The Left-Discrimination Sequence of an Automaton*

WILLIAM R. EDWARDS

Department of Computer Science, University of Southwestern Louisiana,
Lafayette, Louisiana 70504

AND

ZAMIR BAVEL

Department of Computer Science, University of Kansas, Lawrence, Kansas 66044

Left-discrimination of a semigroup is defined and shown to be a sufficient condition that a semigroup be isomorphic to the input semigroup of its semigroup automaton, a necessary condition if the semigroup is finite. The left-discrimination sequence of an automaton is defined as a sequence of semigroups beginning with the input semigroup of the automaton, each member being the input semigroup of the semigroup automaton of its predecessor. It is related directly to a particular monotonically decreasing sequence of subautomata of the original automaton. This sequence is shown to be preserved by homomorphisms and is extended and used in an algorithm for determining the homomorphisms on one finite automaton to another.

1. INTRODUCTION

The input semigroup $I_A$ of an automaton $A$ (the semigroup of input functions of $A$) has been of interest to many researchers in its own right and in relation to other structures and mappings of automata. A somewhat related concept, that of the semigroup automaton $A(J)$ of a semigroup $J$, has been of interest and utility (e.g., Deussen, 1966; Dragan, 1968; Edwards and Bavel, 1975b; and Ginzburg, 1968). It might be assumed that the input semigroup $I_{A(J)}$ of the semigroup automaton $A(J)$ of an arbitrary semigroup $J$ is isomorphic to $J$; i.e., $I_{A(J)} \cong J$. This need not be the case, as is shown in Fig. 1.

This naturally raises two questions: "Under what conditions is a semigroup $J$ isomorphic to $I_{A(J)}$?", and "Given an arbitrary semigroup $J$, is there always an automaton whose input semigroup is isomorphic to $J$?" These questions may be answered from basic semigroup theory once it is realized that $I_{A(J)}$ corresponds

* This work was supported in part by the National Science Foundation under Grant GJ-639.
to the regular representation of \( J \) by right translations on \( J \). For it is easy to show that the regular representation of \( J \) is faithful (a natural homomorphism of \( J \) into \( I_\mathcal{A}(J) \) is monic and epic) if and only if \( J \) is left-reductive. Furthermore, it is known that the extended regular representation of \( J \), which is \( I_{A'} \) for an automaton \( A' \) containing \( \mathcal{A}(J) \), is always faithful, answering the second question. (Definitions and a statement of these results may be found in Clifford and Preston (1961).) We shall use the term “left-discriminative” rather than “left-reductive,” since it is more suggestive from the automata-theoretic point of view.

Thus, it is easy to show that left-discrimination of \( J \) is a sufficient condition for the isomorphism \( I_\mathcal{A}(J) \cong J \), and, for finite \( J \), also a necessary condition. However, when \( J \) is not left-discriminative, \( I_\mathcal{A}(J) \) may not be isomorphic to \( J \), though it is a homomorphic image of \( J \). This fact gives rise to a sequence of semigroups associated with an automaton \( A \), the sequence of input semigroups of successive semigroup automata, called the left-discrimination sequence of \( A \). In the finite case, this sequence eventually reaches a left-discriminative member, isomorphic to all its successors. The length of the sequence to this point is called the left-discrimination characteristic of the automaton.

The left-discrimination sequence of an automaton \( A \) is related to the structure of \( A \) by the “source length” of a state of \( A \), which is the number of inputs in the longest input string leading to the state. The states of \( A \) of source length not less than a given nonnegative integer form a subautomaton of \( A \), and the set of such subautomata forms a nested sequence of subautomata, dividing \( A \) into “concentric shells.” The members of the left-discrimination sequence of \( A \) are shown to be the respective input semigroups of this nested sequence of subautomata. As a consequence, the left-discrimination characteristic of \( A \) is expressed in terms of the distinguishing power of the sequence of subautomata on input strings and, as a further consequence, necessary and sufficient conditions are derived for the input semigroup of an automaton to be left-discriminative.

This nested sequence of subautomata is also useful in that it is preserved by automaton homomorphisms and may be extended to a larger nested sequence, also preserved (in a limited sense) under homomorphisms. These facts are used
in a technique for constructing homorphisms of one finite automaton to another.

For additional examples and details of proofs, the reader is referred to Edwards and Bavel (1975a).

2. Preliminaries

For a nonempty set $\Sigma$ we denote by $\Sigma^*$ the free monoid over $\Sigma$, i.e., the set of all strings of finite length of members of $\Sigma$, including the empty string $\epsilon$, under concatenation; we denote by $\Sigma^+$, the free semigroup on $\Sigma$, i.e., $\Sigma - \{\epsilon\}$, also under concatenation. We denote the concatenation of strings $x$ and $y$ by $xy$. (The use of $\Sigma^*$ for both the set and the monoid presents no difficulty, which is also the case with the use of $\Sigma^+$ for both the set and the semigroup.)

An automaton is a triple $A = (S, \Sigma, \delta)$, where $S$ is a set (of states), $\Sigma$ is a nonempty set (the input alphabet), and $\delta: S \times \Sigma^* \rightarrow S$ is the transition function satisfying: $\forall s \in S$ and $\forall x, y \in \Sigma^*$, $\delta(s, xy) = \delta(\delta(s, x), y)$; and $\delta(s, \epsilon) = s$, $\forall s \in S$. For all automata to be considered in this article, we assume $S \neq \emptyset$. The symbols $A, S, \Sigma$, and $\delta$ are used generically when no ambiguity arises. $A$ is said to be finite if and only if $S$ is finite. The length of $x = x_1x_2 \cdots x_n \in \Sigma^*$, where each $x_i \in \Sigma$, is given by $|x| = n$. (By definition, $|\epsilon| = 0$.) If $\delta(s, x) \neq \delta(s, y)$, $x$ and $y$ are said to be distinguished by $s$.

An automaton $B = (T, \Sigma, \delta')$ is a subautomaton of $A = (S, \Sigma, \delta)$, written $B \ll A$, if and only if $T \subseteq S$ and $\delta'$ is the restriction of $\delta$ to $T \times \Sigma^*$. We use $\delta$ for $\delta'$ since no ambiguity arises. $S_B$ denotes the set of states of an automaton $B$. $B$ is a proper subautomaton of $A$ if and only if $B \ll A$, and $\emptyset \neq S_B \neq S_A$.

The set of successors of a state $s$ of an automaton $A$ is $\delta(s) = \{\delta(s, x): x \in \Sigma^*\}$. The source of a state $s$ is $\sigma(s) = \{t \in S_A: s \in \delta(t)\}$. The set of pure successors of a state $s$ is $\delta^+(s) = \{\delta(s, x): x \in \Sigma^+\}$. The pure source of a state $s$ is $\sigma^+(s) = \{t \in S_A: s \in \delta^+(t)\}$. The automaton generated by a state $s$ is $\langle s \rangle = (\delta(s), \Sigma, \delta)$. The set of successors of a set of states and the automaton generated by a set of states are defined by a straightforward extension of the previous definitions. If $A = \langle s \rangle$ for some $s \in S_A$, $s$ is called a generator of $A$ and $A$ is said to be singly generated. A primary of an automaton $A$ is a maximal (with respect to the ordering "$\ll$" on subautomata) singly generated subautomaton of $A$. (Additional subautomata of an infinite automaton are defined to be primaries in Bavel and Thomas (1967), but here we use the term only in the sense just defined.) A nonempty automaton is strongly connected if and only if it is singly generated by each of its states. Where $A_i = (S_i, \Sigma, \delta_i) \ll A$, $\forall i \in K$, for some nonempty indexing set $K$,

$$
\bigcup_{i \in K} A_i = \left( \bigcup_{i \in K} S_i, \Sigma, \delta' \right) \quad \text{and} \quad \bigcap_{i \in K} A_i = \left( \bigcap_{i \in K} S_i, \Sigma, \delta^* \right),
$$
where δ' and δ" are the restrictions of δ to

\[
\left( \bigcup_{i \in K} S_i \right) \times \Sigma^* \quad \text{and} \quad \left( \bigcap_{i \in K} S_i \right) \times \Sigma^*,
\]

respectively. Denote by \( \text{gen}(\delta) \) the set of generators of \( \langle \delta \rangle \).

Let \( A = (S, \Sigma, \delta) \) and \( B = (T, \Sigma, \gamma) \) be automata. A homomorphism on (or of) \( A \) to \( B \) is a mapping \( f: S \rightarrow T \) such that, \( \forall s \in S, \forall x \in \Sigma^*, f(\delta(s, x)) = \gamma(f(s), x) \).

A homomorphism \( f: A \rightarrow B \) is monic (respectively, epic) if and only if \( f: S \rightarrow T \) is monic (respectively, epic). If \( f: A \rightarrow B \) is an epic homomorphism, \( B \) is said to a homomorphic image of \( A \), a proper homomorphic image if \( f \) is not monic. A homorphism \( f: A \rightarrow B \) is an isomorphism if and only if it is monic and epic.

Where \( A = (S, \Sigma, \delta) \) is an automaton, the relation "\( \equiv_A \)" defined on \( \Sigma^* \) by:

\[
\forall x, y \in \Sigma^*, [x \equiv_A y \quad \forall s \in S, \delta(s, x) = \delta(s, y)],
\]

is an equivalence relation and a congruence on the free monoid \( \Sigma^* \), and thus \( \Sigma^*/\equiv_A \) is a monoid. We use \([x]_A \) or, more briefly, \([x]\), for the equivalence class of \( x \) under "\( \equiv_A \)". We also say \( x \equiv_A y \), if \( \forall t \in T, \delta(t, x) = \delta(t, y) \), where \( T \subseteq S \) and \( x, y \in \Sigma^* \). We define the input monoid of \( A \) by \( I_A^* = \Sigma^*/\equiv_A \) and the input semigroup of \( A \) by \( I_A = \Sigma^+/\equiv_A \), with the semigroup operation inherited from \( \Sigma^* \) and \( \Sigma^+ \) (whose symbol we may omit in order to simplify the notation, as no ambiguity arises).

Given a semigroup \( (J, \cdot) \), the semigroup automaton of \( J \) is \( \mathcal{A}(J) = (J, J, \delta) \), where \( \delta(i, j) = i \cdot j, \forall i, j \in J \). If \( (J, \cdot) \) is a monoid we may refer to \( \mathcal{A}(J) \) as the monoid automaton. We are particularly concerned with the case where \( J = I_A \) for an automaton \( A = (S, \Sigma, \gamma) \), and in this case we simplify the description of \( \mathcal{A}(I_A) \) by defining it to be \( (I_A, \Sigma, \delta) \) rather than \( (I_A, I_A, \delta) \). This is called the input-restricted input-semigroup automaton of \( A \), but may be abbreviated as the input-semigroup automaton of \( A \). The input-restricted input-monoid automaton is defined and abbreviated analogously.

We use the symbol "\( \cong \)" to denote isomorphism of automata as well as isomorphism of semigroups. \( N \) and \( N^+ \) denote the set of nonnegative integers and the set of positive integers, respectively.

3. The Left-Discrimination Sequence

As was shown in the Introduction, not every semigroup \( J \) is isomorphic to the input semigroup of its semigroup automaton. We now formalize two definitions and two results stated informally in the Introduction. Since the results are translations of known facts of semigroup theory, their proofs are omitted.

Definition 1. A semigroup \((J, \cdot)\) is said to be left-discriminative if and only if \( \forall i, j \in J, [\forall k \in J, ki = kj] \Rightarrow i = j \).
THEOREM 1. Let \((J, \cdot)\) be a semigroup and let \(\mathcal{A}(J) = (J, J, \delta)\) be the semigroup automaton of \((J, \cdot)\). Then

(i) \(I_{\mathcal{A}(J)}\) is a homomorphic image of \((J, \cdot)\).

(ii) If \((J, \cdot)\) is left-discriminative, then \(I_{\mathcal{A}(J)} \cong (J, \cdot)\).

(iii) If \((J, \cdot)\) is finite, then \((J, \cdot)\) is left-discriminative if and only if \(I_{\mathcal{A}(J)} \cong (J, \cdot)\).

As is intimated by the statement of parts (ii) and (iii) of Theorem 1, it is possible for an infinite semigroup \(J\) to be isomorphic to \(I_{\mathcal{A}(J)}\), without being left-discriminative. An example of such a semigroup is given at the end of this section.

We now define an automaton whose input semigroup is isomorphic to the extended regular representation of a given semigroup \(J\), and hence isomorphic to \(J\), as stated in Clifford and Preston (1961).

DEFINITION 2. Let \((J, \cdot)\) be a semigroup and let \(\lambda \notin J\). We define \(A_{\lambda}(J) = (\{\lambda\} \cup J, \cdot, \delta)\), where \(\delta: (\{\lambda\} \cup J) \times J \to \{\lambda\} \cup J\) is defined by, \(\forall i, j \in J\), \(\delta(i, j) = i \cdot j\) and \(\delta(\lambda, j) = j\).

THEOREM 2. Let \((J, \cdot)\) be a nonempty semigroup. Then \(I_{A_{\lambda}(J)} \cong (J, \cdot)\).

The fact the \(I_{\mathcal{A}(J)}\) may be a proper homomorphic image of \(I_A\) suggests that the process of repeatedly taking the input semigroup of the semigroup automaton of a semigroup may generate a nontrivial sequence of semigroups, each member of the sequence being a homomorphic image of its predecessor, which ceases to change when a left-discriminative semigroup is reached. We call such a sequence of semigroups "the left-discrimination sequence of \(A\)" when the first member of the sequence is \(I_A\). In view of Definition 2, for any semigroup \(J\), there exists a left-discrimination sequence which starts with \(J\), since \(J \cong I_{A_{\lambda}(J)}\) by Theorem 2. Thus, the left-discrimination sequence may also be regarded as a purely algebraic construction whose starting point is a semigroup rather than an automaton.

DEFINITION 3. Let \(A\) be an automaton. We define the sequence \(\{J_n\}\) of semigroups on \(A\) by

\[
J_0 = I_A = \Sigma^+ / \equiv_A ,
J_n = I_{\mathcal{A}(J_{n-1})} = \Sigma^+ / \equiv_{\mathcal{A}(J_{n-1})} , \quad \forall n \in \mathbb{N}^+.
\]

The sequence \(\{J_n\}\) thus defined is called the left-discrimination sequence of \(A\).

Theorem 1 points out the appropriateness of the name for this sequence, since \(J_{n+1} = J_n\) for finite \(A\) if and only if \(J_n\) is left-discriminative, and \(J_{n+1}\) is a
proper homomorphic image of \( J_n \) as long as \( J_n \) is not left-discriminative. As a simple consequence of the definition, we have

**Lemma 1.** Let \( \{ J_n \} \) be the left-discrimination sequence of an automaton \( A \). If \( \exists n \in \mathbb{N} \) such that \( J_n = J_{n+1} \), then, \( \forall m \in \mathbb{N}, \ J_n = J_{n+m} \).

When \( A \) is finite, \( \{ J_n \} \) must reach a left-discriminative semigroup which is equal to all of its successors. It is thus meaningful to define the length of the left-discrimination sequence to that point.

**Definition 4.** Let \( A \) be an automaton and let \( \{ J_n \} \) be the left-discrimination sequence of \( A \). The left-discrimination characteristic (l.d. characteristic) of \( A \) is denoted by \( X(A) \) and is defined by

\[
X(A) = \begin{cases} 
\text{the least integer } k \geq 0, & \text{such that } J_k = J_{k+1} \\
\infty, & \text{if no such integer } k \text{ exists.}
\end{cases}
\]

Where \( X(A) = k \), \( J_k \) is said to be the final semigroup of \( \{ J_n \} \).

If \( A \) is infinite, it is possible that \( X(A) = \infty \), as is shown by an example at the end of this section.

The process of finding \( X(A) \) and the members of \( \{ J_n \} \), as previously described, involves the given automaton \( A \) only as a starting point. The succeeding automata used in the sequence are semigroup automata, rather than subautomata of \( A \). The following several definitions and results lead to Theorems 3 and 4, which characterize the members of \( \{ J_n \} \) and \( X(A) \) in terms of \( A \) itself.

**Definition 5.** Let \( A = (S, \Sigma, \delta) \) be an automaton and let \( s \in S \). The source length \( l_A(s) \) of \( s \) relative to \( A \) is defined by

\[
l_A(s) = \begin{cases} 
0, & \text{if } \sigma^+(s) = \emptyset, \\
\sup \{ |x| : \delta(u, x) = s, x \in \Sigma^+, u \in S \}, & \text{otherwise.}
\end{cases}
\]

When the referent \( \sigma \) is clear from the context, \( "l(s)" \) may be used for \( "l_A(s)" \).

The properties of the source length included in Lemma 2 are immediate from the definition and elementary automata theory and thus are presented without proof.

**Lemma 2.** Let \( A \) be an automaton and let \( s \in S_A \).

\begin{align*}
\text{(i) } & \text{If } l(s) = 0 \text{ then } \langle s \rangle \text{ is a primary of } A. \\
\text{(ii) } & \text{If } \langle s \rangle \text{ is a primary of } A, \text{ then either } l(s) = 0 \text{ or } l(s) = \infty. \\
\text{(iii) } & \text{If } t \in \delta(s) \text{, then } l(t) \geq l(s); \text{ and if } s \neq t \text{ and } l(s) \neq \infty, \text{ then } l(t) > l(s).}
\end{align*}
(iv) If $t \in \sigma(s)$, then $l(t) \leq l(s)$; and if $s \neq t$ and $l(t) \neq \infty$, then $l(t) < l(s)$.

(v) If $\langle s \rangle$ has more than one generator, $l(t) = \infty$ for all $t \in \delta(s)$.

(vi) If $A$ is finite, then $\exists s \in S_A$ such that $l(s) = \infty$.

(vii) $\forall x \in \Sigma^*, \forall s \in S$, $l(\delta(s, x)) \geq |x|$.

It should be noted that a state of source length $n$ is no longer accessible after an input sequence of length greater than $n$, and is never accessible from a state of source length greater than $n$. Thus, the set of states whose source lengths are at least $n$ (for any $n \geq 0$) is closed under $\delta$ and is therefore the set of states of a subautomaton of $A$. The same closure property, and hence the same conclusion, holds for the set of states of infinite source length. These sets, and the corresponding subautomata, describe "concentric" subautomata of the automaton, i.e., a nested sequence of subautomata, and are important to what follows. We therefore formalize their definition.

**DEFINITION 6.** Let $A = (S, \Sigma, \delta)$ be an automata and let $n \in \mathbb{N}$. Then define $S^n = \{s \in S : l(s) \geq n\}$, and $A^n = (S^n, \Sigma, \delta) = \langle S^n \rangle$. Also define $S^\infty = \{s \in S : l(s) = \infty\}$, and $A^\infty = (S^\infty, \Sigma, \delta) = \langle S^\infty \rangle$.

In Theorem 3, below, we show that the input semigroup $J_{A^n}$ of $A^n$ is precisely the corresponding member $J_n$ of the left-discrimination sequence of $A$. The reader will recall that "$x \equiv_A y$" denotes the equivalence on the automaton $B$ of the two input strings $x$ and $y$; i.e., $\delta(s, x) = \delta(s, y), \forall s \in S_B$.

**THEOREM 3.** Let $A$ be an automaton with left-discrimination sequence $\{J_n\}$. Then, for each integer $n \geq 0$, $J_n = I_{A^n}$.

**Proof.** We prove this theorem by induction on $n$. $J_0 = I_A = I_{A^0}$ by definition. Let $k \in \mathbb{N}$ and suppose that $J_k = I_{A^k}$. Then $J_{k+1} = I_{A^{k+1}}$, again by definition. For any $x$ and $y$ in $\Sigma^*$,

\[
x \equiv_{I_{A^k}} y \iff \exists z \in \Sigma^* \text{ such that } [zx]_{A^k} \neq [zy]_{A^k} \\
\iff \exists z \in \Sigma^* \text{ such that } zx \neq A^k zy \\
\iff \exists s \in S^k \text{ such that } \delta(s, zx) \neq \delta(s, zy) \\
\iff \exists t \in S^{k+1} \text{ such that } \delta(t, x) \neq \delta(t, y) \text{ (where } t = \delta(s, z)) \\
\iff x \neq_{A^{k+1}} y.
\]

Hence $I_{\equiv_{I_{A^k}}} = I_{A^{k+1}} = I_{A^{k+1}}^1 = I_{A^{k+1}}^2$.

It is an immediate consequence of this theorem and Lemma 1 that, if $A$ is finite, the final semigroup of $\{J_n\}$ is identical to the input semigroup of $A^n$. We thus state it without further proof.
**Corollary.** Let \( A \) be a finite automaton. Then, the final semigroup of the left discrimination sequence of \( A \) is \( I_{A^n} \).

It should be clear that the l.d. characteristic may also be related to the sub-automata \( A^n \) of \( A \). Since \( J_n = I_{A^n} \) for all \( n \in \mathbb{N} \), then clearly \( J_n = J_{n+1} \) if and only if \( I_{A^n} = I_{A^{n+1}} \), possible only if all pairs of input strings distinguished on states of \( A^n \) are also distinguished on states of \( A^{n+1} \). This is the content of the following definition and theorem.

**Definition 7.** Let \( A = (S, \Sigma, \delta) \) be an automaton. A set \( V \subseteq S \) is said to be imitable by \( U \subseteq S \) if and only if \( \forall v \in V, \forall x, y \in \Sigma^+ \) such that \( \delta(v, x) \neq \delta(v, y) \), \( \exists u \in U \) such that \( \delta(u, x) \neq \delta(u, y) \).

**Theorem 4.** Let \( A = (S, \Sigma, \delta) \) be an automaton. Then \( X(A) \) is the least \( n \in \mathbb{N} \) such that \( S^n \) is imitable by \( S^{n+1} \). If no such integer exists, \( X(A) = \infty \).

An immediate consequence of Theorem 4 is given in the following corollary.

**Corollary 1.** Let \( A \) be an automaton. If \( n = \max \{ l(s) : s \in S, \ l(s) < \infty \} \), then \( X(A) \leq n + 1 \).

Some results on left-discrimination also emerge directly from Theorems 3 and 4, and may be stated without proof.

**Corollary 2.** Let \( A \) be an automaton. Then the following are equivalent:

(i) \( I_A \) is left-discriminative.

(ii) There is a generating set \( U \) of \( A \) (i.e., a set \( U \subseteq S \) such that \( \langle U \rangle = A \)) such that \( \forall u \in U, \{ u \} \) is imitable by \( \delta^+(u) \). \( \delta^+(u) \) is the set of successors of \( u \) by nonempty input strings.

(iii) \( \forall x, y \in \Sigma^+ \) such that \( x \equiv_A y \), \( \exists t \in S \) such that \( \sigma^+(t) \neq \emptyset \) and \( \delta(t, x) \neq \delta(t, y) \).

From the equivalence of (i) and (ii), it follows that an automaton such that \( \delta^+(S) = S \) (for example, a singly generated automaton with more than one generator or with a nonempty input string leading from a generator to itself) must have a left-discriminative input semigroup. In particular, a strongly connected automaton has a left-discriminative input semigroup.

We now give an example of an infinite automaton \( A \) such that \( X(A) = \infty \). If we let \( K = I_A \), it is also the case that \( K \cong I_{\sigma(K)} \), but \( K \) is not left-discriminative. Thus Theorem 1 (iii) does not hold in the infinite case.
Let $A = (S, \Sigma, \delta)$ where $S$, $\Sigma$, and $\delta$ are defined as follows.

$\Sigma = N^+$.

$S = \{\varepsilon\} \cup \{n_1 n_2 \ldots n_k : k \in N^+ \text{ and } \forall i \leq k, n_i \geq i\}$.

(Note that $S \subseteq \Sigma^*$.)

$\delta(\varepsilon, j) = j$.

$\delta(n_1 n_2 \ldots n_k, j) = \begin{cases} n_1 n_2 \ldots n_k j, & \text{if } j \geq k + 1; \\ n_1 n_2 \ldots n_k (k + 1), & \text{otherwise.} \end{cases}$

Part of the state diagram of $A$ is shown in Fig. 2.

![Fig. 2. Part of the state diagram of $A$.](image)

It is immediate from the definition of $A$ that $\forall u, v \in S$ such that $u \neq v$ and $|u| = |v|$, $\delta(u) \cap \delta(v) = \emptyset$ and $\langle u \rangle \preceq \langle v \rangle$, and that, $\forall n \in N$, $A_n = \bigcup_{|u| = n} \langle u \rangle$.

But, $\forall n \in N$, since $A_n$ is a union of isomorphic copies of, for example, $\langle 1 \ 2 \ 3 \ldots n \rangle$, it follows that $J_n = I_{A_n} = I_{\langle 1 \ 2 \ldots n \rangle}$. Furthermore, the transition function has been so defined that $1 \not\preceq \langle 1 \ 2 \ldots n \rangle$, but $1 \equiv \langle 1 \ 2 \ldots n + 1 \rangle$. Therefore, $\forall n \in N$, $S_n$ is not imitable by $S_{n+1}$ and thus $J_n$ is not left-discriminative and $X(A) = \infty$.

Since the state $\varepsilon$ distinguishes every pair of input strings distinguished by any of its successors, the input semigroup $I_A = \Sigma^*/\equiv_A = \Sigma^*/\equiv_{\langle 1 \rangle} = \{[u] : u \in S\}$. Likewise $I_1 = I_{\langle 1 \rangle} = I_{\langle 1 \rangle} = \{[u] : u \in \delta(1)\}$. $I_A \neq I_1$, as already shown, but if a mapping $f$ is defined from $I_A$ to $I_{\langle 1 \rangle}$ by

$f([\varepsilon]) = [\varepsilon],
\quad f([u]) = f([u])[n + 1], \quad \forall u \in S$,

it is not difficult to show that $f$ is an isomorphism and $J_0 = I_A \cong I_{\langle 1 \rangle} = I_{\langle 1 \rangle} = J_1$. In a similar manner $J_n \cong J_{n+1}$, $\forall n \in N$. 
4. LEFT-DISCRIMINATION AND HOMOMORPHISMS

It is well known that an epic homomorphism $f$ of an automaton $A$ onto an automaton $B$ induces an epic semigroup homomorphism of $I_A$ onto $I_B$. By a straightforward induction argument on the definition of the left-discrimination sequence, it is clear that such an $f$ also induces a semigroup homomorphism of $J_n^A$ onto $J_n^B$, $\forall n \in \mathbb{N}$. Thus, in this sense, the left-discrimination sequence is preserved under homomorphism.

In passing, we note that the left-discrimination characteristic is not necessarily preserved by homomorphisms; i.e., when $A$ and $B$ are finite and $f: A \to B$ is an epic homomorphism, $X(A)$ may be smaller than $X(B)$. (It is clear that $X(B)$ may be smaller than $X(A)$, since $B$ may be the one-state automaton and $X(B) = 0$.)

To illustrate the point, the automaton $A$ of Fig. 3 has a left-discriminative input semigroup, as is indicated by the table, as well as by the fact that every state of $A$ has an infinite source length. Now $\mathcal{A}(I_A^e)$ also has a left-discriminative input semigroup.

$A = (I_A, \Delta_A)$

\[ I_A = \mathcal{A}(I_A^e) \]

**Fig. 3.** $A$, its monoid automaton and input semigroup.
semigroup—the same one. Thus \( X(\mathcal{A}(I_3)) = 0 \). But \( \mathcal{A}(I_3) \) may be mapped homomorphically onto \( \langle b \rangle \), and \( X(\langle b \rangle) = 1 \). In a similar manner, it is possible to map an automaton of l.d. characteristic 0 onto a singly generated automaton of any given l.d. characteristic, since the latter may be embedded as a sub-automaton in an automaton of l.d. characteristic 0, whose monoid automaton will also have l.d. characteristic 0, and, by a theorem of Deussen (1966), the monoid automaton may always be mapped homomorphically onto the sub-automaton.

The left-discrimination sequence \( \{f_n\} \) of an automaton \( A \) was shown to be preserved under epic homomorphisms. It is thus not particularly surprising in view of Theorem 3, that the concentric shells \( \{A^n\} \) of an automaton are also preserved under homomorphisms.

The sequence of results which follows deals with the effect of homomorphisms on the source length function and on the sequence of subautomata \( \{A^n\} \). Lemma 3 aids in the proofs of these results, and states that if \( s \) is a state of finite source length \( n \), there exists a path of length \( n \) leading to \( s \) from a state of source length 0, and also that, if the automaton is finite, a state of infinite source length is reachable from a state which leads back into itself.

**Lemma 3.** (i) Let \( A = (S, \Sigma, \delta) \). For each \( s \in S \) such that \( l(s) = n < \infty \), \( \exists u \in S, \exists x \in \Sigma^* \) such that \( l(u) = 0 \), \( |x| = n \), and \( \delta(u, x) = s \).

(ii) Let \( A \) be finite. For each \( s \in S \) such that \( l(s) = \infty \), \( \exists v \in S, \exists x, y \in \Sigma^* \) such that \( \delta(v, x) = s \), \( |y| > 0 \), and \( \delta(v, y) = v \).

It is easily proved that homomorphisms of automata are monotonic non-decreasing on source lengths of states. This is the content of the following lemma.

**Lemma 4.** Let \( A = (S, \Sigma, \delta) \) and \( B = (T, \Sigma, \gamma) \) be automata, let \( f: A \to B \) be a homomorphism, and let \( s \in S \). Then \( l(s) \leq l(f(s)) \).

Lemma 5 indicates that, for every state of finite source length (of any source length in a finite automaton) of the range automaton of a homomorphism, there exists a preimage with the same source length in the domain automaton.

**Lemma 5.** Let \( A = (S, \Sigma, \delta) \) and \( B = (T, \Sigma, \gamma) \) be automata. Let \( f: A \to B \) be an epic homomorphism and \( t \in T \). If \( A \) is finite or if \( l(t) < \infty \), then \( \exists s \in S \) such that \( l(s) = l(t) \) and \( f(s) = t \).

**Proof.** If \( l(t) = n < \infty \), then \( \forall w \in T, \exists x \in \Sigma^* \) such that \( l(w) = 0 \), \( \gamma(w, x) = t \), and \( |x| = n \), by Lemma 3 (i). Also \( \exists v \in S \) such that \( f(v) = w \). By Lemma 4, \( l(v) = 0 \). Let \( s = \delta(v, x) \). Then \( f(s) = \gamma(f(v), x) = \gamma(w, x) = t \), and \( l(s) \geq n = l(t) \), by definition of source length. But \( l(s) \leq l(t) \) by Lemma 4, and hence \( l(s) = l(t) \).
If \( l(t) = \infty \) and \( A \) is finite, then \( B \) is also finite and by Lemma 3(iii), \( \exists w \in T, \exists x, y \in \Sigma^* \) such that \( \gamma(w, x) = t, |y| > 0, \) and \( \gamma(w, y) = w \). Also, \( \exists v \in S \) such that \( f(v) = w \). Since \( A \) is finite, \( \exists n \in N, \exists m \in N^+ \) such that \( \delta(v, y^n) = \delta(v, y^{n+m}) \). Then \( l(\delta(v, y^n)) = \infty \). But \( f(\delta(v, y^n)) = \gamma(f(v, y^n) = \gamma(w, y^n) = w \). Thus, with \( s = \delta(v, y^nx) \), \( l(s) = \infty \), and \( f(s) = f(\delta(v, y^nx) = \gamma(w, x) = t \).

Lemma 4 implies that, for \( 0 \leq n \leq \infty \), \( S^n \) is mapped into \( T^n \) by any homomorphism of \( A \) into \( B \), and Lemma 5 implies that for \( 0 \leq n \leq \infty \), \( S^n \) is mapped onto \( T^n \) by any epic homomorphism from \( A \) onto \( B \), and that \( S^n \) is mapped onto \( T^n \) if \( A \) is finite. These facts are summarized in the following.

**Theorem 5.** \( \forall n \in N, \) if \( f: A \to B \) is an epic homomorphism then \( f \) maps \( A^n \) onto \( B^n \). If \( A \) is finite, \( f \) maps \( A^n \) onto \( B^n \).

It should be noted that, if \( A \) is infinite, \( S^n \) may be empty, as in the case with the automaton \( A \) of Fig. 4. \( B \) is a homomorphic image of \( A \), but \( S_{A^n} = \emptyset \) is clearly not mapped onto \( S_{B^n} = S_B \).

**Fig. 4.** \( B \), a homomorphic image of \( A \).

5. **Recursively Generated Subautomata, \( * \)-Levels, and Homomorphisms**

The sequence of concentric subautomata \( A = A^0, A^1, A^2, ..., A^\infty \) of a finite automaton has been shown to be preserved by epic homomorphism. This fact could be used to construct (some or all) homomorphisms on one finite automaton to another. However, too often \( A^\infty \) is too large a subautomaton for the sequence to offer much economy. In this section, we define an extension of the sequence just mentioned. The new members of the sequence are subautomata of \( A^\infty \) and this sequence too is preserved by homomorphisms. The new members are numbered “from the inside out” until one of them is coincident with \( A^\infty \). The resulting sequence and its preservation by homomorphisms present an interesting technique for constructing homomorphisms, whose starting point is, in a sense, opposite to that described in Bavel (1968). The following several definitions and lemmas lead to the main results, on which the suggested technique is based.

**Definition 8.** Let \( A = (S, \Sigma, \delta) \) be an automaton. A state \( s \) of \( A \) is said to be **recurrent** if and only if \( \exists x \in \Sigma^+ \) such that \( \delta(s, x) = s \). A subautomaton
B = (T, Σ, δ) of A is said to be recurrently generated (abbreviated, r.g.) if and only if B = ⟨R⟩ for some R ⊆ S such that ∀s ∈ R, s is recurrent.

It is easy to show that, if B is finite, B is recurrently generated if and only if ∀t ∈ T, l_b(t) = ∞. The proof is also straightforward that the homomorphic image of a recurrent state is recurrent, and that every recurrent state in the range of a homomorphism on a finite automaton is the image of some recurrent state in the domain.

We are now ready to define the extension of the sequence of concentric subautomata. 

**Definition 9.** Let A = (S, Σ, δ) be a finite automaton and let B₁, ..., Bₖ be its strongly connected subautomata. Define

\[
\text{core}(A) = \bigcup_{i=1}^{k} B_i
\]

We define the sequence \{Aₙₙ\} recursively as follows.

\[
A₁ = A = \text{core}(A).
\]

For each \(n \in \mathbb{N}^+\), let \(\mathcal{R}_n^A = \{⟨s⟩ ≤ A: ⟨s⟩ is r.g., ⟨s⟩ ≤ A_n, and for all proper r.g. subautomata C of ⟨s⟩, C ≤ A_n\}. Then define

\[
A_{(n+1)*} = A_n^* \cup \left( \bigcup_{R ∈ \mathcal{R}_n^A} R \right), \quad \forall n \in \mathbb{N}^+.
\]

(For completeness, we let \(\mathcal{R}_0^A = \text{core}(A)\). We also use \(\mathcal{R}_n\) when the automaton referred to is obvious).

To illustrate the definition, we refer to the automaton A of Fig. 5. Here,
\[ A^\infty = \langle b \rangle \] and the r.g. singly generated subautomata are \( \langle b \rangle, \langle c \rangle, \langle d \rangle, \langle f \rangle, \langle g \rangle, \) and \( \langle h \rangle; A_+ = \langle g \rangle \cup \langle h \rangle, A_{2+} = A_+ \cup \langle d \rangle \cup \langle f \rangle \) (\( \langle c \rangle \) and \( \langle b \rangle \) are excluded since \( \langle f \rangle \not\subseteq \langle c \rangle \) and \( \langle d \rangle \not\subseteq \langle b \rangle \)), \( A_{2+} = A_{2+} \cup \langle e \rangle \) (\( \langle e \rangle \) is not r.g.), and \( A_{4+} = A_{2+} \cup \langle b \rangle \).

**Lemma 6.** Let \( A \) be a finite automaton and let \( n \in \mathbb{N}^+ \). Then \( A_{n+} = A_{(n+1)+} \) implies \( A_{n+} = A^\infty \).

**Proof.** By Definition 9, \( A_{n+} = A_{(n+1)+} \) implies \( \mathcal{R}_n = \emptyset \). Hence, \( \forall s \in S - S_{A_{n+}} \), \( l_A(s) < \infty \). Since \( A_{n+} \) is r.g., \( \forall s \in S_{A_{n+}} \), \( l_{A_{n+}}(s) = \infty \) and hence \( l_A(s) = \infty \).

Consequently, \( A_{n+} = A^\infty \).

Lemma 6 points out the fact that, since \( A \) is finite, the sequence \( A_{n+} \) must reach \( A_{\infty} \) in a finite number of steps. Lemma 7, below, shows that every r.g. \( \langle s \rangle \subseteq A \) is a primary of \( A_{n+} \) for some \( n \in \mathbb{N} \) (for finite \( A \)).

**Lemma 7.** Let \( A \) be a finite automaton. Then, for every r.g. \( \langle s \rangle \subseteq A \), there exists a unique \( n \in \mathbb{N} \) such that \( \langle s \rangle \in \mathcal{R}_n \). Moreover, the members of \( \mathcal{R}_n \), together with those primaries of \( A_{n+} \) which are not subautomata of members of \( \mathcal{R}_n \), are precisely the primaries of \( A_{(n+1)+} \).

**Proof.** The first conclusion of the lemma follows directly from the definition of \( \mathcal{R}_n \) (Definition 9). Clearly, each member of \( \mathcal{R}_n \) is a primary of \( A_{(n+1)+} \), again by the definition of \( \mathcal{R}_n \). If there exists a primary of \( A_{(n+1)+} \) which is not a member of \( \mathcal{R}_n \), it must be a subautomaton of \( A_{n+} \), since \( A_{(n+1)+} = A_{n+} \cup (\bigcup_{s \in \mathcal{R}_n} R) \). But \( A_{n+} \subseteq A_{(n+1)+} \) and hence a primary of \( A_{(n+1)+} \), which is a subautomaton of \( A_{n+} \), must be primary of \( A_{n+} \).

By Lemma 7, for every recurrent state \( s \) of \( A \) there exists a unique integer \( m \) such that \( \langle s \rangle \in \mathcal{R}_m \); hence \( A_{(m+1)+} \) is the first member of the sequence \( \{A_{n+}\} \) of which \( \langle s \rangle \) is a primary. This motivates the following definition.

**Definition 10.** Let \( A \) be a finite automaton and let \( \langle s \rangle \subseteq A \) be r.g. Then \( l_\times(s) \), the \( \times \)-level of \( s \), is given by

\[ l_\times(s) = n + 1, \quad \text{where} \quad \langle s \rangle \in \mathcal{R}_n. \]

Equivalently,

\[ l_\times(s) = \min\{n \in \mathbb{N}^+: \langle s \rangle \text{ is a primary of } A_{n+}\}. \]

It should be noted that a homomorphism \( f: A \to B \), even if it is epic, need not map \( A_{n+} \) onto \( B_{n+} \), since several primaries of "different \( \pi \)-levels" in \( A \) may be mapped into the "same \( \pi \)-level in \( B \)" (although a recurrent state must still be mapped to a recurrent state). However, there is sufficient monotonicity
exhibited by the sequence \( \{A_n\} \) to render it useful, as is indicated in the following two theorems.

**Theorem 6.** Let \( A = (S, \Sigma, \delta) \) and \( B = (T, \Sigma, \gamma) \) be finite automata, let \( f: A \to B \) be a homomorphism, let \( \langle a \rangle \) be a primary of \( A_n \) for some \( n \in \mathbb{N}^+ \). Then \( \exists k \in \mathbb{N}^+ \) such that \( k \leq n \) and \( f(\langle a \rangle) \) is a primary of \( B_{k} \).

**Proof.** As previously argued, if \( \langle a \rangle \) is a primary of \( A_n \), \( f(\langle a \rangle) = f(\langle a \rangle) \) is r.g. Then, by Lemma 7, \( f(\langle a \rangle) \) is primary of \( B_{k} \), for some \( k \in \mathbb{N}^+ \). We prove \( k \leq n \) by induction on \( n \).

If \( n = 1 \), \( \langle a \rangle \) is primary of \( A_1 \), and therefore, is strongly connected. Hence \( f(\langle a \rangle) \) is strongly connected and hence a primary of \( B_{k} \).

Let \( n > 1 \) and suppose that, \( \forall m \in \mathbb{N}^+ \) such that \( m < n \), if \( \langle a \rangle \) is a primary of \( A_n \), then \( f(\langle a \rangle) \) is a primary of \( B_{k} \) for some \( k \leq m \). Let \( \langle a \rangle \) be a primary of \( A_n \). Let \( \langle t \rangle \) be a proper r.g. subautomaton of \( f(\langle a \rangle) \). Then, there exists an r.g. subautomaton \( \langle s \rangle \) of \( \langle a \rangle \) such that \( f(s) = t \). (Suppose such is not the case, let \( f(s, x) = f(t, x) \) for some \( x \in Z^+ \), and let \( s \in S \) such that \( f(s) = t \). Then, \( \delta(s, x) \neq s \) and \( f(\delta(s, x)) = f(t, x) = t \). An iteration of the same argument produces an infinite sequence of \( x \)-successors of \( s \), each different from its predecessors, which is impossible in a finite automaton.) Now \( \langle s \rangle \) is a proper subautomaton of \( \langle a \rangle \), since \( f(\langle s \rangle) \) is a proper subautomaton of \( f(\langle a \rangle) \). Consequently, \( \langle s \rangle \) is a primary of some \( A_{p} \), where \( p < n \), by Lemma 7. By the inductive hypothesis, \( f(\langle s \rangle) = \langle t \rangle \) is a primary of \( B_{q} \), for some \( q \in \mathbb{N} \), where \( q \leq p < n \); hence \( l_p(f(a)) \leq n - 1 \) or \( f(a) \in \mathcal{R}_{n-1} \) and is thus a primary of \( B_{n} \). In either case, the desired result follows.

The proof of Theorem 6 exhibits the strong dependence of homomorphic mappings of r.g. \( \langle a \rangle \) on its proper r.g. subautomata. This dependence allows yet a stronger conclusion, i.e., that the homomorphic image of a primary of \( A_n \) cannot "skip *-levels" in the range automaton, as is shown by the following theorem.

**Theorem 7.** Let \( A = (S, \Sigma, \delta) \) and \( B = (T, \Sigma, \delta) \) be finite automata, let \( f: A \to B \) be a homomorphism, let \( \langle a \rangle \) be a primary of \( A_n \) for some \( n \in \mathbb{N}^+ \), and let \( m = \max \{ l_p(f(s)) \mid \langle s \rangle \text{ is a proper r.g. subautomaton of } \langle a \rangle \} \). Then, \( l_p(f(a)) = m \) or \( l_{p}(f(a)) = m + 1 \).

**Proof.** As was shown in the proof of Theorem 6, every proper singly generated r.g. subautomaton of \( f(a) \) is the \( f \)-image of a properly singly generated r.g. subautomaton of \( a \). Thus, where \( \langle t \rangle \) is a proper r.g. subautomaton of \( f(a) \), \( \langle t \rangle \leq B_{(m+1)} \), since \( l_{p}(t) \leq m \). Hence, \( \langle t \rangle \leq \mathcal{R}_p \), for some \( p \in \mathbb{N} \) such that \( p \leq m \). Therefore, either \( f(a) \leq B_{(m+1)} \), implying \( l_p(f(a)) = m + 1 \), or else \( f(a) \leq B_{(m+1)} \). In the later case, \( f(a) \in \mathcal{R}_m \) and \( l_p(f(a)) = m \), since \( f(a) \) has a proper r.g. subautomaton \( \langle s \rangle \) such that
$l_*(f(s)) = m$, implying that $\langle f(s) \rangle \in \mathcal{P}_m^B$ and $\langle f(s) \rangle \ll B_{m^*}$; since $f(s) \ll \langle f(a) \rangle$, $\langle f(a) \rangle \ll B_{m^*}$ and thus $\langle f(a) \rangle \not\in \mathcal{P}_q^B$, $\forall q < m$.

An algorithm for finding all homomorphisms on one finite automaton to another is discussed in Bavel (1968). It operates "from the outside in" by mapping homomorphically entire primaries of the domain and then extending the homomorphisms on the primaries to the entire automaton through matching the images on the states common to distinct primaries. Although that algorithm appears efficient, there are cases where the large number of states of the range automaton which must be considered as possible images of the generators of primaries of the domain renders the effort excessive.

Theorems 6 and 7 suggest another algorithm for the same purpose, one which works "from the inside out," which may offer economies, at least in some important cases. The latter algorithm starts the mapping process with the core of the domain and proceeds by mapping each primary of a new *-level in all ways permitted by Theorem 7, on the basis of the mappings of the preceding *-levels. A detailed presentation of this algorithm is too voluminous for this article; however, a brief description should suffice to impart its drift.

Suppose that the recurrent states of $A$ and $B$ and their respective successors have been determined, and that the strongly connected subautomata of both automata are known (e.g., by algorithms in Bavel (1968)). Determine all homomorphisms of core $(A)$ to core $(B)$ by an algorithm such as Bavel (1968). Now complete the mapping of $A_2$, before starting to map $A_2^*$, and continue moving out *-level by *-level until $A^*$ is mapped. In each *-level, use Theorem 7 to restrict the number of possible images of a selected generator of a primary of this *-level to the two *-levels dictated by the mappings already recorded for primaries of the preceding levels which are subautomata of the present primary. At each such stage, all "valid homomorphisms" (nonempty extensions, in the terminology of Bavel (1968) are found and recorded for further use in the process of mapping the next higher *-level. This process ends with the mappings of $A^\infty$, at which point use is made of the sequence $\{A_n\}$ to close the gap between $A$ and $A^\infty$. The latter operation follows a procedure similar to that advocated in the earlier mentioned algorithm. An advantage the algorithm just sketched has over that of Bavel (1968) is that much of the work may be done directly from the state diagrams of the automata, rather than relying exclusively on the transition tables.

7. Concluding Remarks

The reader may have noted already that, in the definition of $\mathcal{P}_n$ (Definition 9), the requirement that $\langle s \rangle$ be r.g. could have been deleted without damaging the desirable properties of the resulting sequence. The sequence of concentric
shells thus generated would differ from \( \{A_n\} \), and would terminate in \( A \), rather than in \( A^\omega \). The analogs of Theorems 6 and 7 for this sequence hold, and consequently, an algorithm similar to the one described above may be constructed.

The sequence \( \{A_n\} \) was employed in this article primarily because the primaries of its members provide convenient intermediate substructures between the primaries of the entire automaton on one hand and every singly generated subautomation on the other. Both of these extremes are likely in our view to result in considerably more effort than the course taken here. What makes possible the use of the class of r.g. subautomata is the fact that it shares the following property with the class of all singly generated subautomata: a homomorphic image of a member of the class in the domain automaton is a member of the same class of the range automaton.

As the final remark, we offer a combined extended sequence of concentric subautomata and the corresponding extended left-discrimination sequence. The sequence \( \{A_n\} \) of subautomata of \( A^\omega \) was defined “from the inside out.” Each is suitable for its principal application: the left-discrimination sequence in the case of \( \{A^n\} \) and homomorphisms in the case of \( \{A_n\} \). It is possible to join these two sequences in a single-numbering scheme, while at the same time extending \( \{A_n\} \) to include intermediate automata between \( A_n \) and \( A_{(n+1)} \) which are not r.g. Let \( A \) be a finite automaton and let \( A_k = A^\omega \) and \( A_{(k-1)} \neq A^\omega \), for some \( k \in N^+ \). Define a sequence \( \{A_{n^\infty+m}\} \) as follows. For all \( m \geq 0 \) and all \( n \) such that \( 1 \leq n \leq k \), define \( A_{n^\infty+m} = A^m \); \( A_{n^\infty} = A_{(k-n+1)^\infty} \); \( S_{n^\infty+1} = S_{n^\infty} = \{\text{gen}(s)\}: s \text{ is recurrent and } l(s) = k - n + 1 \}; \( A_{n^\infty+1} = (S_{n^\infty+1}, \Sigma, \delta) \); and \( A_{n^\infty+m} = (A_{n^\infty+1})^{m-1} \).

It easily follows that \( A^{(n+1)^\infty} = (A_{n^\infty+1})^\infty \). The left-discrimination sequence \( \{J_n\} \) of semigroups associated with \( A \) may also be extended simply by defining \( J_{n^\infty+m} = I_{A_{n^\infty+m}} \). With this device helping the sequence to cross the left-discriminative boundary, \( J_{n^\infty+m} \) for \( m \neq 1 \) may resort to the original definition. \( J_{n^\infty+m} \) is still a homomorphic image of all its predecessors in the sequence, as is \( J_n \), but \( \{J_{n^\infty+m}\} \) need not be preserved by epic automata homomorphisms as was \( \{J_n\} \), since \( \{A_n\} \), and hence \( \{A_{n^\infty+m}\} \) is not so preserved.

Received: September 1, 1971; Revised: October 1976

References


Bavel, Z., and Thomas, J. W. (1967), On the decomposability of monadic algebras and


