ABSTRACT FAMILIES OF LANGUAGES*

Seymour Ginsburg
System Development Corporation, Santa Monica, California
University of Southern California, Los Angeles, California

Sheila Greibach
Harvard University, Cambridge, Massachusetts
Consultant, System Development Corporation, Santa Monica, California

Abstract

The notion of an abstract family of languages (AFL) as a family of sets of words satisfying certain properties common to many types of formal languages is introduced. Operations preserving AFL are then considered. The concept of an abstract family of acceptors (AFA) is also introduced and shown to give rise to an AFL. A necessary and sufficient condition on an AFL is presented in order that the AFL come from some AFA. Finally, abstract families of transducers (AFA with output) are discussed.

Introduction

During the past decade, a large number of families of formal languages of interest to either linguists, computer scientists, or automata theorists have appeared in the literature. In the course of these studies, certain properties have been discovered which are enjoyed by many of these families. (For example, many of these languages are closed under intersection with regular sets.) The aim of the present investigation is to abstract this commonality. In particular, we abstract the notions of "family of languages," "family of acceptors," and "family of transducers," and study relations and properties between them. Many of these relations are already known for specific types of languages, acceptors, and transducers. Thus we are able to extend and unify the theory as well as obviate the need to rederive these relations and properties for each type of language once the axiomatic properties are satisfied.

The paper is divided into five sections. Section one presents the basic definition of an abstract family of languages (abbreviated AFL). The definition differs from the traditional methods of introducing languages, i.e., by grammars and by acceptors. It defines an AFL as any family of sets of words which is closed under six types of operations ($\cup$, $\cdot$, $+$, inverse homomorphism, ϵ-free homomorphism, and intersection with regular sets).

Section two is concerned with operations which preserve AFL, $\mathcal{L}$, i.e., operations f such that f(L) is in $\mathcal{L}$ for each L in $\mathcal{L}$. For example, it is shown that ϵ-free a-transducers, a general type of finite state machine, preserve AFL. This result is indirectly used to prove that each AFL contains all ϵ-free regular sets. Each AFL is also closed under substitution by ϵ-free regular sets. If an AFL is closed under arbitrary homomorphism, then it is closed under a-transduction, initial subwords, and all subwords.

Section three is concerned with one-way acceptors. The notion of an abstract family of acceptors (abbreviated AFA) is introduced and is shown to encompass many types of acceptors already in the literature. Each AFA yields an AFL in a rather natural way (Theorem 3.2). Acceptors with certain space restrictions and acceptors with certain time restrictions lead to additional AFL (Theorems 3.3 and 3.4).

In section four, the smallest AFL containing a given countable family of sets and closed under arbitrary homomorphism is characterized in two ways (Theorem 4.1 and its corollary). A consequence of this result is a characterization of those AFL obtained from at least one AFA (namely, an AFL, $\mathcal{L}$, is obtained from some AFA if and only if $\mathcal{L}$ is countable and closed under arbitrary homomorphism). However, every countable AFL containing the set $\{\epsilon\}$ can be derived from the time-restricted acceptors in some AFA (Theorem 4.3).

Section five concerns transducers, i.e., acceptors with an output structure. Two characterizations involving transducers are presented of when an AFL derived from some AFA is closed under intersection with an arbitrary element of some given AFL (Theorem 5.1). Also presented are two characterizations involving transductions of when an arbitrary set is in the AFL derived from some given AFA (Theorem 5.3). The notion of a deterministic transducer is then formulated and used to present another characterization of when an arbitrary set is in the AFL derived from some given AFA.

While a knowledge of acceptor theory and formal language theory is helpful for background, motivation, and heuristic purposes, it is not really necessary except in understanding some of the examples.

*Research sponsored in part by the Air Force Cambridge Research Laboratories, Office of Aerospace Research, USAF, under Contract F1962867C0008, and by the Air Force Office of Scientific Research, Office of Aerospace Research, USAF, under AFOSR Grant No. AF-AFOSR-1203-67.

Section 1. Basic Definition

We shall be concerned with certain families of subsets of words over an infinite alphabet. These families of sets, called "abstract families of languages," are to have some of the closure properties enjoyed by many families of formal languages studied in mathematical linguistics, computer science, and automata theory. In the present section we introduce these abstract families of languages.

Definition. An abstract family of languages (AFL) is a pair $(\Sigma, \mathcal{L})$, or $\mathcal{L}$ when Σ is understood, where

(1) Σ is a countably infinite set of symbols,

(2) for each L in $\mathcal{L}$ there is a finite set $\Sigma_1 \subseteq \Sigma$(1) such that $L \subseteq \Sigma_1^*$,

(3) $\mathcal{L}$ is closed under the operations of $\cup$, $\cdot$, $+$, inverse homomorphism(2) ϵ-free homomorphism,(3) and intersection with regular sets,

(4) $L \neq \emptyset$ for some L in $\mathcal{L}$.

Condition (1) permits a family of languages to contain sets over larger and larger alphabets, while condition (2) requires each set to be over a finite alphabet. Condition (4) is to avoid triviality. Condition (3), of course, is the core of the definition, implying the existence of certain sets in $\mathcal{L}$ from the existence of others.

Each of the following families of formal languages is an AFL: Regular sets [3],(4) ϵ-free regular sets [3],(5) context-free languages [3], ϵ-free context-free languages [3], context-sensitive languages [4,12], deterministic context-sensitive languages [4,12], one-way nondeterministic stack languages [6], nondeterministic real-time list languages [7], recursive sets, recursively enumerable sets, nondeterministic real-time n-tape Turing machine languages, sets recognized in n^k tape size by a deterministic Turing machine [10], and sets recognized in n^k tape size by a nondeterministic Turing machine. Thus the notion of an abstract family of languages as defined here is broad enough to encompass many of the formal languages studied in linguistics, in automata theory, and in computer science. We shall see that many results already proved for different families of languages hold for AFL.

The condition that $\mathcal{L}$ be closed under ϵ-free homomorphism instead of arbitrary homomorphism is given in order to include certain families of well-known languages. Thus the ϵ-free regular sets, the ϵ-free context-free languages, the context-sensitive languages, and the recursive sets would be excluded if AFL were required to be closed under arbitrary homomorphism. For, under arbitrary homomorphism, the closure of the ϵ-free regular sets is the regular sets and the closure of the ϵ-free context-free languages is the context-free languages. Under arbitrary homomorphism, the closure of both the recursive sets and the context-sensitive languages is the recursively enumerable sets [2,8].

Section 2. Closure Operations

The main properties of an AFL, those given in (3) of the definition, involve closure of the family under certain operations. It is thus natural to consider other operations which preserve AFL. In this section we present a number of functions which preserve AFL. Most of these functions have already been considered in connection with individual families of languages, such as the context-free sets and the regular sets.

A trivial example of an operation which preserves AFL is subtraction by a regular set. (For let L be in $\mathcal{L}$ and R regular. Then $L \subseteq \Sigma_1^*$ for some Σ_1. Since Σ_1^*-R is regular, $L \cap (\Sigma_1^*-R) = L-R$ is in $\mathcal{L}$.)

One way to obtain a number of specific operations which preserve AFL is to consider the following device.

Definition. A sequential transducer with accepting states, abbreviated a-transducer,(6) is a 6-tuple $M = (K, \Sigma_1, \Sigma_2, \lambda, p_0, F)$, where

(1) K, Σ_1, and Σ_2 are finite sets of symbols (states, inputs, and outputs respectively).

(1) Unless stated otherwise, Σ_1, Σ_2, Σ_3, etc., will always denote a finite subset of Σ.

(2) By a homomorphism, we mean a homomorphism of a specific set Σ_1^* into a specific set Σ_2^*. As noted in footnote 1, Σ_1 and Σ_2 are finite subsets of Σ. A function g of Σ_2^* into $2^{\Sigma_1^*}$ is said to be an inverse homomorphism if there exists a homomorphism f of Σ_1^* into Σ_2^* such that $g(x) = \{y/f(y) = x\}$.

(3) A function f is said to be ϵ-free if $f(x) \neq \epsilon$ for all $x \neq \epsilon$.

(4) Citations are given to [3] whenever possible, although in most cases this is not the original source.

(5) A set is said to be ϵ-free if it does not contain ϵ.

(6) This device is called an "f-transducer" in [9].

(2) λ is a mapping of a finite subset of $K \times \Sigma_1^* \times K$ into the finite subsets of Σ_2^*.

(3) p_0 is in K (the start state).

(4) $F \subseteq K$ (set of accepting states).

If λ maps its domain into the finite subsets of Σ_2^+, then M is called ϵ-free.

The a-transducer effects an operation as follows.

Notation. For each a-transducer $M = (K,\Sigma_1,\Sigma_2,\lambda,p_0,F)$ and each w in Σ_1^* let M(w) be the union of all sets of the form

$$\lambda(p_0,a_{i_1},p_{i_1})\lambda(p_{i_1},a_{i_2},p_{i_2})\ldots\lambda(p_{i_{n-1}},a_{i_n},p_{i_n}),$$

where $n \geq 1$, $w = a_{i_1}\ldots a_{i_n}$, each a_{i_j} in Σ_1^*, each p_{i_j} in K, and p_{i_n} in G. Let $M(L) = \bigcup_{w \text{ in } L} M(w)$.

When considering an a-transducer $M = (K,\Sigma_1,\Sigma_2,\lambda,p_0,F)$ in conjunction with an AFL, $\mathcal{L}$, we shall always assume that Σ_1 and Σ_2 are subsets of Σ.[7] In treating M(L), L in $\mathcal{L}$, it suffices to consider $L \subseteq \Sigma_1^*$. For, otherwise, we could consider $L \cap \Sigma_1^*$ in $\mathcal{L}$.

A basic result is the following.

Theorem 2.1. Let $\mathcal{L}$ be an AFL. Then $\mathcal{L}$ is closed under ϵ-free a-transducers. If $\mathcal{L}$ is closed under arbitrary homomorphisms, then $\mathcal{L}$ is closed under arbitrary a-transducers.

An important case of an a-transducer occurs when F = K and the domain of λ is essentially $K \times \Sigma_1 \times K$. In particular,

Definition. A nondeterministic gsm (or nondeterministic generalized sequential machine) is a 5-tuple $M = (K,\Sigma_1,\Sigma_2,\lambda,p_0)$,[8] where

(1) K, Σ_1, and Σ_2 are finite sets,

(2) λ is a mapping of $K \times (\Sigma_1 \cup \{\epsilon\}) \times K$ into the finite subsets of Σ_2^*,

(3) $\lambda(q,\epsilon,q) = \{\epsilon\}$ for each q in K, and $\lambda(q,\epsilon,q') = \emptyset$ if $q \neq q'$.

---

(7) See footnote 1.

(8) More formally, is a 6-tuple $M = (K,\Sigma_1,\Sigma_2,\lambda,p_0,K)$.

The nondeterministic gsm M is called ϵ-free if $\lambda(q,a,q') \subseteq \Sigma_2^+$ for each (q,a,q') in $K \times \Sigma_1 \times K$.

Suppose M is an ϵ-free nondeterministic gsm and $\mathcal{L}$ is an AFL. Let M' be the a-transducer $(K,\Sigma_1,\Sigma_2,\lambda',p_0,K)$, where $\lambda' = \lambda$ on $K \times \Sigma_1 \times K$ and $\lambda' = \emptyset$ otherwise. Let L be in $\mathcal{L}$. Then $M(L) = M(L-\{\epsilon\}) = M'(L)$ if ϵ is not in L, and $M(L) = M(L-\{\epsilon\}) \cup \{\epsilon\} = M'(L) \cup \{\epsilon\}$ if ϵ is in L. Since M' is an ϵ-free a-transducer, M'(L) is in $\mathcal{L}$. If ϵ is in L, then $L \cap \{\epsilon\} = \{\epsilon\}$ is in $\mathcal{L}$, so that $M'(L) \cup \{\epsilon\}$ is in $\mathcal{L}$. In either case, M(L) is in $\mathcal{L}$. Thus we get

Corollary 1. Let $\mathcal{L}$ be an AFL. Then $\mathcal{L}$ is closed under ϵ-free nondeterministic gsm mappings. If $\mathcal{L}$ is closed under arbitrary homomorphisms, then $\mathcal{L}$ is closed under arbitrary nondeterministic gsm mappings.

The form of a nondeterministic gsm used most frequently is when $\lambda(q,a,q')$ always consists of at most one element, and for each (q,a) in $K \times \Sigma_1$, $\lambda(q,a,q')$ is nonempty for exactly one q'. This instance is called a gsm [3] (generalized sequential machine) and has customarily been defined as a 6-tuple $M = (K,\Sigma_1,\Sigma_2,\delta,\lambda,p_0)$ with δ a mapping of $K \times \Sigma_1$ into K (next state function), λ a mapping of $K \times \Sigma_1$ into Σ_2^* (output function), and p_0 in K (start state).

We now consider the "pseudoinverse" of a nondeterministic gsm. For later use, we define the pseudoinverse of an arbitrary function of $2^{\Sigma_1^*}$ into $2^{\Sigma_2^*}$.

Definition. Let f be a function of $2^{\Sigma_1^*}$ into $2^{\Sigma_2^*}$. The pseudoinverse of f is the mapping f^{pi} of $2^{\Sigma_2^*}$ into $2^{\Sigma_1^*}$ defined by $f^{pi}(U) = \{w/f(w) \cap U \neq \emptyset\}$ for each $U \subseteq \Sigma_2^*$.

Corollary 2. Each AFL is closed under every pseudoinverse of a nondeterministic gsm mapping.

A number of closure results are summarized in the next two corollaries.

Corollary 3. Let $\mathcal{L}$ be an AFL closed under arbitrary homomorphism. For L in $\mathcal{L}$ and R regular,

$L/_R = \{w/wy \text{ in } L \text{ for some } y \text{ in } R\}$

and $_R\backslash L = \{w/yw \text{ in } L \text{ for some } y \text{ in } R\}$

are both in $\mathcal{L}$.

Corollary 4. If $\mathcal{L}$ is an AFL closed under arbitrary homomorphism, then for each L in $\mathcal{L}$,

$\mathrm{Init}(L) = \{w \neq \epsilon / wy \text{ in } L \text{ for some } y\}$,

$\mathrm{Fin}(L) = \{w \neq \epsilon / yw \text{ in } L \text{ for some } y\}$,

and $\mathrm{Sub}(L) = \{w \neq \epsilon / uwv \text{ in } L \text{ for some } u, v\}$

are in $\mathcal{L}$.

In the previous theorem we showed that AFL are preserved under ϵ-free a-transducers. We now generalize that result by permitting a bounded number of consecutive ϵ-outputs.

Definition. Let $M = (K,\Sigma_1,\Sigma_2,\lambda,p_0,F)$ be an a-transducer. Then M is ϵ-output bounded under non-ϵ-inputs if there exists an integer k with the following property: If $x_1,\ldots,x_{k+1}$, $p_1,\ldots,p_{k+2}$ are such that each x_i is in Σ_1^*, each p_j is in K, and ϵ is in $\lambda(p_i,x_i,p_{i+1})$ for all $1 \leq i \leq k+1$, then $x_i = \epsilon$ for some i.

Theorem 2.2. Let $\mathcal{L}$ be an AFL containing $\{\epsilon\}$. If M is an a-transducer which is ϵ-output bounded under non-ϵ-inputs and L is in $\mathcal{L}$, then M(L) is in $\mathcal{L}$.

As already noted, the ϵ-free regular sets form an AFL. Using Theorem 2.1 and the next lemma, we can show that each AFL contains the ϵ-free regular sets.

Lemma 2.1. For each ϵ-free regular set R there exists a nondeterministic ϵ-free gsm M such that $R = M^{pi}(\{a,b\}^*c)$.

Theorem 2.3. Let $\mathcal{L}$ be an AFL. Then $\mathcal{L}$ contains each ϵ-free regular set. $\mathcal{L}$ contains each regular set if and only if there exists some L in $\mathcal{L}$ containing ϵ (equivalently, $L = \{\epsilon\}$ is in $\mathcal{L}$).

The previous results do not use the closure of $\mathcal{L}$ under $\cdot$ and $+$. (Indeed, using $\cdot$ and $+$, we can omit Lemma 2.1 and simplify the proof of Theorem 2.3. In fact, Theorem 2.3 could then be proved in section one.) Hence the previous theorems hold for those families of formal languages, such as the linear context-free languages [3] and the nonterminal bounded context-free languages [9], which are only closed under intersection with regular sets, inverse homomorphism, ϵ-free homomorphism, and union.

Now (a) $\mathcal{L}$ is closed under $\cup$, $\cdot$, and $+$, and (b) the ϵ-free regular sets are the smallest family of sets containing the ϵ-free finite sets and closed under $\cup$, $\cdot$, and $+$. Thus it follows that $\mathcal{L}$ is closed under substitution of ϵ-free regular sets by languages.(9) If $\mathcal{L}$ contains $\{\epsilon\}$, thus if $\mathcal{L}$ is closed under arbitrary homomorphism, then $L^* = L^+ \cup \{\epsilon\}$ is in $\mathcal{L}$ for arbitrary L in $\mathcal{L}$. Thus $\mathcal{L}$ is closed under substitution of regular sets by languages if $\mathcal{L}$ contains $\{\epsilon\}$. We next consider substitution of languages by ϵ-free regular sets and by arbitrary regular sets.

Theorem 2.4. Let $\mathcal{L}$ be an AFL. Then $\mathcal{L}$ is closed under substitution by ϵ-free regular sets. If $\mathcal{L}$ is closed under arbitrary homomorphism, then $\mathcal{L}$ is closed under substitution by regular sets.

Remark. The hypothesis that $\mathcal{L}$ be closed under homomorphism in order for substitution by regular sets to preserve $\mathcal{L}$ cannot be removed. For in the degenerate case when the substitution is a homomorphism, it has been noted earlier that there exist families $\mathcal{L}$ of languages not preserved by arbitrary homomorphism but preserved by ϵ-free homomorphism.

We could introduce the notion of "full substitution" in $\mathcal{L}$, i.e., the substitution for each element in $\Sigma_1 \subseteq \Sigma$ of an arbitrary element of $\mathcal{L}$. However, we shall not be concerned with this concept since it does not appear to be of significance for AFL in general.

Section 3. Acceptors

One popular way to obtain a family of formal languages is to define a family of acceptors and consider the set of words recognized by these devices. Motivated by this point of view, we now introduce the notion of an abstract family of (one-way) acceptors and show that it yields, in a natural manner, and AFL.

Definition. An abstract family of (one-way) acceptors (AFA) is an ordered pair $(\Omega,\mathcal{D})$, or $\mathcal{D}$ when Ω is understood, with the following properties:

(1) Ω is a 7-tuple $(K,\Sigma,\Gamma,I,f,g,Z_0)$, where

(a) K and Σ are countably infinite sets, and Γ and I are countable disjoint sets.

(b) f is a mapping from $\Gamma^* \times (\Gamma \cup I)^*$ into $\Gamma^* \cup \{\emptyset\}$ with the property that for all (γ,u) in $\Gamma^* \times (\Gamma \cup I)^*$, every symbol occurring in $f(\gamma,u)$ occurs in γu.

(9)Let Σ_1 be a finite set. For each a in Σ_1 let $\tau(a)$ be a subset of Σ_a^*. Let $\tau(\epsilon) = \{\epsilon\}$ and for each $x_1\ldots x_k$, each x_i in Σ_1, let $\tau(x_1\ldots x_k) = \tau(x_1)\ldots\tau(x_k)$. Then τ is called a substitution, and $\tau(E) = \bigcup_{w \text{ in } E} \tau(w)$ is called the substitution by E by the $\tau(a)$.

(c) g is a function from Γ^* into the finite subsets of Γ^*.

(d) Z_0 is in Γ and $g(Z_0) = \{Z_0\}$.

(e) For each γ in $g(\Gamma^*)$, there exists a word α_γ in $(\Gamma \cup I)^*$ with the property that $f(\gamma', \alpha_\gamma) = \gamma'$ for all γ' such that $g(\gamma')$ contains γ.

(2) $\mathfrak{D}$ is the family of all elements (called <u>acceptors</u>) $D = (K_1, \Sigma_1, \Gamma_1, I_1, G, \delta, p_0, Z_0, F)$, where

(a) K_1, Σ_1, and Γ_1 are finite nonempty subsets of K, Σ, and Γ, respectively.

(b) I_1 and G are finite subsets of I and $g(\Gamma_1^*) = \bigcup_{\gamma \text{ in } \Gamma_1^*} g(\gamma)$, respectively.

(c) $F \subseteq K_1$, p_0 is in K_1, and Z_0 is in $\Gamma_1 \cap G$.

(d) δ is a function from $(K_1 \times (\Sigma_1 \cup \{\epsilon\}) \times G$ into the finite subsets of $K_1 \times (\Gamma_1 \cup I_1)^*$.

Intuitively, K is the set of all possible "states," Σ is the set of all possible "inputs," and Γ is the set of all possible "auxiliary" symbols (i.e., symbols going into the auxiliary storage). Thus each acceptor has only a finite number of states, inputs, and auxiliary storage configurations. Z_0 is the "initial" auxiliary symbol, p_0 the "start" state of the acceptors, and F the set of "accepting" states. δ is the "move" function. I, f, and g will be explained shortly.

We now present the notation which describes the behavior of acceptors.

<u>Notation</u>. Let $\vdash$ (or $\vdash_D$ when the acceptor D is to be emphasized) be the relation on $K_1 \times \Sigma_1^* \times \Gamma_1^*$ defined as follows: For a in $\Sigma_1 \cup \{\epsilon\}$, $(p, aw, \gamma) \vdash (p', w, \gamma')$ if there exist $\bar{\gamma}$ in $g(\gamma) \cap G$ and u such that (p', u) is in $\delta(p, a, \bar{\gamma})$ and $f(\gamma, u) = \gamma'$. Let $\overset{i}{\vdash}$ and $\overset{*}{\vdash}$ (or $\overset{i}{\vdash}_D$ and $\overset{*}{\vdash}_D$ when D is to be emphasized) be the relations on $K_1 \times \Sigma_1^* \times \Gamma_1^*$ defined by $(p, aw, \gamma) \overset{0}{\vdash} (p, aw, \gamma)$, $(p, aw, \gamma) \overset{1}{\vdash} (p', w, \gamma')$ if $(p, aw, \gamma) \vdash (p', w, \gamma')$, and for $k \geq 0$, $(p, w, \gamma) \overset{k+1}{\vdash} (p', w', \gamma')$ if there exists (p'', w'', γ'') such that $(p, w, \gamma) \overset{k}{\vdash} (p'', w'', \gamma'') \vdash (p', w', \gamma')$. Write $(p, w, \gamma) \overset{*}{\vdash} (p', w', \gamma')$ if $(p, w, \gamma) \overset{k}{\vdash} (p', w', \gamma')$ for some $k \geq 0$.

Since each occurrence of a symbol of the input is read at most once, namely from left to right, each acceptor is a one-way device.

Informally speaking, g is a generalized read head which examines the auxiliary storage to select the symbol or symbols being scanned. $g(Z_0) = \{Z_0\}$ is required for uniformity. For each acceptor D, G denotes that subset of $g(\Gamma_1^*)$ on which D can operate. This is done in order that D be finitely specified. (For example, in a list machine,[10] if the auxiliary storage is $w_1 \sigma w_2$, g selects all $y_1 \sigma y_2$ such that $w_1 \sigma w_2 = w_1' y_1 \sigma y_2 w_2'$ for some w_1' and w_2'.) The function g is allowed to be multivalued to account for devices, such as the list machine and certain pda models, which permit a number of different words to be selected. Usually g is single valued.

The function f is a generalized write head which uses an instruction, i.e., element of $(\Gamma \cup I)^*$ to transform the auxiliary storage. It is important for f to be single valued even if D is nondeterministic. While there may be many instructions which can be used, once the instruction is chosen the next configuration of D is uniquely determined (if it exists). The condition that each symbol in $f(\gamma, u)$ occur in γu ensures that for each Γ_1 and I_1, f maps $\Gamma_1^* \times (\Gamma_1 \cup I_1)^*$ into $\Gamma_1^* \cup \emptyset$.

In a configuration (p, aw, γ), g usually selects some subsequence $\bar{\gamma}$ (in $g(\gamma) \cap G$) of γ. For (p', u) in $\delta(p, a, \bar{\gamma})$, p' is the next state and u the instruction. f uses this instruction (in conjunction with γ) to give the next auxiliary storage $f(\gamma, u)$. The net effect is $(p, aw, \gamma) \vdash (p', w, f(\gamma, u))$. If $f(\gamma, u)$ is empty, then either γ is not a "legitimate" auxiliary storage configuration, or u is not a "legitimate" instruction, or γ and u are "legitimate" but not in relation to each other. Moreoever, D operates only when γ and $f(\gamma, u)$ are in Γ_1^*, i.e., γ and $f(\gamma, u)$ both must be permissible auxiliary storage configurations. In all other cases, D blocks.

Condition (1e) of the definition of an AFA is needed for technical reasons. Its purpose is to give a uniform procedure for leaving the auxiliary storage invariant while changing the state or advancing the input.

In general, $\mathfrak{D}$ is not offered as a model for a computational system. Rather, it serves as a formulation of devices which yield sets of words by the following procedure. (In Theorem 3.2 we shall show that the resulting sets form an AFL.)

---

(10) See example 6.

Definition. Let $\mathfrak{D}$ be an AFA. For each acceptor $D = (K_1, \Sigma_1, \Gamma_1, I_1, G, \delta, p_o, Z_o, F)$, let $L(D)$, called the set accepted by D, be the set of words

$$\{w \text{ in } \Sigma_1^*/(p_o, w, Z_o) \overset{*}{\vdash} (p, \epsilon, Z_o) \text{ for some } p \text{ in } F\}.$$

Let $L(\mathfrak{D}) = \{L(D)/D \text{ in } \mathfrak{D}\}$.

For many families of acceptors, it has been customary to define w in $L(\mathfrak{D})$ if $(p_o, w, Z_o) \overset{*}{\vdash} (p, \epsilon, \gamma)$ for some p in F and some γ in Γ_1^*. We need the extra condition that $\gamma = Z_o$ in order to guarantee that the auxiliary storage can be reset to the initial condition, a situation needed in many results. It is easy to see that this variation of acceptance does not change the total sets of words accepted for most of the well-known families of acceptors.

We now show that the model of AFA as formulated encompasses (with minor variations) a number of types of one-way acceptors already in the literature.

Examples. (1) Let $\Gamma = \{Z_o\}$ and $I = \emptyset$. Let $f(\gamma, u) = \gamma$ for all u in Γ^*.[11] Let $g(\epsilon) = \emptyset$, $g(Z_o^i) = \{Z_o\}$ for $i \geq 1$, and $G = \{Z_o\}$. Thus each acceptor in the resulting AFA does not depend on the auxiliary storage. In case δ is a function from $K_1 \times (\Sigma_1 \cup \{\epsilon\}) \times G$ into the finite subsets of $K_1 \times Z_o$, with $\delta(q, \epsilon, Z_o) = \{(q, Z_o)\}$ for all q in K_1, the acceptor is the customary nondeterministic finite state acceptor [13]. If, in addition, $\delta(q, \epsilon, Z_o)$ maps $K_1 \times \Sigma_1 \times G$ into the unit subsets of $K_1 \times Z_o$, then the acceptor D becomes the familiar finite state acceptor. In all cases, $\mathcal{L}(\mathfrak{D})$ is the family of regular sets.

(2) Pushdown acceptors (pda) [3]. Let K, Σ, and Γ be countably infinite. Let $g(\epsilon) = \emptyset$ and $g(\gamma Z) = \{Z\}$ for each γ in Γ^* and each Z in Γ. Let $f(\gamma Z, u) = \gamma u$ for all Z in Γ, γ in Γ^*, and u in Γ^*. Then $G \subseteq \Gamma_1$ and δ is the usual pda transition function. Observe that $f(\gamma Z, Z) = \gamma Z$, so that (1e) is satisfied. Now $G \subseteq \Gamma_1$ and δ is undefined for Z not in G. Thus we can assume that $G = \Gamma_1$, as is customary in a pda. $\mathcal{L}(\mathfrak{D})$ is, of course, the family of context-free languages.

(3) One-way, nondeterministic 1-counters [2]. Let $I = \emptyset$, $\Gamma = \{Z_o\}$, $g(Z_o^k) = \{Z_o\}$ for $k \geq 1$, and $g(\epsilon) = \{\epsilon\}$. Let $f(Z_o^k, Z_o^\ell) = Z_o^{k+\ell-1}$ if $k + \ell - 1 \geq 1$ and $f(Z_o^k, Z_o^\ell) = \epsilon$ if $0 \leq k + \ell \leq 1$. Note that δ is a function from $K_1 \times (\Sigma_1 \cup \{\epsilon\}) \times \{Z_o, \epsilon\}$ into the finite subsets of $K_1 \times Z_o^*$. Since $f(Z_o^k, Z_o) = Z_o^k$, condition (1e) is satisfied.

(4) One-way stack acceptors [6]. Let Γ contain at least two elements, Z_o and $\uparrow$. Let $\overline{\Gamma} = \Gamma - \{\uparrow\}$ and $I = \{+, -\}$. Let $g(Z_o) = \{Z_o\}$ and $g(\gamma_1 Z \uparrow \gamma_2) = \{Z\}$ for all γ_1 and γ_2 in $\overline{\Gamma}^*$ and Z in $\overline{\Gamma}$. Let $f(Z_o, Z_o) = Z_o$, $f(Z_o, \uparrow +) = Z_o \uparrow$, $f(Z_o \uparrow, \uparrow -) = Z_o$, and for all Z_1, Z_2 in $\overline{\Gamma}$, γ_1, γ_2 in $\overline{\Gamma}^*$, and γ_3, γ_4 in $\overline{\Gamma}^*$, with $\gamma_3 \gamma_4 \neq \epsilon$, let $f(\gamma_1 Z_2 Z_1 \uparrow \gamma_2, -) = \gamma_1 Z_2 \uparrow Z_1 \gamma_2$, $f(\gamma_1 Z_1 \uparrow Z_2 \gamma_2, +) = \gamma_1 Z_1 Z_2 \uparrow \gamma_2$, $f(\gamma_1 Z_1 \uparrow \gamma_2, Z_1) = \gamma_1 Z_1 \uparrow \gamma_2$, and $f(\gamma_3 Z_1 \uparrow, \gamma_4) = \gamma_3 \gamma_4 \uparrow$. (The interpretation of the instruction + is to move right on the stack, and of - to move left. $f(Z_o, \uparrow +) = Z_o \uparrow$ allows the pointer $\uparrow$ to be added to the right of Z_o. $f(Z_o, \uparrow -)$ allows the pointer $\uparrow$ to be removed from $Z_o \uparrow$.) Since $f(\gamma_1 Z_1 \uparrow \gamma_2, Z_1) = \gamma_1 Z_1 \uparrow \gamma_2$ and $f(Z_o, Z_o) = Z_o$, (1e) is satisfied. Note that $L(D) = \emptyset$ for the one-way stack acceptor $D = (K_1, \Sigma_1, \Gamma_1, I_1, G, \delta, p_o, Z_o, F_1)$ unless $(p', \uparrow +)$ is in $\delta(p_o, a, Z_o)$ and $(p'', \uparrow -)$ is in $\delta(p, b, Z_o)$ for some a, b in $\Sigma_1 \cup \{\epsilon\}$, p', p in K_1, and p'' in F_1. $\mathcal{L}(\mathfrak{D})$ is the family of one-way stack languages.

If $f(\gamma_3 Z_1 \uparrow, \gamma_4) = \gamma_3 \gamma_4 \uparrow$ is altered to $f(\gamma_3 Z_1 \uparrow, \gamma_4) = \gamma_3 Z_1 \gamma_4 \uparrow$, then the resulting $\mathcal{L}(\mathfrak{D})$ is the family of nonerasing one-way stack languages [5].

We now obtain a sequence of results leading to the theorem (3.2) that each $\mathcal{L}(\mathfrak{D})$ is an AFL.

Remarks. (1) Let $D_1 = (K_1, \Sigma_1, \Gamma_1, I_1, G, \delta, p_o, Z_o, F)$. Let $\Gamma_1 \subseteq \Gamma_2$ and $I_1 \subseteq I_2$. Then $L(D_1) = L(D_2)$, where $D_2 = (K_1, \Sigma_1, \Gamma_2, I_2, G, \delta, p_o, Z_o, F)$. To see this, it suffices to show that for γ_1 in Γ_1^*, $(p_1, w_1, \gamma_1) \vdash_{\overline{D}_1} (p_2, w_2, \gamma_2)$ if and only if $(p_1, w_1, \gamma_1) \vdash_{\overline{D}_2} (p_2, w_2, \gamma_2)$. The "only if" being obvious, consider the "if." Suppose γ_1 is in Γ_1^* and $(p_1, w_1, \gamma_1) \vdash_{\overline{D}_2} (p_2, w_2, \gamma_2)$. Then γ_2 is in Γ_2^*. Also, there exist $\overline{\gamma}$ in $g(\gamma_1) \cap G$ and u in $(\Gamma_2 \cup I_2)^*$ such that (p', u) is in $\delta(p, a, \overline{\gamma})$ and $f(\gamma_1, u) = \gamma_2$.

[11] The functions f and g are always to be $\emptyset$ except where otherwise stated.

By definition of δ, u is in $(\Gamma_1 \cup I_1)^*$. Since the symbols of γ_2 are in $\gamma_1 u$ and $\Gamma \cap I = \emptyset$, γ_2 is in Γ_1^*. Thus $(p_1, w_1, \gamma_1) \mid_{\overline{D}_1} (p_2, w_2, \gamma_2)$.

(2) Suppose that $L = L(D)$ for $D = (K_1, \Sigma_1, \Gamma_1, I_1, G, \delta, p_o, Z_o, F)$. Then by (1e) in the definition of an AFA, for each γ in G there exists an instruction α_γ in $(\Gamma \cup I)^*$ with the property that $f(\gamma', \alpha_\gamma) = \gamma'$ for all γ' such that $g(\gamma')$ contains γ. For each such γ let E_γ be the set of symbols in α_γ. Let $\Gamma_2 = \Gamma_1 \cup \bigcup_\gamma (\Gamma \cap E_\gamma)$ and $I_2 = I_1 \cup \bigcup_\gamma (I \cap E_\gamma)$. By the previous remark, $L(D_1) = L(D_2)$ for the acceptor $D_2 = (K_1, \Sigma_1, \Gamma_2, I_2, G, \delta, p_o, Z_o, F)$. Thus we may always assume that each α_γ is in $(\Gamma_1 \cup I_1)^*$.

The first preliminary result yields the fact that $\mathcal{L}(\mathcal{D})$ is closed under $\cup$, $\cdot$, and $*$.

<u>Theorem 3.1</u>. If $R \subseteq \Sigma_1^*$ is regular and τ is a substitution such that $\tau(a)$ is in $\mathcal{L}(\mathcal{D})$ for each a in Σ_1, then $\tau(R)$ is in $\mathcal{L}(\mathcal{D})$.

<u>Corollary</u>. $\mathcal{L}(\mathcal{D})$ is closed under $\cup$, $\cdot$, and $*$.

<u>Theorem 3.2</u>. If $\mathcal{D}$ is an AFA, then $\mathcal{L}(\mathcal{D})$ is an AFL closed under arbitrary homomorphism.

We have just seen that an arbitrary AFA gives rise to an AFL in a natural manner. We shall shortly (Theorems 3.4 and 3.5) place restrictions on the auxiliary storage and on the time (i.e., number of moves) to arrive at other connections between acceptors and languages.

<u>Notation</u>. Let T be a mapping of the non-negative integers into the positive integers. An acceptor $D = (K_1, \Sigma_1, \Gamma_1, I_1, G_1, \delta_1, p_o, Z_o, F_1)$ in $\mathcal{D}$ is said to be in $\mathcal{D}_{T(n)}$ if, for all w in Σ_1^*, $(p_o, w, Z_o) \mid\!\overset{*}{-} (p, \epsilon, \gamma)$ implies $|\gamma| \le T(|w|)$. Let $\mathcal{L}_{T(n)} = \bigcup_{k \ge 0} \mathcal{L}(\mathcal{D}_{T(kn)})$.

Thus $\mathcal{D}_{T(kn)}$ is the set of acceptors in $\mathcal{D}$ whose auxiliary storage (starting from p_o and Z_o) is restricted in length by T and k. Such a restriction has already been considered for the family of Turing acceptors [10].

<u>Example 5</u>. Turing acceptors and linear bounded acceptors [12]. Let K, Σ, and Γ be be infinite, with + and - not in Γ. Let Γ contain the special symbols Z_o, B, and $\upharpoonleft$, and let $\overline{\Gamma}$ and $\Gamma - \{Z_o, \upharpoonleft\}$. Let $I = \{+, -\}$. Let $g(\gamma_1 Z \upharpoonleft \gamma_2) = Z$ for all Z in $\overline{\Gamma}$ and γ_1, γ_2 in $\overline{\Gamma}^*$. Let $f(Z_o, B\upharpoonleft) = B\upharpoonleft$ and $f(B\upharpoonleft, Z_o) = Z_o$. For all Z, Z_1, Z_2 in $\overline{\Gamma}$ and γ_1, γ_2 in $\overline{\Gamma}^*$, let $f(\gamma_1 Z_1 \upharpoonleft \gamma_2, Z_2) = \gamma_1 Z_2 \upharpoonleft \gamma_2$, $f(\gamma_1 Z \upharpoonleft \gamma_2, -) = \gamma_1 \upharpoonleft Z \gamma_2$ if $\gamma_1 \ne \epsilon$, $f(Z \upharpoonleft \gamma_2, -) = B \upharpoonleft Z \gamma_2$, $f(\gamma_1 Z \upharpoonleft \gamma_2, +) = \gamma_1 Z Z' \upharpoonleft \gamma_2'$, where Z' is in $\overline{\Gamma}$ and $\gamma_2 = Z' \gamma_2'$, $f(\gamma_1 Z \upharpoonleft, +) = \gamma_1 Z B \upharpoonleft$, $f(\gamma_1 B \upharpoonleft, B-) = \gamma_1 \upharpoonleft$ if $\gamma_1 \ne \epsilon$, and $f(B \upharpoonleft Z \gamma_1, B+) = Z \upharpoonleft \gamma_1$. Thus the only legitimate instructions are

Z : replace scanned symbol by A,

\- : move left, adding B if at left end,

\+ : move right, adding B if at right end,

B- : erase leftmost B if at left end,

and B+ : erase rightmost B if at right end.

This family consists of the Turing acceptors which receive inputs, one at a time, on demand, perform the operation of either write, erase, move left, or move right, and can compute at any time without receiving new inputs. Note that, for an acceptor to operate, it must start by replacing the initial Z_o with $B\upharpoonleft$. At the end of a computation which accepts a word, $B\upharpoonleft$ must be replaced with Z_o.

It is clear that $\mathcal{L}(\mathcal{D}_{kn}) = \mathcal{L}(\mathcal{D}_n)$ for each $k \ge 1$, so that $\mathcal{L}_n(\mathcal{D}) = \mathcal{L}(\mathcal{D}_n)$. $\bigcup_{k \ge 1} \mathcal{D}_{kn}$ is the family of linear bounded acceptors and $\mathcal{L}_n(\mathcal{D})$ is the family of context sensitive languages [11].

We can show that under a mild restriction on the function T, $\mathcal{L}_{T(n)}(\mathcal{D})$ is an AFL.

<u>Theorem 3.3</u>. If T is a monotonically increasing function, then $\mathcal{L}_{T(n)}(\mathcal{D})$ is an AFL.

<u>Remarks</u>. (1) It can be shown, though not done here, that $\mathcal{L}(\mathcal{D}_{T(n)})$ need not be an AFL.

(2) In general, it is not true that $\bigcup_{k \ge 1} \mathcal{D}_{T(kn)}$ is an AFA. If it were, then by Theorem 3.2, $\mathcal{L}_{T(n)}(\mathcal{D})$ would be closed under arbitrary homomorphism. However, in example 5, an AFA, $\mathcal{D}$, is presented in which $\mathcal{L}_n(\mathcal{D})$ is the family of context-sensitive languages, a family not closed under arbitrary homomorphism [2,8].

We now place a time restriction on the acceptors.

<u>Definition</u>. Let k be a positive integer and $\mathcal{D}$ an AFA. Let $\mathcal{D}_k^t$ be the set of all D in $\mathcal{D}$ such that $(p, \epsilon, q) \mid\!\frac{\ell}{D} (p', \epsilon, q')$ implies $\ell \le k$. Let $\mathcal{L}^t(\mathcal{D}) = \bigcup_{k=0}^{\infty} \mathcal{L}(\mathcal{D}_k^t)$. Each L in $\mathcal{L}^t(\mathcal{D})$ is called <u>quasi-real-time</u>.

Thus D is in $\mathcal{D}_k^t$ if there are no more than k consecutive ϵ-moves of D.

Example 6. Let $\overline{\Gamma} = \Gamma - \{Z_o\}$ and $I = \emptyset$. For w in $\overline{\Gamma}^* Z_o \overline{\Gamma}^*$, let

$$g(w) = \{\gamma_1 Z_o \gamma_2 / w = \gamma_3 \gamma_1 Z_o \gamma_2 \gamma_4 \text{ for some } \gamma_3 \text{ and } \gamma_4\}.$$

For $w_1, w_2, \gamma_1, \gamma_2, \gamma_3, \gamma_4$ in $\overline{\Gamma}^*$, let

$$f(w_1\gamma_1 Z_o \gamma_2 w_2, \gamma_1 Z_o \gamma_2 Z_o \gamma_3 Z_o \gamma_4) = w_1 \gamma_3 Z_o \gamma_4 w_2.$$

Then $\mathcal{D}$ is the family of one-way list-storage acceptors considered in [7]. It is shown in [7] that $\mathcal{L}(\mathcal{D}_o^T) = \mathcal{L}^T(\mathcal{D})$.

Theorem 3.4. $\mathcal{L}^t(\mathcal{D})$ is an AFL.

Remarks. (1) One can show that $\mathcal{L}(\mathcal{D}_o^t)$ has all the properties of an AFL except that inverse homomorphism need not hold.

(2) For each D in $\mathcal{D}$, one can effectively find a homomorphism h and a D' in $\mathcal{D}_o^t$ such that $h(L(D')) = L(D)$. Suppose that $\mathcal{L}(\mathcal{D})$ includes some recursively enumerable set which is not recursive. Suppose further that f is total recursive and g(q) is recursive for each q. This implies that L(D) is recursive for each D in $\mathcal{D}_k^t$. For $\mathcal{D}$ satisfying the above (and there certainly exist such AFA), $\mathcal{L}^t(\mathcal{D})$ is not closed under arbitrary homomorphism.

Section 4. Countable AFL

In the previous section we saw that AFA give rise to AFL but not conversely. Here we shall present a necessary and sufficient condition for an AFL to be obtained from some AFA.

Since $\mathcal{L}(\mathcal{D})$ is always a countable family for each $\mathcal{D}$, in order to obtain the desired necessary and sufficient condition we shall study countable families of languages. In particular, we shall examine three special types of countable AFL.

Definition. Let Σ be a countably infinite set and $S = \{S_1, \ldots, S_i, \ldots\}$ a countable (possibly finite) family of nonempty subsets of Σ^* with the property that for each S_i in S a finite set $\Sigma_i \subseteq \Sigma$ can be found satisfying $S_i \subseteq \Sigma_i^*$. Then $(\Omega, \mathcal{D}(S))$ denotes any AFA with the following properties:

(1) Z_o is not in Σ.

(2) $\Gamma = \Sigma \cup \{Z_o\}$.

(3) $I = \{0\} \cup \{i/S_i \text{ in } S\}$.

(4) g is defined by $g(Z_o) = Z_o$, $g(\gamma Z) = Z$ for all γ in Σ^* and Z in Σ, and $g = \emptyset$ otherwise.

(5) f is defined by $f(Z_o, i) = Z_o$ for all i in I, $f(Z_o, Z) = Z$ for all Z in Σ, $f(\gamma, 0) = \gamma$ for all γ in Σ^*, $f(\gamma, Z) = \gamma Z$ for all γ in Σ^* and Z in Σ, $f(\gamma, i) = Z_o$ for all i in I-{0} and γ in S_i, and $f = \emptyset$ otherwise.

Obviously for any given S there is at least one family $(\Omega, \mathcal{D}(S))$. Furthermore, except for the labeling of the elements in K, $(\Omega, \mathcal{D}(S))$ is unique, so that it may be written as $\mathcal{D}(S)$. By Theorem 3.2, $\mathcal{L}(\mathcal{D}(S))$ is an AFL closed under arbitrary homomorphism. Clearly $\mathcal{L}(\mathcal{D}(S))$ is countable.

Since f in $\mathcal{D}(S)$ is defined only over $\Gamma^* \times (\Sigma \cup I)$, there is no loss in assuming that δ maps $K \times (\Sigma_o \cup \{\epsilon\}) \times G$ into the finite subsets of $K \times ((\Gamma_o - \{Z_o\}) \cup I_o)$ for each given acceptor $D = (K, \Sigma_o, \Gamma_o, I_o, G, \delta, p_o, Z_o, F)$ in $\mathcal{D}(S)$.

In the remainder of this section, Σ and $S = \{S_1, \ldots\}$ are assumed given. Σ_i will always denote a particular finite subset of Σ satisfying $S_i \subseteq \Sigma_i^*$.

Definition. Let $\mathcal{F}(S)$ be the smallest AFL containing S. Let $\hat{\mathcal{F}}(S)$ be the smallest AFL containing S and closed under arbitrary homomorphism.

Obviously $\mathcal{F}(S)$ and $\hat{\mathcal{F}}(S)$ both exist and are countable. The family $\mathcal{F}(S)$ is the intersection of all AFL containing S. The family $\hat{\mathcal{F}}(S)$ is the intersection of all those AFL containing S and closed under arbitrary homomorphism.

We now examine the relationship between $\mathcal{F}(S)$, $\hat{\mathcal{F}}(S)$, and $\mathcal{L}(\mathcal{D}(S))$.

Theorem 4.1. $\mathcal{L}(\mathcal{D}(S)) = \hat{\mathcal{F}}(S)$. Thus $\mathcal{L}(\mathcal{D}(S)) = \mathcal{F}(S)$ if and only if $\mathcal{F}(S)$ is closed under arbitrary homomorphism.

Suppose S consists of a finite number of sets, say $S = \{S_1, \ldots, S_n\}$. Then Theorem 4.1 is still true if Γ in $(\Omega, \mathcal{D}(S))$ is defined to be $\bigcup_{i=1}^{n} \Sigma_i \cup \{Z_o\}$.

As a corollary, we have another characterization of $\hat{\mathcal{F}}(S)$.

Corollary. $\hat{\mathcal{F}}(S) = \mathcal{A}(S)$, where $\mathcal{A}(S)$ is the smallest family of sets containing the family {M(S)/S in S, M an a-transducer} and closed under $\cup$, $\cdot$, and $*$.

Example. As an application of the corollary of Theorem 4.1, we exhibit an AFL which is not closed under word reversal.[12] Let

$$S = \{a^n b^m / n > m \geq 0\} \text{ and } U = \{a^n b^m / 0 \leq n < m\},$$

where a and b are distinct symbols. Obviously S^R is in $\hat{\mathcal{F}}(S)$ if and only if U is in $\hat{\mathcal{F}}(S)$. To show that $\hat{\mathcal{F}}(S)$ is not closed under word reversal, it thus suffices to show that U is not in $\hat{\mathcal{F}}(S)$.

Suppose U is in $\hat{\mathcal{F}}(S)$. Let $\mathcal{K}(S) = \{M(S)/M$ an a-transducer$\}$. By the previous corollary

(1) U is obtained from elements in $\mathcal{K}(S)$ by a finite number of applications of $\cup$, $\cdot$, and $*$.

Since $U \subseteq a^* b^*$, the $*$ operations occurring in (1) can be applied only to subsets of a^* or b^*. Now S is context free, so that each set in $\mathcal{K}(S)$, and thus in $\hat{\mathcal{F}}(S)$, is context free. As is known [3], each context-free subset of a^* or b^* is regular. It is easily seen that each regular subset of a^* or b^* is in $\mathcal{K}(S)$. Thus $*$ can be eliminated from (1), i.e.,

(2) U can be obtained from elements of $\mathcal{K}(S)$ by a finite number of applications of $\cup$ and $\cdot$, thus by a finite number of products of sets in $\mathcal{K}(S)$.

Since $U \subseteq a^* b^*$, in each product all but one of the sets must be subsets of a^* or b^* and thus regular. As is easily seen, if V is regular and M is an a-transducer, then there exist a-transducers M_1 and M_2 such that $M_1(S) = M(S)V$ and $M_2(S) = VM(S)$. Thus the operation $\cdot$ can be eliminated, i.e.,

(3) there exist $n \geq 1$ and $L_1, \ldots, L_n$ in $\mathcal{K}(S)$ such that $U = \bigcup_{i=1}^{n} L_i$.

Since B is not regular, there exists an integer i(0) such that $L_{i(0)}$ is not regular. Since $L_{i(0)}$ is in $\mathcal{K}(S)$, there exists an a-transducer M such that $L_{i(0)} = M(S)$. Let h be the homomorphism of $\{a,b\}^*$ into a^* defined by $h(a) = a$ and $h(b) = \epsilon$. Suppose $h(L_{i(0)})$ is finite. Then $L_{i(0)} = \bigcup_{i=1}^{m} a^{r_i} R_i$, where $r_i \geq 0$ and $R_i \subseteq b^*$ is context free. Then R_i is regular, so that $L_{i(0)}$ is regular, a contradiction. Thus $h(L_{i(0)})$ is infinite. Hence

(4) there exists an a-transducer M such that $M(S) \subseteq U$ and $h(M(S))$ is infinite.

By a long and complicated argument (involving an intimate knowledge of context-free languages), which is not relevant to the present paper, we can show that (4) does not occur. We omit the details.

Another corollary of Theorem 4.1 which we obtain is a characterization of those AFL which can be derived from AFA.

Theorem 4.2. Let $\mathcal{F}$ be an AFL. Then there exists an AFA, $\mathcal{D}$, such that $\mathcal{F} = \mathcal{L}(\mathcal{D})$ if and only if $\mathcal{F}$ is countable and closed under arbitrary homomorphism.

Suppose that $\mathcal{F}$ is a countable AFL not closed under arbitrary homomorphism. The problem arises as to how to relate $\mathcal{F}$ in some way to a family of devices. One answer to that problem is now given.

Theorem 4.3. $\mathcal{F}(S \cup \{\{\epsilon\}\}) = \mathcal{L}^t(\mathcal{D}(S))$.

Corollary 1. If $\mathcal{F}$ is an AFL, then $\mathcal{F} = \mathcal{L}^t(\mathcal{D}(\mathcal{F}))$ if and only if $\mathcal{F}$ is countable and $\{\epsilon\}$ is in $\mathcal{F}$.

Corollary 2. $\mathcal{F}(S \cup \{\{\epsilon\}\})$ is the smallest family of sets containing the family $\{M(S)/S$ in $S \cup \{\{\epsilon\}\}$, M an ϵ-output limited a-transducer$\}$.

Section 5. Transducers

We now consider the devices which arise when an output structure is added to the acceptors in an AFA. We shall use these devices, called "transducers," to characterize (a) when $\mathcal{L}(\mathcal{D})$ and $\mathcal{L}^t(\mathcal{D})$ are closed under intersection with an arbitrary element of a given AFL, and (b) when a given set L is in $\mathcal{L}(\mathcal{D})$ or $\mathcal{L}^t(\mathcal{D})$.

Definition. Let $(\Omega, \mathcal{D})$, with $\Omega = (K, \Sigma, \Gamma, I, f, g, Z_0)$, be an AFA. Let $(\Omega, \mathcal{D}^0)$, or $\mathcal{D}^0$ when Ω is understood, be the set of all 9-tuples $M = (K_1, \Sigma_1, \Sigma_2, \Gamma_1, I_1, G, \delta, p_0, Z_0)$, called transducers, such that

(a) K_1, Σ_1, Σ_2, and Γ_1 are finite nonempty subsets of K, Σ, Σ, and Γ, respectively.

(b) I_1 and G are finite subsets of I and $\bigcup_{\gamma \text{ in } \Gamma_1^*} g(\gamma)$ respectively.

(c) p_0 is in K_1 and Z_0 in $\Gamma_1 \cap G$.

[12] Let $\epsilon^R = \epsilon$ and for each word $a_1 \ldots a_k$, $k \geq 1$, each a_i in Σ, let $(a_1 \ldots a_k)^R = a_k \ldots a_1$. The operation involved is called word reversal.

(d) δ is a function from $K_1 \times (\Sigma_1 \cup \{\epsilon\}) \times G$ into the finite subsets of $K_1 \times (\Gamma_1 \cup I_1)^* \times \Sigma_2^*$.

$\mathcal{D}^o$ is said to be an abstract family of transducers (abbreviated AFT).

K_1, Σ_1, Σ_2, and Γ_1 are called the set of states, inputs, outputs, and auxiliary symbols, respectively.

The notation for the movement of transducers is similar to that for acceptors.

Notation. Let $\vdash$ be the relation on $K_1 \times \Sigma_1^* \times \Gamma_1^* \times \Sigma_2^*$ defined as follows. For a in $\Sigma_1 \cup \{\epsilon\}$, $(p,aw,\gamma,z) \vdash (p',w,\gamma',z')$ if there exist $\bar{\gamma}$ in $g(\gamma) \cap G$, u, and y such that $z' = zy$, (p',u,y) is in $\delta(p,a,\bar{\gamma})$, and $f(\gamma,u) = \gamma'$. Let $\overset{k}{\vdash}$ and $\overset{*}{\vdash}$ be the relations on $K_1 \times \Sigma_1^* \times \Gamma_1^* \times \Sigma_2^*$ defined by $(p,aw,\gamma,z) \overset{0}{\vdash} (p,aw,\gamma,z)$, $(p,aw,\gamma,z) \overset{1}{\vdash} (p',w,\gamma',z')$ if $(p,aw,\gamma,z) \vdash (p',w,\gamma',zz')$, and for $k \geq 0$, $(p,w,\gamma,z) \overset{k+1}{\vdash} (p',w',\gamma',z')$ if there exists (p'',w'',γ'',z'') such that $(p,w,\gamma,z) \overset{k}{\vdash} (p'',w'',\gamma'',z'') \vdash (p',w',\gamma',z')$. Write $(p,w,\gamma,z) \overset{*}{\vdash} (p',w',\gamma',z')$ if $(p,w,\gamma,z) \overset{k}{\vdash} (p',w',\gamma',z')$ for some $k \geq 0$.

A transducer serves as a transformation device in the following way.

Notation. Let $\mathcal{D}^o$ be an AFT and let $M = (K_1,\Sigma_1,\Sigma_2,\Gamma_1,I_1,G,\delta,p_o,Z_o)$ be a transducer. For each w in Σ_1^* let

$$M(w) = \{z/(p_o,w,Z_o,\epsilon) \overset{*}{\vdash} (p,\epsilon,Z_o,z) \text{ for some p in } K_1\}.$$

For each $L \subseteq \Sigma_1^*$, let $M(L) = \bigcup_{w \text{ in } L} M(w)$.

By a discussion similar to the remarks preceding Theorem 3.1, we may assume that a given transducer satisfies the following: For each γ in G, there exists a word α_γ in $(\Gamma_1 \cup I_1)^*$ with the property that $f(\gamma',\alpha_\gamma) = \gamma'$ for all γ' such that $g(\gamma')$ contains γ. Without loss of generality, we shall assume that all acceptors and transducers selected from $\mathcal{D}$ and $\mathcal{D}^o$ have the above property.

Our first two theorems give characterizations of when $\mathcal{L}(\mathcal{D})$ and $\mathcal{L}^t(\mathcal{D})$ are closed under intersection with an arbitrary element in $\mathcal{F}$, $\mathcal{F}$ an AFL. The characterizations are in terms of transducers and pseudoinverse of transducers.

Lemma 5.1. Let $\mathcal{L}$ be an AFL. For each L in $\mathcal{L}$ and each regular set R, the set of "shuffles"

$$\text{Shuf}(L,R) = \{w_1y_1w_2y_2\ldots w_ny_n/n \geq 1, w_1\ldots w_n \text{ in L}, y_1\ldots y_n \text{ in R}\}$$

is in $\mathcal{L}$.

Lemma 5.2. Let $\mathcal{F}$ be an AFL and $\mathcal{D}$ an AFA.[13] Then $\mathcal{L}(\mathcal{D})$ is closed under intersection with all elements of $\mathcal{F}$ if and only if M(L) is in $\mathcal{L}(\mathcal{D})$ for all M in $\mathcal{D}^o$ and L in $\mathcal{F}$.

To bring a pseudoinverse of a transducer into our characterizations we need the next lemma.

Definition. M in $\mathcal{D}^o$ is ϵ-input bounded if there exists $k \geq 0$ such that for all p, p', γ, γ', z, and z', $(p,\epsilon,\gamma,z) \overset{\ell}{\vdash} (p',\epsilon,\gamma',z')$ implies $\ell \leq k$. M in $\mathcal{D}^o$ is ϵ-output bounded if there exists $k \geq 0$ such that for all p, p', w, w', γ, and γ', $(p,w,\gamma,\epsilon) \overset{\ell}{\vdash} (p',w',\gamma',\epsilon)$ implies $\ell \leq k$.

Thus M is ϵ-input bounded (ϵ-output bounded) if there exists an integer k so that M never has more than k consecutive moves with ϵ as input (ϵ as output).

Lemma 5.3. For each M in $\mathcal{D}^o$, there exists $\overline{M}$ in $\mathcal{D}^o$ such that z is in M(w) if and only if w is in $\overline{M}(z)$. (Thus $\overline{M}(w) = M^{pi}(w)$ for all w.) Furthermore, $\overline{M}$ is ϵ-input bounded if and only if M is ϵ-output bounded, and $\overline{M}$ is ϵ-output bounded if and only if M is ϵ-input bounded.

We are now able to present two characterizations of when $\mathcal{L}(\mathcal{D})$ is closed under intersection with arbitrary elements of $\mathcal{F}$.

Theorem 5.1. Let $\mathcal{F}$ be an AFL and $\mathcal{D}$ an AFA. Then the following three statements are equivalent:

(a) $\mathcal{L}(\mathcal{D})$ is closed under intersection with arbitrary elements of $\mathcal{F}$.

(b) M(L) is in $\mathcal{L}(\mathcal{D})$ for all M in $\mathcal{D}^o$ and all L in $\mathcal{F}$.

(c) $M^{pi}(L)$ is in $\mathcal{L}(\mathcal{D})$ for all M in $\mathcal{D}^o$ and all L in $\mathcal{F}$.

[13] It is to be understood that the AFL and AFA involved are actually $(\Sigma,\mathcal{F})$ and $(\Sigma,\mathcal{D})$.

Letting $\mathcal{F}$ be $\mathcal{L}(\mathcal{D})$ we get

Corollary. The following three statements are equivalent for each $\mathcal{L}(\mathcal{D})$.

(a) $\mathcal{L}(\mathcal{D})$ is closed under intersection.

(b) $M(L)$ is in $\mathcal{L}(\mathcal{D})$ for all M in $\mathcal{D}^o$ and all L in $\mathcal{L}(\mathcal{D})$.

(c) $M^{pi}(L)$ is in $\mathcal{L}(\mathcal{D})$ for all M in $\mathcal{D}^o$ and all L in $\mathcal{L}(\mathcal{D})$.

Turning to $\mathcal{L}^t(\mathcal{D})$ we have

Theorem 5.2. Let $\mathcal{F}$ be an AFL and $\mathcal{D}$ an AFA. Then the following three statements are equivalent:

(a) $\mathcal{L}^t(\mathcal{D})$ is closed under intersection with arbitrary elements of $\mathcal{F}$.

(b) $M(L)$ is in $\mathcal{L}^t(\mathcal{D})$ for all L in $\mathcal{F}$ and all ϵ-output bounded M in $\mathcal{D}^o$.

(c) $M^{pi}(L)$ is in $\mathcal{L}^t(\mathcal{D})$ for all L in $\mathcal{F}$ and all ϵ-input bounded M in $\mathcal{D}^o$.

Using transducers and the pseudoinverse of transducers we now give two characterizations of L being in $\mathcal{L}^t(\mathcal{D})$.

Theorem 5.3. The following three statements are equivalent for each $\mathcal{L}(\mathcal{D})$ and each set L:

(a) L is in $\mathcal{L}(\mathcal{D})$.

(b) $L = M(R)$ for some M in $\mathcal{D}^o$ and some regular set R.

(c) $L = M^{pi}(R)$ for some M in $\mathcal{D}^o$ and some regular set R.

Theorem 5.4. The following three statements are equivalent for each $\mathcal{L}^t(\mathcal{D})$ and each set L:

(a) L is in $\mathcal{L}^t(\mathcal{D})$.

(b) $L = M(R)$ for some ϵ-output bounded transducer M in $\mathcal{D}^o$ and some regular set R.

(c) $L = M^{pi}(R)$ for some ϵ-input bounded transducer M in $\mathcal{D}^o$ and some regular set R.

We now turn to the notion of a deterministic transducer and its effect upon regular sets. Intuitively speaking, a deterministic transducer M is one in which for each state, each input, and each auxiliary storage configuration, M has at most one next move. This, of course, does not guarantee that $M(w)$ is at most single valued for each w. Thus if

$$(p_o,w,Z_o,\epsilon) \,|\!\frac{*}{M}\, (p,\epsilon,Z_o,z_1) \,|\!\frac{}{M}\, (p',\epsilon,Z_o,z_1z_2)$$

then both z_1 and z_1z_2 are in $M(w)$.

We now formulate the notion of a deterministic transducer.

Definition. Let $(\Omega,\mathcal{D}^o)$ be an AFT. Let $(\Omega,\mathcal{D}^o_{det})$, or $\mathcal{D}^o_{det}$ when Ω is understood, be the family of all transducers $M = (K_1,\Sigma_1,\Delta_1,\Gamma_1,I_1,G,\delta,p_o,Z_o)$ with the following properties: For each (p,a) in $K_1 \times (\Sigma_1 \cup \{\epsilon\})$ let $G(p,a) = \{\gamma' \text{ in } G/\delta(p,a,\gamma') \neq \emptyset\}$. For each (p,a,γ) in $K_1 \times \Sigma_1 \times \Gamma_1^*$,

(a) if $g(\gamma) \cap G(p,\epsilon) \neq \emptyset$, then there exists γ' in G such that $g(\gamma) \cap G(p,\epsilon) = \{\gamma'\}$, $g(\gamma) \cap G(p,b) = \emptyset$ for all b in Σ_1, and $\delta(p,\epsilon,\gamma')$ contains exactly one element;

(b) if $g(\gamma) \cap G(p,a) \neq \emptyset$, then there exists γ' in G such that $g(\gamma) \cap G(p,a) = \{\gamma'\}$ and $\delta(p,a,\gamma')$ contains exactly one element.

Each M in $\mathcal{D}^o_{det}$ is said to be a deterministic transducer.

Conditions (a) and (b) guarantee that the acceptor has at most one next move. That is, if $(p,a,\gamma,\epsilon) \vdash (p',b,\gamma',v)$ and $(p,a,\gamma,\epsilon) \vdash (p'',b'',\gamma'',v'')$, then $p' = p''$, $b = b''$, $\gamma' = \gamma''$, and $v = v''$. However, $M(w)$ may still be multi-valued.

In the spirit of Theorem 5.3, we now characterize $\mathcal{L}(\mathcal{D})$ in terms of $\mathcal{D}^o_{det}$ and regular sets.

Theorem 5.5. L is in $\mathcal{L}(\mathcal{D})$ if and only if $L = M(R)$ for some regular set R and some (ϵ-input bounded) M in $\mathcal{D}^o_{det}$, with each $M(w)$ containing at most one element.

Corollary. L is in $\mathcal{L}^t(\mathcal{D})$ if and only if there exist a regular set R and an ϵ-input bounded, ϵ-output bounded M in $\mathcal{D}^o_{det}$, with $M(w)$ at most single valued for each w, such that $L = M(R)$.

Theorem 5.5 has already been proved elsewhere [2] for the special cases of finite state acceptors, pushdown acceptors, and Turing acceptors.

In conclusion, we remark that by appropriate coding and introduction of additional states, we can make the set R be $\{a,b\}^*c$ (a, b, and c distinct symbols) in (b) and (c) of both Theorems 5.3 and 5.4, and in Theorem 5.5 and its corollary.

Bibliography

[1] Elgot, C. C. "Decision problems of finite automata design and related arithmetics," Transactions of the American Mathematical Society, Vol. 98, 1961, pp. 21-51.

[2] Evey, R. J. "The theory and application of pushdown store machines," Mathematical Linguistics and Automatic Translation Report No. NSF-10, The Computation Laboratory of Harvard University, May 1963.

[3] Ginsburg, S. "The mathematical theory of context-free languages," McGraw-Hill Book Company, New York, 1966.

[4] Ginsburg, S. and Greibach, S. "Mappings which preserve context-sensitive languages," Information and Control, Vol. 9, 1966, pp. 563-582.

[5] Ginsburg, S., Greibach, S., and Harrison, M. "Stack automata and compiling," Journal of the Association for Computing Machinery, Vol. 14, 1967, pp. 172-201.

[6] Ginsburg, S., Greibach, S., and Harrison, M. "One-way stack automata," Journal of the Association for Computing Machinery, Vol. 14, 1967, pp. 389-418.

[7] Ginsburg, S. and Harrison, M. "One-way real-time list-storage languages," System Development Corporation Report TM-738/029/00, January 1967.

[8] Ginsburg, S. and Rose, G. F. "Preservation of languages by transducers," Information and Control, Vol. 9, 1966, pp. 153-176.

[9] Ginsburg, S. and Spanier, E. H. "Finite-turn pushdown automata," SIAM Journal on Control, Vol. 4, 1966, pp. 429-453.

[10] Hartmanis, J., Lewis, II, P. M., and Stearns, R. F. "Classifications of computations by time and memory requirements," Proceedings of the IFIP Congress 65, Spartan Books, Inc., 1965, pp. 31-35.

[11] Kuroda, S. Y. "Classes of languages and linear-bounded automata," Information and Control, Vol. 7, 1964, pp. 207-223.

[12] Landweber, P. S. "Three theorems on phrase structure grammars of type 1," Information and Control, Vol. 6, 1963, pp. 131-136.

[13] Rabin, M. O. and Scott, D. "Finite automata and their decision problems," IBM Journal of Research and Development, Vol. 3, 1959, pp. 114-125.