behind this situation we discovered that these operations could be realized using a device with two input lines and one output line. This in turn led to an investigation of languages and two-tape devices (i.e., devices with two input lines), the results of which constitute this paper.

In Section 2 we show (Theorem 2.1) that a two-tape machine (with one output line) transforms a pair consisting of a language and a regular set into a language. This fact resolves the problem initiating the study. We also show that a partial inverse to this machine operation also has the same property with respect to a language and a regular set. These results are extended to nondeterministic two-tape machines in Section 3.

Section 4 concerns some relationships between two-tape automata and two-tape machines. Thus it is proved (Theorem 4.1) that if R is a regular set and M a two-tape machine, then $M^{-1}(R)$ is the set of tapes accepted by some two-tape automaton. A partial converse to this is the result (Theorem 4.2) that for each two-tape automaton A there exists a two-tape machine M_A and a regular set R_A such that $M_A(T_2(A)) = R_A$ and $T_2(A) = M_A^{-1}(R_A)$.

Section 5 discusses a modified form of the devices previously considered. One particular result is that for each generalized sequential machine M, there exists a modified two-tape automaton A such that for arbitrary pairs of words (w_1, w_2), $M(w_1) = w_2$ if and only if (w_1, w_2) is in $T_2(A)$.

While we restrict ourselves to two-tape devices it will be abundantly clear to the reader that many of our results generalize to n-tape devices, $n > 2$. (In a two-tape machine one of the given sets is regular and the remaining is a language. In an n-tape machine all but one of the given sets is to be regular and the remaining a language.)

Presented at the 5th Annual Symposium on Switching Theory and Logical Design, Princeton University, November, 1964.

Throughout the paper we assume a slight knowledge of both machine theory and context free language theory. A brief resume is given in Section 1. Additional details and facts appear in [1, 7, 9].

1. *Preliminaries*

We are interested in certain devices and certain sets of words defined in various ways. In this section we recall two well-known devices, namely, "generalized sequential machines" and "automata." We then review two familiar kinds of sets of words, namely, "regular sets "and "context free languages." Our interest in context free languages arises from the fact that they coincide with the constituent parts in the ALGOL-like programming languages [6]. Thus deriving theorems about the effect of various devices upon context free languages is an attempt to understand how computers interact with programming languages. Results on this interaction also are useful in studying the structure of these languages. Our concern with regular sets arises from the facts that they enable us to obtain fact about languages.

Notation. For each set E let $\theta(E)$ be the free semigroup with identity ϵ generated by E. Thus $\theta(E)$ is the set of all finite sequences over E, or words, and ϵ is the empty word. For each word w let $|w|$ denote its length.

Definition. A *generalized sequential machine* (gsm) is a 6-tuple $M = (K, \Sigma, \Delta, s_0, \delta, \lambda)$, where (i) K, Σ, Δ are finite nonempty sets (of "states," "inputs," and "outputs" respectively); (ii) δ (the "next state function") and λ (the "output function") are mappings from $K \times \Sigma$ into K and $K \times \Sigma$ into $\theta(\Delta)$ respectively; (iii) s_0 is a distinguished element of K (the "start" state).

The elements of $\theta(\Sigma)$ and $\theta(\Delta)$ are called *input tapes* and *output tapes* respectively.

The functions δ and λ are extended by induction so that for any state s, input tape x, and input y, (i) $\delta(s, \epsilon) = s$ and $\lambda(s, \epsilon) = \epsilon$; (ii) $\delta(s, xy) = \delta[\delta(s, x), y]$ and $\lambda(s, xy) = \lambda(s, x)\lambda[\delta(s, x), y]$.[1]

For each gsm M and each input tape x, let $M(x) = \lambda(s_0, x)$. The operation so defined is called the *machine mapping M*.

The purpose of a gsm is as a transformation device, i.e., to transform input tapes into output tapes.

An *automaton* is a 5-tuple $A = (K, \Sigma, \delta, s_0, F)$, where (i) K, Σ are finite nonempty sets; (ii) δ is a function from $K \times \Sigma$ into K; (iii) s_0 is a distinguished element of K (the "start" state); (iv) F is a subset of K (the set of "final" states).

The purpose of an automaton is to select a set of words in the manner next indicated.

A set U is said to be *regular* if there exists some automaton A such that $U = \{w \text{ in } \Sigma/\delta(s_0, w) \text{ in } F\}$.

The family of regular sets is the smallest family of words containing the finite sets and closed under the operations of $\cup$, $\cdot$,[2] and[3] $*$ [10].

Regular sets are one kind of sets of words of interest to us. Another kind is now defined.

[1] For any device, every time δ is given as a function from $K \times \Sigma$ into K or λ from $K \times \Sigma$ into $\theta(\Delta)$, it is to be understood that δ or λ is extended to $K \times \theta(\Sigma)$ in the above manner. δ is always called the "next state function" and λ the "output function."

[2] For U and V sets of words, $U \cdot V = \{uv/u \text{ in } U, v \text{ in } V\}$, where uv is the concatenation of u and v. It is customary to write $U \cdot V$ as UV.

[3] For each set U of words, $U^* = \bigcup_{i=0}^{\infty} U^i$ where $U^0 = \epsilon$ and $U^{i+1} = U^iU$ for $i \geq 0$.

A *grammar* G is a 4-tuple (V, Σ, P, σ), where V is a finite set, Σ is a subset of V, σ is a distinguished element of V-Σ, and P is a finite set of ordered pairs of the form (ξ, w) with ξ in V-Σ and w in $\theta(V)$. P is called the set of *productions* of G. We write $\xi \rightarrow w$ when (ξ, w) is in P. For x, y in $\theta(V)$ we write $x \Rightarrow y$ if either $x = y$ or there exist w_1, w_2, w_3 in $\theta(V)$ and ξ in V such that $x = w_1\xi w_3$, $y = w_1w_2w_3$ and $\xi \rightarrow w_2$. For x, y in $\theta(V)$ we write $x \Rightarrow^* y$ if either $x \Rightarrow y$ or there exist $x_1, \cdots, x_k$ such that $x = x_1$, $x_k = y$ and $x_i \Rightarrow x_{i+1}$ for each i. In this case we write

$$(*) \quad x_1 \Rightarrow x_2 \Rightarrow \cdots \Rightarrow x_k$$

and call $(*)$ a *derivation* or *generation* of x_k (from x_1). The *language generated by* G, denoted by $L(G)$, is the set of words $\{w/\sigma \Rightarrow^* w, w \text{ in } \theta(\Sigma)\}$. A *context free language* (abbreviated *language*) over Σ is a language $L(G)$ generated by some grammar $G = (V, \Sigma, P, \sigma)$.

We assume an elementary knowledge about languages such as the fact that a regular set is a language or that the union and product of a pair of languages are languages. Of special note are the following results.

THEOREM 1.1. *If L is a language and R is a regular set, then $L \cap R$ is a language* [1].

THEOREM 1.2. *If L is a language (regular set) and M is a* gsm, *then*[4] *$M(L)$ and $M^{-1}(L) = \{w/M(w)$ in $L\}$ are languages (regular sets)* [7].

2. *Two-Tape Machines*

In this section we introduce the notion of a two-tape machine. Then we examine the image of the mapping defined by a two-tape machine upon pairs of languages.

We first consider an auxiliary concept.

Definition. A *shuffle* of words x, y is a word z such that x, y are mutually disjoint subsequences of z which exhaust z. That is, let $x = x_1 \cdots x_k$ and $y = y_1 \cdots y_m$, each x_i and y_j in Σ.[5] Then $z = z_1 \cdots z_{k+m}$, each z_i in Σ, is a shuffle of x, y if there exist subsets $\{f(1), \cdots, f(k)\}$ and $\{g(1), \cdots, g(m)\}$ of $\{1, \cdots, k + m\}$ such that $\{f(1), \cdots, f(k), g(1), \cdots, g(m)\} = \{1, \cdots, k + m\}$, $x_1 \cdots x_k = z_{f(1)} \cdots z_{f(k)}$ and $y_1 \cdots y_m = z_{g(1)} \cdots z_{g(m)}$.

The words xy and yx are shuffles of x, y. Similarly the word formed by alternating the letters of x with the letters of y until one of the words is exhausted and then adjoining at the end the terminal subword left on the other word is a shuffle of x, y. That is, if $x = x_1 \cdots x_k$ and $y = y_1 \cdots y_m$, each x_i, y_j in Σ, then $x_1y_1 \cdots x_ky_ky_{k+1} \cdots y_m$ if $k \leq m$ and $x_1y_1 \cdots y_mx_{m+1} \cdots x_k$ if $k > m$ is a shuffle of x, y.

We now present the definition of a two-tape machine.

Definition. A *two-tape machine* is an 8-tuple $M = (K, \Sigma, \Delta, \delta, \lambda, s_0, S_1, S_2)$ where (i) K, Σ, Δ, are finite nonempty sets (of "states," "inputs," and "outputs" respectively); (ii) δ and λ are mappings from $K \times \Sigma$ into K and $K \times \Sigma$ into $\theta(\Delta)$ respectively; (iii) s_0 is a distinguished element of K (the "start" state); (iv) $S_1 \cup S_2 = K$ and $S_1 \cap S_2$ is empty.

As may be seen below, the function of the sets S_1, S_2 is to determine which of the two tapes is currently being scanned. For each i, if the present state is an element of S_i then the ith tape is being scanned.

A two-tape machine transforms pairs of input tapes into output tapes as follows.

[4] If f is a function and E is a set, then $f(E) = \{f(x)/x \text{ in } E\}$.
[5] As usual, $x = \epsilon$ if $k = 0$ and $y = \epsilon$ if $m = 0$.

Definition. Let $(x_{11} \cdots x_{1k}, x_{21} \cdots x_{2m})$ be an element in $\theta(\Sigma) \times \theta(\Sigma)$ and let $M = (K, \Sigma, \Delta, \delta, \lambda, s_0, S_1, S_2)$ be a two-tape machine. Let $z_0 = x_{f(0)1}$ where s_0 is in $S_{f(0)}$. Let $s_1 = \delta(s_0, z_0)$. Continuing by induction suppose that the states $s_0, \cdots, s_n$ have been defined and $x_{11}, \cdots, x_{1r}$ and $x_{21}, \cdots, x_{2t}$ have been scanned, with $n = r + t$. Let s_n be in $S_{f(n)}$. If $f(n) = 1$ then the machine scans $z_n = x_{1(r+1)}$ if $x_{1(r+1)}$ exists, and stops otherwise with $x_{2(t+1)} \cdots x_{2m}$ unscanned. If $f(n) = 2$ then the machine scans $z_n = x_{2(t+1)}$ if $x_{2(t+1)}$ exists, and stops otherwise with $x_{1(r+1)} \cdots x_{1k}$ unscanned. Let $s_{n+1} = \delta(s_n, z_n)$ if z_n exists. Let $z_0, \cdots, z_u$ be all the symbols defined by this process. Let

$$M(x, y) = \lambda(s_0, z_0)\lambda(s_1, z_1) \cdots \lambda(s_u, z_u).$$

The mapping M of $\theta(\Sigma) \times \theta(\Sigma)$ into $\theta(\Delta)$ so defined is called the *machine mapping*.

Note that $z_0 \cdots z_u z_1' \cdots z_v'$, where $z_1', \cdots, z_v'$ are the unscanned symbols on the ith tape when the machine M tried unsuccessfully to scan a symbol on the jth tape, $i \neq j$, is a shuffle.

Notation. Let M be a two-tape machine. For U and V subsets of $\theta(\Sigma)$ let

$$M(U, V) = \{M(w_1, w_2)/w_1 \text{ in } U, w_2 \text{ in } V\}.$$

For $V \subseteq \theta(\Sigma)$ and $W \subseteq \theta(\Delta)$ let

$$\bar{M}^{-1}(V, W) = \{w_1 \text{ in } \theta(\Sigma)/M(w_1, w_2) \text{ in } W \text{ for some } w_2 \text{ in } V\}.$$

We now come to the main result of this section. This result is the analogue for two-tape machines of Theorems 1.2 for gsm.

THEOREM 2.1. *For each two-tape machine M the functions M and $\bar{M}^{-1}$ map (i) pairs of regular sets into regular sets, and (ii) pairs composed of one regular set and one language into a language.*

PROOF. Let $M = (K, \Sigma, \Delta, \delta, \lambda, s_0, S_1, S_2)$. Let $G = (V, \bar{\Sigma}, P, \sigma)$ be the grammar defined as follows. V-$\bar{\Sigma}$ consists of σ and all pairs (s, x), where s is in K and x is in $\Sigma \cup \{\epsilon\}$. $\bar{\Sigma}$ consists of all elements $\overline{(s, x)}$ for s in K and x in Σ, and all elements $\bar{x}$, $\bar{\bar{x}}$ for each x in Σ. The productions are of four kinds.

(a) $\sigma \to (s_0, x)$ x in $\Sigma \cup \{\epsilon\}$.

(b) If $x \neq \epsilon$, $(s, x) \to \overline{(s, x)}(\delta(s, x), x')$ for all x' in $\Sigma \cup \{\epsilon\}$.

(c) $\begin{cases} (s, \epsilon) \to \bar{x}(s, \epsilon) \text{ for } s \text{ in } S_2 \text{ and } x \text{ in } \Sigma. \\ (s, \epsilon) \to \bar{\bar{x}}(s, \epsilon(\text{ for } s \text{ in } S_1 \text{ and } x \text{ in } \Sigma. \end{cases}$

(d) $(s, \epsilon) \to \epsilon$.

Since each production is linear on the right,[6] $L(G)$ is a regular set [2]. Let ψ_1, ψ_2 be the *homomorphisms*[7] from $\theta(\bar{\Sigma})$ to $\theta(\Sigma)$ defined by $\psi_1(\bar{x}) = x$, $\psi_1(\bar{\bar{x}}) = \epsilon$, $\psi_2(\bar{x}) = \epsilon$, $\psi_2(\bar{\bar{x}}) = x$, $\psi_i(\overline{(s, x)}) = \epsilon$ if s is not in S_i, and $\psi_i(\overline{s, x})) = x$ if s in S_i, $i = 1, 2$. Since ψ_1, ψ_2 can be effected by a (one-state) gsm, both[8] ψ_1^{-1} and ψ_2^{-1} preserve regular sets and languages.

Now any derivation of a word w in $L(G)$ is obtained by using one production of type (a), $n + 1 \geq 0$ productions of type (b), $n + 1 \geq 0$ productions of type (c), and one production of type (d). The word w in $\theta(\bar{\Sigma})$ has the form

$$w = \overline{(s_0, z_0)} \cdots \overline{(s_n, z_n)} y_0 \cdots y_m,$$

[6] A production is *linear on the right* if it is of the form $\xi \to w\nu$ or $\xi \to w$, w in $\theta(\bar{\Sigma})$, ν in $V - \bar{\Sigma}$.

[7] A function f from $\theta(\bar{\Sigma})$ to $\theta(\Sigma)$ is said to be a *homomorphism* if $f(xy) = f(x)f(y)$ for all x, y. Each mapping f from $\bar{\Sigma}$ to $\theta(\Sigma)$ can be extended to be a homomorphism from $\theta(\bar{\Sigma})$ into $\theta(\Sigma)$ by letting $f(\epsilon) = \epsilon$ and $f(x_1 \cdots x_k) = f(x_1) \cdots f(x_k)$.

[8] If f is a mapping of $\theta(A)$ into $\theta(B)$ and $U \subseteq \theta(B)$, then $f^{-1}(U) = \{x/f(x) \text{ in } U\}$.

where $s_i = (\delta(s_{i-1}, z_i)$ for $1 \leq i \leq n$, $y_i = \bar{v}$ or $\bar{\bar{v}}_i$ (v_i in Σ) for each i, and $y_0 \ldots y_m$ is either of the form $\bar{v}_0 \cdots \bar{v}_m$ or $\bar{\bar{v}}_0 \cdots \bar{\bar{v}}_m$ depending on whether $\delta(s_n, z_n)$ is in S_2 or in S_1. If $n = -1$, there are no elements of the form $(\overline{s_i, z_i})$. If $m = -1$, there are no elements of the form y_j. (Thus $w = \epsilon$ if $n = -1$ and $m = -1$.)

From the definition of G, it is readily seen that for

$$w = (\overline{s_0, z_0}) \cdots (\overline{s_n, z_n}) y_0 \cdots y_m$$

in $L(G)$, if $(\psi_1(w), \psi_2(w))$ is used as a pair of input tapes for the machine M, then the successive states are $s_0, s_1, \cdots, s_n$, the successive inputs scanned are $z_0, \cdots, z_n$, and the unscanned terminal subword left on one of the tapes is $y_0 \cdots y_m$. Also, given any pair (w_1, w_2) in $\theta(\Sigma) \times \theta(\Sigma)$ there is a word w in $L(G)$ such that $w_1 = \psi_1(w)$ and $w_2 = \psi_2(w)$. (Merely define s_i inductively by $s_i = \delta(s_{i-1}, z_{i-1})$ where $z_0 \cdots z_n z_0' \cdots z_m'$ is the shuffle of w_1 and w_2 determined by the machine M and let $y_0 \cdots y_m = \bar{z}_1' \cdots \bar{z}_m'$ or $y_0 \cdots y_m = \bar{\bar{z}}_0' \cdots \bar{\bar{z}}_m'$ depending on whether $\delta(s_n, z_n)$ is in S_2 or in S_1.)

Let τ be the homomorphism from $\theta(\bar{\Sigma})$ to $\theta(\Delta)$ defined by $\tau(\bar{x}) = \epsilon$, $\tau(\bar{\bar{x}}) = \epsilon$, and $\tau((\overline{s, x})) = \lambda(s, x)$. Since τ is realized by a (one-state) gsm, both τ and τ^{-1} preserve regular sets and languages.

(i) For U and V subsets of $\theta(\Sigma)$ it follows from the comments above that

$$M(U, V) = \tau(L(G) \cap \psi_1^{-1}(U) \cap \psi_2^{-1}(V)).$$

Suppose U and V are regular. Being the intersection of regular sets, $L(G) \cap \psi_1^{-1}(U) \cap \psi_2^{-1}(V)$ is regular. Since τ preserves regular sets, $M(U, V)$ is regular. Suppose U or V is regular and the other a language. Being the intersection of a language and a regular set, $\psi_1^{-1}(U) \cap \psi_2^{-1}(V)$ is a language. Hence $L(G) \cap \psi_1^{-1}(U) \cap \psi_2^{-1}(V)$ is a language. Since τ preserves languages, $M(U, V)$ is a language.

(ii) For $V \subseteq \theta(\Sigma)$ and $W \subseteq \theta(\Delta)$ it is easily seen that

$$\bar{M}^{-1}(V, W) = \psi_1(L(G) \cap \tau^{-1}(W) \cap \psi_2^{-1}(V)).$$

If V and W are both regular, then so is $\bar{M}^{-1}(V, W)$ (for ψ_1, ψ_2^{-1} and τ^{-1} preserve regular sets). If one of the sets V, W is regular and the other is a language, then $L(G) \cap \tau^{-1}(W) \cap \psi_2^{-1}(V)$ is a language (for τ^{-1}, ψ_2^{-1} both preserve regular sets and languages). Since ψ_1 preserves languages, $\bar{M}^{-1}(V, W)$ is a language.

Corollary. *For each pair (U, V), one regular and the other a language (both regular), the set of all words obtained by shuffling alternately words of U with words of V is also a language (regular set).*

Proof. Let α be a special symbol not in Σ. Then $(U\alpha, V\alpha)$ is a pair consisting of one regular set and one language (both regular). Let M be the 4-state two-tape machine defined by $S_1 = \{s_0, s_2\}$, $S_2 = \{s_1, s_3\}$, $\delta(s_0, x) = s_1$ and $\delta(s_1, x) = s_0$ for $x \neq \alpha$, $\delta(s_0, \alpha) = s_3$, $\delta(s_1, \alpha) = s_2$, $\delta(s_2, x) = s_2$ and $\delta(s_3, x) = s_3$ for any x, $\lambda(s, x) = x$ for $x \neq \alpha$ and arbitrary s, and $\lambda(s, \alpha) = \epsilon$ for arbitrary s. Then the set of alternate shuffles is $M(U\alpha, V\alpha)$ and is a language (regular set) by Theorem 2.1.

Remarks. (1) For a two-tape machine M and pairs (V, W), one regular and the other a language, it is not necessarily true that $\{w/M(w, V) \subseteq W\}$ is a language. First consider the case where V is regular and W is a language. Let $V = \{c, d\}$ and $W = \{ca^ib^ie^j/i, j \geq 1\} \cup \{da^ib^je^j/i, j \geqq 1\}$. Let M be a two-tape machine which (i) scans the first symbol on the second tape and thereafter scans just the first tape, (ii) from either state yields the input as output. Then $\{w/M(w, V) \subseteq W\} = \{a^ib^ie^i/i \geq 1\}$, which is not a language. Next consider the case where V is a language and W regular. For $V = \{a^i b^i d/i \geq 1\} \cup \{b^j c^j d/j \geq 1\}$ and $W = \{d\}$, there exists a two-tape machine such that $\{w/M(w, V) \subseteq W\} = \{a^i b^i c^i/i \geq 1\}$, which is not a language.

(2) For M a two-tape machine and U, V both languages, it is not always true that $M(U, V)$ is a language. To see this observe that if it were always true, then the corollary to Theorem 2.1 could be extended to the case when both U and V are languages. In general, however, the set of alternate shuffles of a pair of languages need not be a language. For example, the languages $\{a^i b^i/i \geq 1\}$ and $\{c^{2j} d^j/j \geq 1\}$ yield a set X of alternate shuffles which is not a language. For suppose X is a language. Then the subset Y of words in X having a terminal subword of the form $c\, d^k$, $k \geq 1$, being the intersection of X and the regular set $\theta(a, b, c, d)c\, d\, d^*$, is also a language. However

$$Y = \{(ac)^i(bc)^i c^{2j-2i} d^j/1 \leq i \leq j\},$$

which is not a language. For if Y is a language, since there is a gsm which maps Y onto $Z = \{a^i b^i d^j/1 \leq i \leq j\}$, Z is a language. But Z is not a language.

As another illustration of Theorem 2.1 consider the following. For each (w_1, w_2) in $\theta(\Sigma) \times \theta(\Sigma)$ let (i) $T(w_1, w_2) = w_1$ if $|w_1| > |w_2|$, (ii) $T(w_1, w_2) = w_1 w_4$ if $|w_1| \leq |w_2|$, where $w_2 = w_3 w_4$ and $|w_3| = |w_1|$. Suppose (U, V) is a pair consisting of one regular set and one language. Then $(U\alpha, V\alpha)$ is also such a pair, α being a symbol not in Σ. It is easy to construct a two-tape machine M such that $M(w_1\alpha, w_2\alpha) = T(w_1, w_2)$. Thus $T(U, V)$ is a language. Similarly $T(U, V)$ is regular if both U and V are regular.

With the help of Theorem 2.1 the following result can also be shown. For a given positive integer k and a pair (U, V), one a language and one regular (both regular), the set of all words y in U for which there exists some word z in V such that $|y| = k|z|$ is a language.

3. *Nondeterministic Machines*

We now investigate the mapping defined by a nondeterministic two-tape machine. We shall see that a nondeterministic two-tape machine, like a (deterministic) two-tape machine, maps a pair of sets, one regular and the other a language, into a language.

To simplify the presentation we consider the nondeterministic gsm and then generalize to the nondeterministic two-tape machine.

Definition. A *nondeterministic gsm* is a 5-tuple $M = (K, \Sigma, \Delta, \mu, s_0)$ where (i) K, Σ, Δ are finite nonempty sets; (ii) μ is a function from $K \times \Sigma$ to the *finite* nonempty subsets of $K \times \theta(\Delta)$; and (iii) s_0 is a distinguished element of K (the "start" state).[9]

For s in K and x in Σ, $\mu(s, x)$ is to be interpreted as the set of pairs of next states and output tapes of the machine at state s under input x.

The machine is a gsm if $\mu(s, x)$ consists of a single element for all s, x.

For each input tape w there is a set of output tapes determined (in the obvious way) by the start state and μ. Thus the nondeterministic machine M defines a

[9] The notion of nondeterministic gsm can be extended to the case where there is more than one "start" state, i.e., where $M = (K, \Sigma, \Delta, \mu, S_0)$, with S_0 a nonempty subset of K. Under this generalization, Theorem 3.1 still holds.

function, also denoted by M, from subsets of $\theta(\Sigma)$ to subsets of $\theta(\Delta)$ which assigns to each subset U of $\theta(\Sigma)$ the set $M(U)$ of all possible output tapes for input tapes in U. Similarly there is an inverse mapping M^{-1} from subsets of $\theta(\Delta)$ to subsets of $\theta(\Sigma)$ which assigns to each subset V of $\theta(\Delta)$ the set $M^{-1}(V)$ of all input tapes having *at least one* possible output tape in V.

We shall show that M and M^{-1} both preserve regular sets and languages by obtaining a gsm closely related to the given nondeterministic gsm.

Definition. Let M be a nondeterministic gsm. For a fixed element x of Σ a *determinism* (with respect to M) is a function d_x from K to $K \times \theta(\Delta)$ such that $d_x(s)$ is in $\mu(s, x)$ for each s in K.

Let D_x denote the set of all determinisms for x. Thus D_x is a finite set whose power is the product $\prod_{s \text{ in } S} n_{s,x}$, where $n_{s,x}$ is the number of elements in $\mu(s, x)$.

Consider the gsm $M_1 = (K, \bar{\Sigma}, \Delta, \delta, \lambda, s_0)$ where $\bar{\Sigma} = \bigcup_{x \text{ in } \Sigma} D_x$ and δ, λ are defined by

$$(\delta(s, d_x), \lambda(s, d_x)) = d_x(s).$$

The machine mapping M_1 and its inverse M_1^{-1} both preserve regular sets and languages. The relation between M, M^{-1}, M_1^{-1} is expressed by

LEMMA 3.1. *Let ν be the substitution mapping from $\theta(\Sigma)$ to subsets of*[10] *$\theta(\bar{\Sigma})$ defined by $\nu(x) = D_x$, x in Σ, and let N be the (one-state) gsm which maps each element of D_x into x. Then*

$$M = M_1\nu \quad \text{and} \quad M^{-1} = NM_1^{-1}.$$

PROOF. It suffices to observe that M_1 is related to M in such a way that the possible pairs of next states and output tapes at state s under input x in M coincides with the set of pairs of next states and output tapes at state s of M_1 under an input in D_x.

Since M_1, M_1^{-1}, ν, and τ preserve regular sets and languages we get Theorem 3.1.

THEOREM 3.1. *For a nondeterministic machine, both the machine mapping and its inverse preserve regular sets and languages.*

Remark. The statement of Theorem 3.1 for the machine mapping appears in [3, p. 191] where it is credited as being proved in [7]. However, the machine mapping result proved in [7] is for deterministic machines, not nondeterministic ones. The machine mapping theorem for a nondeterministic machine is proved (by a different method) in [4, Theorem 3.1].

It is a straightforward procedure to generalize the notion of a nondeterministic two-tape machine from a (deterministic) two-tape machine. Formally we have the following definition.

Definition. A *nondeterministic two-tape machine* is a 7-tuple $M = (K, \Sigma, \Delta, \mu, s_0, S_1, S_2)$ where (i) K, Σ, Δ are finite nonempty sets; (ii) μ is a function from $K \times \Sigma$ to the finite nonempty subsets of $K \times \theta(\Delta)$; (iii) s_0 is a distinguished element of K; (iv) $S_1 \cup S_2 = K$ and $S_1 \cap S_2$ is empty.

The same considerations given above in replacing a nondeterministic gsm by a (deterministic) gsm occur in replacing a nondeterministic two-tape machine by a (deterministic) two-tape machine. Hence there follows:

THEOREM 3.2. *For a nondeterministic two-tape machine let M be the machine map-*

[10] Let ν be a function from $\theta(\Sigma)$ to subsets of $\theta(\bar{\Sigma})$. If $\nu(\epsilon) = \epsilon$ and $\nu(x_1 \cdots x_k) = \nu(x_1) \cdots \nu(x_k)$, each x_i in Σ, then ν is called a *substitution* mapping. Each substitution mapping ν such that $\nu(x)$ is finite for x in Σ preserves regular sets and languages [1].

ping of $\theta(\Sigma) \times \theta(\Sigma)$ into $\theta(\Delta)$ and let $\bar{M}^{-1}$ be the mapping of $\theta(\Sigma) \times \theta(\Delta)$ into subsets of $\theta(\Sigma)$ defined by

$$\bar{M}^{-1}(w_2, w) = \{w_1/w \text{ is in } M(w_1, w_2)\}.$$

Then both M and $\bar{M}^{-1}$ maps pairs of regular sets into regular sets and pairs composed of a regular set and a language into a language.

COROLLARY 1. *If U is a language (regular set) and R is a regular set, then the set of all shuffles of each word of U with each word of R is a language (regular set).*

PROOF. Let M be the nondeterministic two-tape machine for which $S_1 = \{s_0\}$, $S_2 = \{s_1\}$ and $\mu(s_0, x) = \mu(s_1, x) = \{(s_0, x), (s_1, x)\}$ for any input x. The result then follows from the fact that the set of all shuffles is $M(U, R)$.

Remark. The set of all shuffles of each word of a language with each word of another language is not, in general, a language. For example, let W be the set of all shuffles of each word of the language $\{a^i b^i/i \geq 1\}$ with each word of the language $\{c^{2j} d^j/j \geq 1\}$. Suppose W is a language. Then the intersection of W with the regular set $(ac)^*(bc)^*c^* d^*$ is also a language. However, this intersection is

$$\{(ac)^i(bc)^i c^{2j-2i} d^j/1 \leq i \leq j\},$$

which, as noted earlier, is not a language.

The fact that the quotient of a language by a regular set is a language [8] can easily be derived from Theorem 3.2. We consider the following "double quotient."

COROLLARY 2. *Let U be a language (regular set) and R a regular set. Then the set of all words w for which uwv is in U for some word uv in R is a language (regular set).*

PROOF. Let α be a special symbol not occurring in Σ. Let M be the nondeterministic two-tape machine with $S_2 = \{s_0, s_2\}$, $S_1 = \{s_1, s_3\}$, $\mu(s_0, x) = \{(s_0, x), (s_1, x)$ if $x \neq \alpha$, $\mu(s_0, \alpha) = \{(s_3, \epsilon)\}$, $\mu(s_1, x) = \{(s_1, x)\}$ if $x \neq \alpha$, $\mu(s_1, \alpha) = \{(s_2, \epsilon)\}$, $\mu(s_2, x) = \{(s_2, x)\}$ for all x, and $\mu(s_3, x) = \{(s_3, x)\}$ for all inputs x. Since $R\alpha$ is regular and $U\alpha$ is a language (regular), $\bar{M}^{-1}(R\alpha, U\alpha)$ is a language (regular set). Therefore

$$\bar{M}^{-1}(R\alpha, U\alpha) \cap \theta(\Sigma)\alpha = \{w\alpha \text{ in } \bar{M}^{-1}(R\alpha, U\alpha)/w \text{ in } \theta(\Sigma)\}$$

is a language (regular set). Thus $\{w \text{ in } \theta(\Sigma)/w\alpha \text{ in } \bar{M}^{-1}(R\alpha, U\alpha)\}$, which is the desired set, is a language (regular set).

With the help of Theorem 3.2 the following result is readily seen. Let k be a positive integer and (U, V) a pair consisting of one language and one regular set (both regular). Then the set

$$\{u \text{ in } U/|u| \leq |v| \leq k|u| \text{ for some } v \text{ in } V\}$$

is a language (regular).

4. *Two-Tape Automata*

We now consider two-tape automata. In [10] two kinds of two-tape automata are discussed, the first having special end-of-tape symbols on its input tapes and the second having no special end-of-tape symbol.[11] We concentrate on the latter since

[11] The end-of-tape symbol appears in the definition of two-tape automata as given in [10]. However in the proof of Theorem 16 of [10], the authors revert to input tapes without end-of-tape symbols.

they bear the same relationship to (one-tape) automata as two-tape machines bear to gsm. We briefly discuss the former in Section 5.

Let us recall some elementary concepts about two-tape automata.

Definition. A *two-tape automaton* is a 7-tuple $A = (K, \Sigma, \delta, s_0, F, S_1, S_2)$ where (i) K and Σ are finite nonempty sets; (ii) δ is a mapping from $K \times \Sigma$ into K; (iii) s_0 is a distinguished element of K; (iv) F is a subset of K (the set of "final" states); (v) $S_1 \cup S_2 = K$ and $S_1 \cap S_2$ is empty.

The purpose of a two-tape automaton is to "accept" certain pairs of tapes as follows.

Definition. Let $(x_{11} \cdots x_{1k}, x_{21} \cdots x_{2m})$ be an arbitrary pair of tapes in $\theta(\Sigma) \times \theta(\Sigma)$ and let $A = (K, \Sigma, \delta, s_0, F, S_1, S_2)$ be a two-tape automaton. Let $z_0 = x_{f(0)1}$ where s_0 is in $S_{f(0)}$. Let $s_1 = \delta(s_0, z_0)$. Continuing by induction, suppose the states $s_0, \cdots, s_n$ have been defined and $x_{11}, \cdots, x_{1r}$ and $x_{21}, \cdots, x_{2t}$ have been scanned, with $n = r + t$. Let s_n be in $S_{f(n)}$. If $f(n) = 1$ then the automaton scans $z_n = x_{1(r+1)}$ if $x_{1(r+1)}$ exists, and stops otherwise with $x_{2(t+1)} \cdots x_{2m}$ unscanned. If $f(n) = 2$ then the automaton scans $z_n = x_{2(t+1)}$ if $x_{2(t+1)}$ exists, and stops otherwise with $x_{1(r+1)} \cdots x_{1k}$ unscanned. The pair $(x_{11} \cdots x_{1k}, x_{21} \cdots x_{2m})$ is said to be *accepted* by A if s_{u+1} is an element of F.

Notation. The set of all pairs accepted by a two-tape automaton A is denoted by $T_2(A)$.

Our interest in two-tape automata stems from the next result.

THEOREM 4.1. *Let M be a two-tape machine and R a regular subset of $\theta(\Delta)$. Then $M^{-1}(R) = T_2(A)$ for some two-tape automaton A.*

PROOF. Let $M = (K, \Sigma, \Delta, \delta, \lambda, s_0, S_1, S_2)$. Since R is regular, there exists some automaton $(K', \Delta, \delta', s_0', F')$ such that $R = \{w \text{ in } \theta(\Delta)/\delta'(s_0', w) \text{ in } F'\}$. Consider the two-tape automaton

$$A = (K \times K', \Sigma, \delta'', (s_0, s_0'), K \times F', S_1 \times K', S_2 \times K'),$$

where $\delta''(s, s'), x) = (\delta(s, x), \delta'(s', \lambda(s, x)))$. We shall show that $M^{-1}(R) = T_2(A)$.

Suppose (w_1, w_2) is in $T_2(A)$. Let $z_0, \cdots, z_n$ be the successive inputs scanned and $(s_0, s_0'), (s_1, s_1'), \cdots, (s_n, s_n')$ the successive states. Then $s_{i+1} = \delta(s_i, z_i)$ and $s_{i+1}' = \delta'(s_i', \lambda(s_i, z_i))$ for $0 \leq i \leq n$. For the element (w_1, w_2) in $\theta(\Sigma) \times \theta(\Sigma)$, the two-tape machine scans the inputs $z_0, z_1, \cdots, z_n$ in that order and moves successively through the states $s_0, \cdots, s_n$. Therefore

$$M(w_1, w_2) = \lambda(s_0, z_0) \cdots \lambda(s_n, z_n).$$

Since $s_{i+1}' = \delta'(s_i', \lambda(s_i, z_i))$ and (w_1, w_2) is in $T_2(A)$, s_{n+1}' is in F'. Thus $\lambda(s_0, z_0) \cdots \lambda(s_n, z_n)$ is in R, and (w_1, w_2) is in $M^{-1}(R)$.

Now assume that (w_1, w_2) is in $M^{-1}(R)$. Then the above process can be reversed to show that (w_1, w_2) is in $T_2(A)$, completing the proof.

Remark. The converse of Theorem 4.1 is, in general, false. In fact, for a two-tape automaton A, the image of $T_2(A)$ under a two-tape machine is not necessarily a language. For example, the set of pairs

$$Y = \{(a^i b^j c, d^i e^j f X)/i \geq 0, j \geq 0, X \text{ in } \theta(\Sigma)\}$$

is easily seen to be a set accepted by some two-tape automaton. Clearly there exists a two-tape machine which maps Y onto

$$Z = \{a^i b^j d^i e^j/i \geq 0, j \geq 0\}.$$

However Z is not a language.

We now present a partial converse to Theorem 4.1.

THEOREM 4.2. *For each two-tape automaton A there exists a two-tape machine M_A and a regular set R_A such that $M_A(T_2(A)) = R_A$ and $T_2(A) = M_A^{-1}(R_A)$.*

PROOF. Let $A = (K, \Sigma, \delta, s_0, F, S_1, S_2)$. Let $M_A = (K, \Sigma, \Sigma, \delta, \lambda, s_0, S_1, S_2)$, where $\lambda(s, x) = x$ for each s, x. Consider the (one-tape) automaton $A' = (K, \Sigma, \delta, s_0, F)$. Let $R_A = \{w \text{ in } \theta(\Sigma)/\delta(s_0, w) \text{ in } F\}$. It is easily verified that a pair (w_1, w_2) is accepted by A if and only if $M(w_1, w_2)$ is in A; and furthermore, that to each tape w in R_A there is a pair (w_1, w_2) in $T_2(A)$ such that $M(w_1, w_2) = w$. Thus M_A and R_A satisfy the theorem.

The following corollary generalizes exercise 4 on page 137 in [9].

COROLLARY. *Let A be a two-tape automaton and V a regular set (language). Then the set*

$$\{w_1/(w_1, w_2) \quad \text{in} \quad T_2(A) \quad \text{for some} \quad w_2 \quad \text{in} \quad V\}$$

is regular (a language).

PROOF. By Theorem 4.2, there is a two-tape machine M and a regular set R' such that $T_2(A) = M^{-1}(R')$. Now the desired set is $\bar{M}^{-1}(V, R')$. By Theorem 2.1, $\bar{M}^{-1}(V, R')$ is regular if V is regular and is a language if V is a language.

5. *Modified Devices.*

If (w_1, w_2) is a pair accepted by some two-tape automaton A, then either (w_1X, w_2) is also accepted by A for all X in $\theta(\Sigma)$ or (w_1, w_2X) is accepted by A for all X in $\theta(\Sigma)$. This condition implies some special properties on $T_2(A)$. For example, if $T_2(A)$ is nonempty then it is infinite. In order that the class of sets accepted be larger we eliminate some of these pairs by the addition of a special end-of-tape symbol.

Definition. A *modified two-tape automaton* (m-automaton), is an 8-tuple $A = (K, \Sigma, \alpha, \delta, s_0, F, S_1, S_2)$, where K, Σ, s_0, F, S_1, S_2 have their usual properties and δ is a mapping of $K \times (\Sigma \cup \{\alpha\})$ into K, α being a special symbol not in Σ.

Definition. The set $T_2(A)$ of all pairs *accepted* by an m-automaton A is the set of all pairs (w_1, w_2) in $\theta(\Sigma) \times \theta(\Sigma)$ such that $(w_1\alpha, w_2\alpha)$ is accepted by the two-tape automaton $(K, \Sigma \cup \{\alpha\}, \delta, s_0, F, S_1, S_2)$.

We modify the notion of a two-tape machine in an analogous fashion.

Definition. A *modified two-tape machine* (m-machine), is a 9-tuple $M = (K, \Sigma, \alpha, \Delta, \delta, \lambda, s_0, S_1, S_2)$, where K, Σ, Δ, s_0, F, S_1, S_2 have their usual properties, δ is a mapping of $K \times (\Sigma \cup \{\alpha\})$ into K, and λ is a mapping of $K \times (\Sigma \cup \{\alpha\})$ into $\theta(\Delta)$, α being a special symbol not in Σ.

Notation. For each m-automaton $A = (K, \Sigma, \alpha, \delta, s_0, F, S_1, S_2)$ let A_a be the two-tape automaton $(K, \Sigma \cup \{\alpha\}, \delta, s_0, F, S_1, S_2)$. For each m-machine $M = (K, \Sigma, \alpha, \Delta, \delta, \lambda, s_0, S_1, S_2)$ let M_a be the two-tape machine $(K, \Sigma \cup \{\alpha\}, \Delta, \delta, \lambda, s_0, S_1, S_2)$.

Notation. Given an m-machine M, for each (w_1, w_2) in $\theta(\Sigma) \times \theta(\Sigma)$ let $M(w_1, w_2) = M_a(w_1\alpha, w_2\alpha)$. Let $\bar{M}^{-1}$ be the mapping of $\theta(\Sigma) \times \theta(\Delta)$ into subsets of $\theta(\Sigma)$ which takes each (w_2, w) in $\theta(\Sigma) \times \theta(\Delta)$ into $\{w_1 \text{ in } \theta(\Sigma)/M(w_1, w_2) = w\}$.

The mappings M and $\bar{M}^{-1}$ for m-machines have the same properties as M and $\bar{M}^{-1}$ for two-tape machines in Theorem 2.1, namely,

THEOREM 5.1. *For each m-machine M, both of the functions M and $\bar{M}^{-1}$ map (i) pairs of regular sets into regular sets, and (ii) pairs composed of one regular set and one language into a language.*

PROOF. Let τ be the mapping of (w_1, w_2) in $\theta(\Sigma) \times \theta(\Sigma)$ into $(w_1\alpha, w_2\alpha)$. Then τ

maps pairs of regular sets into pairs of regular sets, and a pair consisting of a regular set and a language into a pair consisting of a regular set and a language. Since $M = M_a\tau$, M has the desired properties.

For $B \subseteq \theta(\Sigma)$ and $C \subseteq \theta(\Delta)$, $\bar{M}^{-1}(B, C) = \{w_1/w_1\alpha \text{ in } \bar{M}_a^{-1}(B\alpha, C)\}$. If B and C are both regular (or one is regular and the other a language), then the same is true for $B\alpha$ and C. Therefore $\bar{M}_a^{-1}(B\alpha, C)$ is regular (or a language) and the same is true for $\bar{M}^{-1}(B, C)$.

We may define a nondeterministic m-machine M in the obvious fashion and derive the fact that M and $\bar{M}^{-1}$ have the properties mentioned in Theorem 3.2.

The results in Section 4 also hold for m-automata and m-machines. Thus for an m-machine M and a regular set (or language) U contained in $\theta(\Delta)$, there exists an m-automaton A such that $M^{-1}(U) = \{(w_1, w_2) \text{ in } \theta(\Sigma) \times \theta(\Sigma)/(w_1\alpha, w_2\alpha) \text{ in } T_2(A)\}$.

Our final result is the following.

THEOREM 5.2. *For each gsm $M = (K, \Sigma, \Sigma, \delta, \lambda, s_0)$ there exists an m-automaton A such that for arbitrary (w_1, w_2) in $\theta(\Sigma) \times \theta(\Sigma)$, $M(w_1) = w_2$ if and only if (w_1, w_2) is in $T_2(A)$.*

PROOF. We define an m-automaton $A = (S_1 \cup S_2, \Sigma, \alpha, \delta', s_0, F, S_1, S_2)$ as follows. Let $S_1 = K \cup \{p_1\}$ and

$$S_2 = \{(s, x, i)/s \text{ in } K, x \text{ in } \Sigma, 1 \leq i \leq |\lambda(s, x)|\} \cup \{p_2\},$$

where p_1, p_2 are special symbols not in K. Let $F = \{p_2\}$. Let δ' be the function defined by

$$\delta'(s, x) = \begin{cases} (s, x, 1) \text{ if } |\lambda(s, x)| \geq 1 \\ \delta(s, x) \text{ if } |\lambda(s, x)| = 0, \text{ i.e., } \lambda(s, x) = \epsilon \end{cases}$$

$$\delta'(s, \alpha) = p_2$$

$$\delta'((s, x, i), y) = \begin{cases} (s, x, i + 1) \text{ if } |\lambda(s, x)| > 1 \text{ and } y \text{ is the } i\text{th letter of } \lambda(s, x) \\ \delta(s, x) \text{ if } |\lambda(s, x)| = i \text{ and } y \text{ is the } i\text{th letter of } \lambda(s, x) \end{cases}$$

$$\delta'(p_2, \alpha) = p_2$$

$$\delta' = p_1 \text{ otherwise.}$$

It is easily seen that $M(w_1) = w_2$ if and only if (w_1, w_2) is in $T_2(A)$.

Remarks. (1) Theorem 5.2 is no longer true if "m-automaton" is replaced by "two-tape automaton." For let M be the one-state gsm whose output coincides with its input. Let

$$Z = \{(w, M(w))/w \text{ in } \theta(\Sigma)\} = \{(w, w)/w \text{ in } \theta(\Sigma)\}.$$

Suppose A is a two-tape automaton such that $Z = T_2(A)$. Then for each (w, w), either $\{(wy, w)/y \text{ in } \theta(\Sigma)\}$ or $\{(w, wy)/y \text{ in } \theta(\Sigma)\}$ is a subset of $T_2(A) = Z$, a contradiction. Thus there is no two-tape automaton A such that $Z = T_2(A)$.

(2) It is an open problem to give a necessary and sufficient condition on $T_2(A)$, A an m-automaton, such that $T_2(A) = \{(w, M(w))/w \text{ in } \theta(\Sigma)\}$ for some gsm M. Related to this problem is the fact that Elgot [5, Theorem 7.1] has characterized those subsets Z of $\theta(\Sigma) \times \theta(\Sigma)$ for which there exists a complete sequential machine[12] such that $Z = \{(w, M(w)/w \text{ in } \theta(\Sigma)\}$.

[12] A *complete sequential machine* is a gsm whose output is the same length as its input.

REFERENCES

1. Bar-Hillel, Perles, and Shamir. On formal properties of simple phrase structure grammars. *Z. Phonetik, Sprachwiss. Kommun. 14* (1961), 143–172.
2. Chomsky, N. On certain formal properties of grammars. *Informat. Contr. 2* (1959), 137–167.
3. ——. Context-free grammars and pushdown storage. Quart. Progr. Rep. No. 65, Res. Lab. of Electron., MIT, Cambridge, Mass., 1962, 187–194.
4. ——. Formal properties of grammars, Chapter 12. In *Handbook of Mathematical Psychology, Vol. 2*, Ed. Luce, Bush, and Galanter, John Wiley, New York, 1963.
5. Elgot, C. Decision problems of finite automata design and related problems. *Trans. Amer. Math. Soc. 98* (1961), 22–51.
6. Ginsburg, S., and Rice, H. G. Two families of languages related to ALGOL. *J. ACM 9* (July 1962), 350–371.
7. ——, and Rose, G. F. Operations which preserve definability in languages. *J. ACM 10* (1963), 175–195.
8. ——, and Spanier, E. H. Quotients of context free languages. *J. ACM 10* (Oct. 1963), 487–492.
9. ——. *An Introduction to Mathematical Machine Theory.* Addison-Wesley, Cambridge, Mass., 1962.
10. Rabin, M., and Scott, D. Finite automata and their decision problems. *IBM J. Res. Develop. 3* (1959), 114–125.