JOURNAL OF COMPUTER AND SYSTEM SCIENCES 4, 308–338 (1970)

Principal AFL*

SEYMOUR GINSBURG**

University of Southern California

AND

SHEILA GREIBACH***

University of California, Los Angeles

Received May 15, 1969

A (full) principal AFL is a (full) AFL generated by a single language, i.e., it is the smallest (full) AFL containing the given language. In the present paper, a study is made of such AFL. First, an AFA (abstract family of acceptors) characterization of (full) principal AFL is given. From this result, many well-known families of AFL can be shown to be (full) principal AFL. Next, two representation theorems for each language in a (full) principal AFL are given. The first involves the generator and one application each of concatenation, star, intersection with a regular set, inverse homomorphism, and a special type of homomorphism. The second involves an a-transducer, the generator, and one application of concatenation and star. Finally, it is shown that if $\mathscr{L}_1$ and $\mathscr{L}_2$ are (full) principal AFL, then so are (a) the smallest (full) AFL containing $\{L_1 \cap L_2 / L_1 \text{ in } \mathscr{L}_1, L_2 \text{ in } \mathscr{L}_2\}$ and (b) the family obtained by substituting ϵ-free languages of $\mathscr{L}_2$ into languages of $\mathscr{L}_1$.

INTRODUCTION

In an earlier paper [4], we abstracted a number of closure properties common to many families of formal languages studied in computer science. These closure properties, union, concatenation, ϵ-free Kleene closure, intersection with regular sets, inverse homomorphism, and ϵ-free homomorphism, became the axioms for a family of languages called an AFL (short for "abstract family of languages"). If an AFL was also closed under arbitrary homomorphism, then it was called a full AFL. It was

* Research sponsored in part by the Air Force Cambridge Research Laboratories, Office of Aerospace Research, USAF, under contract F1962870C0023, and by the Air Force Office of Scientific Research, Office of Aerospace Research, USAF, under AFOSR Grant No. AF-AFOSR-1203-67A.

** Also with System Development Corporation.

*** Also with System Development Corporation. Part of this work was done at Harvard University.

then show that AFL and full AFL coincide with the families of languages related to "natural" families of (one-way) non-deterministic acceptors. It was also shown that each (full) AFL has certain additional properties customarily proved for each new family of languages introduced into the literature. Thus AFL and full AFL are unifying concepts in language theory as it relates to machines.

Since [4], a series of papers have been written [5, 7, 9, 10, 11, 12], dealing either with AFL theory *per se*, or with problems about languages rendered prominent by the discovery of AFL. The present article deals further with AFL theory. In considering potential theorems, there has always been a problem of obtaining counterexamples, i.e., finding (full) AFL with desired properties. (The difficulty is that a (full) AFL is a rather complicated family of languages from the point of view of considering all members in it.) In most instances to date, well-known (full) AFL previously considered in other computer science studies have been used. On a few occasions, however, new (full) AFL have been constructed, usually after much travail. These improvised (full) AFL were described as "the smallest (full) AFL containing L," where L was some explicit set. The purpose of the present work is to study such (full) AFL, i.e., (full) principal AFL. In turns out that many of the well-known families of languages are (full) principal AFL, e.g., the regular sets, the context-free languages, the recursively enumerable sets, the one-way stack languages, and the nested stack languages (the last three noted here for the first time). Thus (full) principal AFL are not extremely special, esoteric families of languages, but include many of the most important ones.

The paper itself is divided into four sections. Section 1 defines the language concepts to be used and presents several elementary results about (full) principal AFL.

Section 2 concerns acceptors. It reviews the concept of an AFA (abstract family of acceptors) and then gives a necessary and sufficient condition on an AFA in order for it to define a (full) principal AFL. The machine characterization is so easy to apply that many well-known families of languages can be proved (full) principal as an immediate consequence.

Section 3 is concerned with two simple representations for each language in a (full) principal AFL with generator L. The first result is that for $L \neq \phi$, each language in the [full] AFL can be represented in the form $M((Lc)^*)$ or $M((Lc)^+)$ $[M((Lc)^*)]$, where M is an appropriate type of a-transducer and c is a new symbol, depending on whether or not L contains the empty word. The second result is that each language in the [full] AFL can be represented in the form $h_2(h_1^{-1}((Lc)^*) \cap R)$, where c is a new symbol, R is a regular set, and h_1 and h_2 are appropriate homomorphisms. It is thus similar to the Chomsky-Schutzenberger theorem about the representation of context-free languages in terms of the Dyck set on two letters.

Section 4 considers the effect of two operators, "$\wedge$" and "substitution" on (full) principal AFL. The main results are that if $\mathscr{L}_1$ and $\mathscr{L}_2$ are (full) principal AFL, then so are (a) the smallest (full) AFL containing $\{L_1 \cap L_2 / L_1 \text{ in } \mathscr{L}_1, L_2 \text{ in } \mathscr{L}_2\}$ and (b) the family obtained by substituting ϵ-free languages of $\mathscr{L}_2$ into languages of $\mathscr{L}_1$.

While the emphasis throughout the paper is on theory, we have tried to focus attention on results which would be meaningful for AFL already in the literature. Accordingly, we have presented applications to well-known AFL whenever feasible. It is our conviction that there is still much basic work to be done on (full) principal AFL.

Section 1. Preliminaries

In this section we first review some concepts about families of languages. We then introduce the concept of concern to us in this paper, namely principal AFL and (full) principal AFL. Finally, we present several elementary results about these AFL.

We now recall the concepts of "language" and "family of languages."

Definition. A *language* is a set L for which there exists a finite set Σ_1 of abstract symbols such that[1] $L \subseteq \Sigma_1{}^*$. For each language L let Σ_L be the smallest set Σ_1 such that $L \subseteq \Sigma_1{}^*$.

Definition. A *family of languages* is a pair $(\Sigma, \mathscr{L})$, or $\mathscr{L}$ when Σ is understood, where

(1) Σ is an infinite set of symbols,

(2) each L in $\mathscr{L}$ is a language, with $\Sigma_L \subseteq \Sigma$, and

(3) $L \neq \phi$ for some L in $\mathscr{L}$.

Henceforth, Σ will always denote a given infinite set of symbols, and Σ with a subscript a finite subset of Σ. All symbols given or constructed, and then used in a language, will be assumed to be in Σ. Also, L and $\mathscr{L}$, with or without a subscript, will always denote a language and a family of languages, respectively.

The notion of a family of languages is usually too general a concept to obtain significant results. In [4], families of languages having several additional properties were introduced and shown to be fruitful for the study of families of languages treated in computer science. These families of languages, called AFL, are the following:

Definition. An *abstract family of languages* (AFL) is a family of languages closed under the operations of union, concatenation, $+$,[2] ϵ-free homomorphism,[3]

[1] For each set Σ_1, $\Sigma_1{}^*$ is the free semigroup with identity ϵ generated by Σ_1. Each element of $\Sigma_1{}^*$ is called a *word* of $\Sigma_1{}^*$.

[2] $A^+ = \bigcup_{i \geqslant 1} A^i$, where $A^{i+1} = A^i A$ for each $i \geqslant 1$. $A^* = A^+ \cup \{\epsilon\}$.

[3] A mapping h from $\Sigma_1{}^*$ into $\Sigma_2{}^*$ is a *homomorphism* if $h(xy) = h(x)\, h(y)$ for all x and y in $\Sigma_1{}^*$. If $h(x) = \epsilon$ implies $x = \epsilon$, then h is said to be *ϵ-fre*

inverse homomorphism,[4] and intersection with regular sets.[5] A *full* AFL is an AFL closed under arbitrary homomorphism.

The reader is referred to [4] for motivation and details on AFL.

In considering AFL, we are frequently interested in describing specific AFL for either illustrative or counterexample purposes. Several methods are popular. One is by a family of grammars (the left-linear, context-free, or context-sensitive grammars). Another is by a family of acceptors (finite-state acceptors or one-way stack acceptors). And a third is "as the smallest AFL containing a given family $\mathscr{S}$ of languages." This leads to the following concept.

Notation. Let Σ be given. For each set of languages $\mathscr{S}$, let $\mathscr{F}(\mathscr{S})$ ($\hat{\mathscr{F}}(\mathscr{S})$) be the smallest (full) AFL containing $\mathscr{S}$.

For each $\mathscr{S}$, $\mathscr{F}(\mathscr{S})$ and $\hat{\mathscr{F}}(\mathscr{S})$ exist.

Important occurrences of $\mathscr{F}(\mathscr{S})$ and $\hat{\mathscr{F}}(\mathscr{S})$ arise when $\mathscr{S}$ contains exactly one element, i.e., $\mathscr{S} = \{L\}$ for some L. (In this case, we write $\mathscr{F}(L)$ instead of $\mathscr{F}(\{L\})$ and $\hat{\mathscr{F}}(L)$ instead of $\mathscr{F}(\{L\})$. This is the situation we shall study in the remainder of the paper. Accordingly, we introduce the following concepts.

DEFINITION. An AFL $\mathscr{L}$ is said to be (*full*) *principal* if there exists a language L such that $\mathscr{L} = \mathscr{F}(L)$ ($\mathscr{L} = \hat{\mathscr{F}}(L)$). Then L is said to be a (*full*) *generator* of $\mathscr{L}$.

Obviously the regular sets form a full principal AFL and the ϵ-free regular sets a principal AFL. We shall see that many of the families of formal languages studied in computer science are principal and/or full principal.

EXAMPLE 1.1. The Chomsky-Schutzenberger theorem asserts ([3], Theorem 3) that given Σ_1 there exist Σ_2, a Dyck set $D \subseteq \Sigma_2^*$, and a homomorphism h from Σ_2^* onto Σ_1^* which satisfy the property that for each context-free language $L \subseteq \Sigma_1^*$ a regular set $R \subseteq \Sigma_2^*$ can be found such that $h(D \cap R) = L$. From this it readily follows that the context-free languages form a full principal AFL, fully generated by the Dyck set on two letters. (The Dyck set on one letter fully generates the one-counter languages [9].) We shall see later (Example 2.1) that the context-free languages form a princepal AFL.

EXAMPLE 1.2. The AFL $\mathscr{L}$ consisting of all recursive sets is not principal. For, suppose $\mathscr{L}$ is principal. Then $\mathscr{L} = \mathscr{F}(L_0)$ for some L_0 in $\mathscr{L}$. Hence, there exists a total recursive, strictly increasing function $T(n)$, constructible in the sense of [13],

[4] If h is a homomorphism from Σ_1^* into Σ_2^*, then the mapping h^{-1} of subsets of Σ_2^* into subsets of Σ_1^* defined by $h^{-1}(Y) = \{x \mid h(x) \text{ in } Y\}$ for all $Y \subseteq \Sigma_2^*$ is called an *inverse homomorphism.*

[5] A *regular* set is any set contained in the least class which contains each finite set of words and which is closed under union, concatenation, and *.

such that L_0 is accepted in $T(n)$ tape space (on a Turing machine). It is straightforward to show that for each L in $\mathscr{F}(L_0)$, there exists $k \geq 1$ such that L is accepted in $T(kn)$ tape space. But there are recursive languages L accepted in $2^{T(n^2)}$ tape space but not in $T(kn)$ tape space for any k. Therefore $\mathscr{L}$ is not principal. (Obviously $\mathscr{L}$ is not full principal, since $\hat{\mathscr{F}}(\mathscr{L})$ is the family of recursively enumerable (r.e.) sets.)

Clearly an AFL that is principal and full is full principal.

An obvious generalization of (full) principal AFL is "(full) finitely generable," i.e., $\mathscr{L} = \mathscr{F}(\mathscr{S})$ ($\mathscr{L} = \hat{\mathscr{F}}(\mathscr{S})$) for some finite set $\mathscr{S}$. We shall see below that such AFL coincide with (full) principal AFL.

LEMMA 1.1. *Let L_1 and L_2 be languages and c a symbol not in $\Sigma_{L_1} \cup \Sigma_{L_2}$. Then $\mathscr{F}(\{L_1, L_2\}) = \mathscr{F}(L)$, where $L = L_1 \cup cL_2 \cup (L_2 \cap \{\epsilon\})$, and $\hat{\mathscr{F}}(\{L_1, L_2\}) = \hat{\mathscr{F}}(L_1 \cup cL_2)$.*

Proof. We give the proof for $\mathscr{F}(\{L_1, L_2\})$, the argument $\hat{\mathscr{F}}(\{L_1, L_2\})$ being similar. Since L is in $\mathscr{F}(\{L_1, L_2\})$, $\mathscr{F}(L) \subseteq \mathscr{F}(\{L_1, L_2\})$. It thus suffices to show that L_1 and L_2 are in $\mathscr{F}(L)$. If L_1 contains ϵ, then $L_1 = L \cap \Sigma_{L_1}^*$. Otherwise, $L_1 = L \cap \Sigma_{L_1}^+$. In either case, L_1 is in $\mathscr{F}(L)$. Let h be the homomorphism on $\Sigma_{L_2} \cup \{c\}$ defined by $h(c) = \epsilon$ and $h(a) = a$ for each a in Σ_{L_2}. Then $L_2 = h(L \cap c\Sigma_{L_2}^+)$ if ϵ is not in L_2 and $L_2 = h(L \cap (c\Sigma_{L_2}^+ \cup \{\epsilon\}))$ if ϵ is in L_2. Now h is ϵ-limited[6] on $L \cap (c\Sigma_{L_2}^+ \cup \{\epsilon\})$ and hence on any subset. Therefore $L_2 - \{\epsilon\}$ is in $\mathscr{F}(L)$ by Corollary 5 of Theorem 2.1 of [4]. If L_2 contains ϵ, then so does L and hence $\{\epsilon\} = L \cap \{\epsilon\}$ is in $\mathscr{F}(L)$. Then $L_2 = (L_2 - \{\epsilon\}) \cup \{\epsilon\}$ is in $\mathscr{F}(L)$. If L_2 does not contain $\{\epsilon\}$, then $L_2 = L_1 - \{\epsilon\}$ is in $\mathscr{F}(L)$. In either case, L_2 is in $\mathscr{F}(L)$.

Remark. A similar argument shows that if $L_1 \neq \phi$ and $L_2 \neq \phi$, then $L_1cL_2 \cup ((L_1 \cup L_2) \cap \{\epsilon\})$ generates $\mathscr{F}(\{L_1, L_2\})$ and L_1cL_2 fully generates $\hat{\mathscr{F}}(\{L_1, L_2\})$.

From Lemma 1.1, there immediately follows

THEOREM 1.1. *An* AFL *$\mathscr{L}$ is (full) principal if and only if there exists a finite set $\mathscr{S}$ of languages such that $\mathscr{L} = \mathscr{F}(\mathscr{S})$ ($\mathscr{L} = \hat{\mathscr{F}}(\mathscr{S})$).*

COROLLARY. *If $\mathscr{L}_1, \ldots, \mathscr{L}_n$ are (full) principal* AFL, *then so is $\mathscr{F}(\mathscr{L}_1 \cup \cdots \cup \mathscr{L}_n)$ ($\hat{\mathscr{F}}(\mathscr{L}_1 \cup \cdots \cup \mathscr{L}_n)$).*

Consider the existence of nonprincipal AFL. If Σ is countable, then the family of regular subsets of Σ is countable and the family of homomorphisms defined on finite subsets of Σ is countable. Thus every member of $\mathscr{F}(L)$ ($\hat{\mathscr{F}}(L)$) can be obtained from L by a finite number of operations, each chosen from a countable collection of operations. Hence $\mathscr{F}(L)$ and $\hat{\mathscr{F}}(L)$ are countable. In other words, if Σ is countable then no uncountable AFL over Σ is either principal or full principal.

[6] A homomorphism h on $\Sigma_1{}^*$ is *ϵ-limited* on $L \subseteq \Sigma_1{}^*$ if there exist $k \geq 0$ such that for all w in L, if $w = xyz$ and $h(y) = \epsilon$, then $|y| < k$.

Another situation yielding an AFL which is not principal is the following special case of a well-known result for algebraic systems.

LEMMA 1.2. *Let $\mathscr{L}_1, \ldots, \mathscr{L}_n, \ldots$ be an infinite sequence of (full)* AFL *such that $\mathscr{L}_n \not\subseteq \mathscr{L}_{n+1}$ for each n. Then $\bigcup_n \mathscr{L}_n$ is a (full)* AFL *which is not (full) principal.*

[*Proof.* Obviously $\mathscr{L} = \bigcup_n \mathscr{L}_n$ is a (full) AFL. Suppose $\mathscr{L}$ is (full) principal, i.e., $\mathscr{L} = \mathscr{F}(L)$ ($\mathscr{L} = \hat{\mathscr{F}}(L)$) for some L. There exists n_0 such that L is in $\mathscr{L}_{n_0}$. Then $\mathscr{L} = \mathscr{F}(L)$ ($\mathscr{L} = \hat{\mathscr{F}}(L)) \subseteq \mathscr{L}_{n_0} \subseteq \mathscr{L}_{n_0+1} \subseteq \mathscr{L}$, so that $\mathscr{L}_{n_0} = \mathscr{L}_{n_0+1}$, a contradiction.]

From Lemma 1.2 and Theorem 1.1 we get

THEOREM 1.2. *Let $\mathscr{L}$ be a countable (full)* AFL. *Then $\mathscr{L}$ is not (full) principal if and only if there exists an infinite sequence of (full) $\mathscr{L}_1, \ldots, \mathscr{L}_n, \ldots$ such that $\mathscr{L} = \bigcup_n \mathscr{L}_n$ and $\mathscr{L}_n \not\subseteq \mathscr{L}_{n+1}$ for each n.*

Proof. By Lemma 1.2, it suffices to consider the "only if." Suppose $\mathscr{L}$ is not (full) principal. Since $\mathscr{L}$ is countable, $\mathscr{L} - \{\phi\} = \{L_1, \ldots, L_n, \ldots\}$. Let $i_1 = 1$. Continuing by induction, for each $m < r$ suppose that i_m exists and that $\mathscr{L}_m = \mathscr{F}(\{L_{i_1}, \ldots, L_{i_m}\})$ ($\mathscr{L}_m = \hat{\mathscr{F}}(\{L_{i_1}, \ldots, L_{i_m}\})$). Furthermore, suppose that $\mathscr{L}_m \not\subseteq \mathscr{L}_{m+1}$ for each m, $1 \leqslant m < r - 1$. Since $\mathscr{L}$ is not (full) principal, thus by Theorem 1.1, not (fully) finitely generable, there exists a smallest integer i_r such that L_{i_r} is not in $\mathscr{L}_{r-1}$. Let $\mathscr{L}_r = \mathscr{F}(\{L_{i_1}, \ldots, L_{i_r}\})$ ($\mathscr{L} = \hat{\mathscr{F}}(\{L_{i_1}, \ldots, L_{i_r}\})$). Then $\mathscr{L}_{r-1} \not\subseteq \mathscr{L}_r$. Thus the induction is extended. Obviously $\mathscr{L} = \bigcup_{m \geqslant 1} \mathscr{L}_m$.

The method of proof also shows

COROLLARY. *Let $\mathscr{L}$ be a countable (full)* AFL. *Then $\mathscr{L}$ is not a (full) principal* AFL *if and only if there exists an infinite sequence $\mathscr{L}_1, \ldots, \mathscr{L}_n, \ldots$ of (full) principal* AFL *such that $\mathscr{L} = \bigcup_n \mathscr{L}_n$ and for each n, $\mathscr{L}_n \not\subseteq \mathscr{L}_{n+1}$.*

As an application of Theorem 1.2, we have

EXAMPLE 1.3. For each $n > 0$ let $\mathscr{S}_n$ be the ultralinear sets of rank n [6] and $\mathscr{L}_n = \hat{\mathscr{F}}(\mathscr{S}_n)$. In particular, $\mathscr{S}_n$ is infinite. It is shown in [9] that $\mathscr{L}_n \not\subseteq \mathscr{L}_{n+1}$ for each n. Then $\mathscr{L} = \bigcup_n \mathscr{L}_n$ is a full AFL which, by Theorem 1.2, is neither principal nor full principal. Each of the families $\mathscr{L}_n$ is full principal [9]. For example, $\mathscr{L}_1$ is fully generated by[7] $\{wcw^R/w \text{ in } \{a, b\}^*\}$.

[7] Let $\epsilon^R = \epsilon$. For $a_1, \ldots, a_k$ in Σ_1, $k \geqslant 1$, let $(a_1 \cdots a_k)^R = a_k \cdots a_1$.

Section 2. Machine Characterization

In [4], we proved that full AFL (AFL) are the families of languages accepted by families of one-way (quasi-realtime) nondeterministic acceptors. In this section we show that the property of being (full) principal has a machine counterpart. In particular, we prove that an AFL $\mathscr{L}$ is (full) principal if and only if some family of acceptors which accepts $\mathscr{L}$ meets a certain condition. Using this characterization result, we are then able to demonstrate that a number of well-known (full) AFL are (full) principal.

We first recall some concepts about abstract families of acceptors.

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

(1) Ω is a 6-tuple $(K, \Sigma, \Gamma, I, f, g)$, where

(a) K and Σ are infinite abstract sets, and Γ and I are abstract sets, with Γ and I nonempty.

(b) f is a mapping from $\Gamma^* \times I$ into $\Gamma^* \cup \{\phi\}$.

(c) g is a function from Γ^* into the finite subsets of Γ^* such that $g(\epsilon) = \{\epsilon\}$, and ϵ is in $g(\gamma)$ if and only if $\gamma = \epsilon$.

(d) For each γ in $g(\Gamma^*)$, there exists an element 1_γ in I having the property that $f(\gamma', 1_\gamma) = \gamma'$ for all γ' such that $g(\gamma')$ contains γ.

(e) For each u in I, there exists a finite set $\Gamma_u \subseteq \Gamma$ with the following property: If $\Gamma_1 \subseteq \Gamma$, γ is in $\Gamma_1{}^*$, and $f(\gamma, u) \neq \phi$, then $f(\gamma, u)$ is in $(\Gamma_1 \cup \Gamma_u)^*$.

(2) $\mathscr{D}$ is the family of all elements (called *acceptors*) $D = (K_1, \Sigma_1, \delta, q_0, F)$, where

(a) K_1 and Σ_1 are finite subsets of K and Σ resp., F is a subset of K_1, and q_0 is in K_1.

(b) δ is a function from $K_1 \times (\Sigma_1 \cup \{\epsilon\}) \times g(\Gamma^*)$ into the finite subsets of $K_1 \times I$ such that the set

$$G_D = \{\gamma \mid \delta(q, a, \gamma) \neq \phi \text{ for some } q \text{ and } a\}$$

is finite.

Intuitively speaking, K is the set of all possible "states," Σ is the set of all possible "inputs," Γ is the set of all possible "auxiliary" symbols (i.e., symbols used to form the auxiliary storage configurations), and I is the set of all possible "instructions." The function f selects the next auxiliary storage configuration. The function g is a generalized read head which examines the auxiliary storage configuration to determine the symbol or symbols being scanned. p_0 is the "start" state of the acceptor and F is the set of "accepting" states. δ is the "move" function. The reader is referred to [4] for further details and explanation.

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

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

$$(p, w, \gamma) \overset{k}{\vdash} (p'', w'', \gamma'') \vdash (p', w', \gamma').$$

Let $\overset{*}{\vdash}$ be the transitive, reflexive extension of $\vdash$, i.e., $(p, w, \gamma) \overset{*}{\vdash} (p', w', \gamma')$ if $(p, w, \gamma) \overset{k}{\vdash} (p', w', \gamma')$ for some $k \geqslant 0$.

An acceptor yields a set of words as follows.

Definition. *Let* $\mathscr{D}$ be an AFA. For each acceptor $D = (K_1, \Sigma_1, \delta, q_0, F)$, let $L(D)$, called the *set accepted by* D, be the set of words

$$\{w \text{ in } \Sigma_1^*/(p_0, w, \epsilon) \overset{*}{\vdash} (p, \epsilon, \epsilon) \text{ for some } p \text{ in } F\}.$$

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

Another concept from [4] of concern to us is the following.

Definition. Let k be a nonnegative integer and $\mathscr{D}$ an AFA. Let $\mathscr{D}_k{}^t$ be the set of all D in $\mathscr{D}$ such that $(p, \epsilon, \gamma) \overset{l}{\vdash} (p', \epsilon, \gamma')$ implies $l \leqslant k$. Let $\mathscr{L}^t(\mathscr{D}) = \bigcup_{k=0}^{\infty} \mathscr{L}(\mathscr{D}_k{}^t)$. Each L in $\mathscr{L}^t(\mathscr{D})$ is called *quasi-realtime.*

The basic connections between AFA and AFL are the following results proved in [4]:

(A) For each AFA $\mathscr{D}$, $\mathscr{L}^t(\mathscr{D})$ is an AFL containing $\{\epsilon\}$ and $\mathscr{L}(\mathscr{D})$ is a full AFL.

(B) For each AFL $\mathscr{L}$ containing $\{\epsilon\}$ (full AFL $\mathscr{L}$), there exists an AFA $\mathscr{D}$ such that $\mathscr{L} = \mathscr{L}^t(\mathscr{D})$ $(\mathscr{L} = \mathscr{L}(\mathscr{D}))$.

We now turn to the problem of characterizing (full) principal AFL by AFA. First we need several definitions.

Definition. Let $(\Omega', \mathscr{D}')$ and $(\Omega, \mathscr{D})$ be AFA, with $\Omega' = (K', \Sigma, \Gamma', I', f', g')$ and $\Omega = (K, \Sigma, \Gamma, I, f, g)$. Then $(\Omega', \mathscr{D}')$ is said to be a sub-AFA of $(\Omega, \mathscr{D})$ if

(a) $K' \subseteq K$, $\Gamma' \subseteq \Gamma$, and $I' \subseteq I$;
(b) f' is the function f restricted to $\Gamma'^* \times I'$;
(c) for each u in I', $\Gamma_u' = \Gamma_u \cap \Gamma'$; and
(d) there exists $H \subseteq \Gamma^*$ such that $g'(\gamma) = g(\gamma) \cap H$ for all γ in Γ'^*.

DEFINITION. $\mathscr{D}$ is *finitely* (*t*-) *encodable* if there exists a sub-AFA $\mathscr{D}'$ such that $\mathscr{L}(\mathscr{D}) = \mathscr{L}(\mathscr{D}')$ ($\mathscr{L}^t(\mathscr{D}) = \mathscr{L}^t(\mathscr{D}')$) and I' and $g'(\Gamma'^*)$ are finite.

We now present a sequence of lemmas leading to the main result that an AFL containing $\{\epsilon\}$ is (full) principal if and only if $\mathscr{L} = \mathscr{L}^t(\mathscr{D})$ ($\mathscr{L} = \mathscr{L}(\mathscr{D})$) for some finitely t-encodable AFA (finitely encodable AFA) $\mathscr{D}$.

LEMMA 2.1. *If $\mathscr{L}^t(\mathscr{D})$ ($\mathscr{L}(\mathscr{D})$) is (full) principal, then $\mathscr{D}$ is finitely t-encodable (finitely encodable).*

Proof. We shall prove the result for $\mathscr{L}(\mathscr{D})$, an analogous argument holding for $\mathscr{L}^t(\mathscr{D})$.

Let $\mathscr{L}(\mathscr{D})$ be full principal. Thus $\mathscr{L}(\mathscr{D}) = \hat{\mathscr{F}}(L)$ for some L in $\mathscr{L}(\mathscr{D})$. Then $L = L(D)$ for some acceptor $D = (K_1, \Sigma_1, \delta, q_0, F)$. Let

$$I_1 = \{u \text{ in } I/(p, u) \text{ in } \delta(q, a, \gamma) \text{ for some } p, q, a, \gamma\} \cup \{1_\gamma \mid \gamma \text{ in } G_D\}.$$

Since δ is finite-valued and G_D is finite, I_1 is finite. Let Γ_1 be the smallest subset of Γ such that $\Gamma_u \subseteq \Gamma_1$ for each u in I_1 and such that $G_D \subseteq \Gamma_1^*$. Clearly Γ_1 exists and is finite. Let f' be the function f restricted to $\Gamma_1^* \times I_1$. For each γ in Γ_1^*, let $g'(\gamma) = g(\gamma) \cap G_D$. Let $(\Omega', \mathscr{D}')$ be the AFA where $\Omega' = (K, \Sigma, \Gamma_1, I_1, f', g')$. Then $\mathscr{L}(\mathscr{D}')$ is a full AFL containing L, so that $\hat{\mathscr{F}}(L) = \mathscr{L}(\mathscr{D}) \supseteq \mathscr{L}(\mathscr{D}') \supseteq \hat{\mathscr{F}}(L)$. Therefore $\mathscr{L}(\mathscr{D}) = \mathscr{L}(\mathscr{D}')$. Since I_1 and $g'(\Gamma_1^*) \subseteq G_D$ are finite, $(\Omega, \mathscr{D})$ is finitely encodable.

The next lemma concerns special sets and is used in the proof of the main result. First though, we need some additional notation.

Notation. Let $(\Omega, \mathscr{D})$ be an AFA, with $\Omega = (K, \Sigma, \Gamma, I, f, g)$. For each $n \geq 0$, let F^n be the function on $\Gamma^* \times I \times \cdots \times I$ (n times) defined as follows: Let $F^0(\gamma) = \gamma$ for each γ in Γ^*. For $n > 0$, γ in Γ^*, and $u_1, \ldots, u_n$ in I, let $F^n(\gamma, u_1, \ldots, u_n) = \phi$ if $F^{n-1}(\gamma, u_1, \ldots, u_n) = \phi$ and let $F^n(\gamma, u_1, \ldots, u_n) = f(F^{n-1}(\gamma, u_1, \ldots, u_{n-1}), u_n)$ otherwise.

Thus $F^n(\gamma, u_1, \ldots, u_n)$ is the end result of starting with γ on the storage tape and applying $u_1, \ldots, u_n$ in sequence.

Notation. Let $(\Omega, \mathscr{D})$ be an AFA, with $\Omega = (K, \Sigma, \Gamma, I, f, g)$. Suppose that I and $g(\Gamma^*)$ are finite. Let h_1 and h_2 be one to one functions on I and $g(\Gamma^*)$ resp., into Σ such that $h_1(I) \cap h_2(g(\Gamma^*)) = \phi$. Then Σ_1 denotes $h_1(I)$, Σ_2 denotes $h_2(g(\Gamma^*))$, and $L_{\mathscr{D}}$ (actually, $L_{\mathscr{D}, h_1, h_2}$) denotes the set of all words of the form $h_2(\epsilon)$ or

$$h_2(\epsilon)\, h_1(u_1)\, h_2(\gamma_1) \cdots h_1(u_n)\, h_2(\gamma_n),$$

where $n \geq 1$, $u_1, \ldots, u_n$ are in I, γ_i is in $g[F^i(\epsilon, u_1, \ldots, u_i)]$ for all i, $i \leq i \leq n$, and $\gamma_n = \epsilon$.

Thus, in addition to $h_2(\epsilon)$, $L_{\mathscr{D}}$ contains the encoding of each sequence $\gamma_0, u_1, \gamma_1, \ldots, u_n, \gamma_n$, where $u_1, \ldots, u_n$ are instructions and $\gamma_0, \ldots, \gamma_n$ are the consecutive (under the instructions) storage information configurations, with $\gamma_0 = \gamma_n = \epsilon$. In other words, $L_{\mathscr{D}}$ represents the behavior of the storage information configurations (interspersed with appropriate instructions) during any possible accepting computation.

For each AFA $(\Omega, \mathscr{D})$ such that I and $g(\Gamma^*)$ are finite, there obviously exist an infinite number of pairs (h_1, h_2) satisfying the above conditions.

LEMMA 2.2. *If* $\mathscr{D}$ *is an* AFA, *with* I *and* $g(\Gamma^*)$ *finite, then* $L_{\mathscr{D}}$ *is in* $\mathscr{L}^t(\mathscr{D})$ *for each* $L_{\mathscr{D}}$.

Proof. Let $K_1 = \{q_0, q_1, q_2\}$ be a set of three distinct symbols of K. Let $D = (K_1, \Sigma_1 \cup \Sigma_2, \delta, q_0, F_1)$, where $F_1 = \{q_1\}$ and δ is defined as follows:

(1) $\delta(q_0, h_2(\epsilon), \epsilon) = \{(q_1, 1_\epsilon)\}$,
(2) $\delta(q_1, h_1(u), \gamma) = \{(q_2, u)\}$ for all γ in $g(\Gamma^*)$ and u in I, and
(3) $\delta(q_2, h_2(\gamma), \gamma) = \{(q_1, 1_\gamma)\}$ for all γ in $g(\Gamma^*)$.

Clearly D is in $\mathscr{D}_0{}^t$ and $L_{\mathscr{D}} = L(D)$ is in $\mathscr{L}^t(\mathscr{D})$.

Notation. Given $L_{\mathscr{D}}$, let T be a finite subset of Σ such that $T \cap (\Sigma_1 \cup \Sigma_2) = \phi$. Denote by h_T (actually, h_{T,h_1,h_2}) the homomorphism on $(T \cup \Sigma_1 \cup \Sigma_2)^*$ defined by $h_T(a) = \epsilon$ for each a in T and $h_T(a) = a$ for each a in $\Sigma_1 \cup \Sigma_2$. Denote by L_T (actually, $L_{T,\mathscr{D},h_1,h_2}$) the set $h_T^{-1}(L_{\mathscr{D}})$. Denote by $\bar{h}_T$ (actually, $\bar{h}_{T,h_1,h_2}$) the homomorphism on $(T \cup \Sigma_1 \cup \Sigma_2)^*$ defined by $\bar{h}_T(a) = a$ for each a in T and $\bar{h}_T(a) = \epsilon$ for each a in $\Sigma_1 \cup \Sigma_2$. For each acceptor $D = (K_1, T, \delta, q_0, F_1)$ in $\mathscr{D}$ denote by H_D (actually, $H_{D,\mathscr{D},h_1,h_2}$) the right-linear, context-free grammar[8] $(K_1 \cup \Sigma_1 \cup \Sigma_2 \cup T \cup \{\sigma\}, \Sigma_1 \cup \Sigma_2 \cup T, P_D, \sigma)$, where σ is a new symbol in Σ and

$$P_D = \{\sigma \to q_0\} \cup \{q \to a h_2(y)\, h_1(u)\, q' / (q', u) \text{ in } \delta(q, a, y)\} \\ \cup \{q \to h_2(\epsilon) / q \text{ in } F_1\}.$$

Denote by R_D (actually $R_{D,\mathscr{D},h_1,h_2}$) the set[9] $L(H_D)$.

Thus L_T consists of all words in $L_{\mathscr{D}}$ obtained by inserting arbitrary words of T^* into words of $L_{\mathscr{D}}$. The homomorphism $\bar{h}_T$ preserves elements of T and erases all other symbols. The set R_D consists of all words of the form

$$a_1 h_2(\gamma_0)\, h_1(u_1) \cdots a_n h_2(\gamma_{n-1})\, h_1(u_n)\, h_2(\gamma_n),$$

where $\gamma_n = \epsilon$ and D has a sequence of states $q_0, \ldots, q_n$, q_n in F_1, with (q_i, u_i) in $\delta(q_{i-1}, a_i, \gamma_{i-1})$ for each i, $1 \leqslant i \leqslant n$.

Since H_D is right linear, R_D is regular set [2, Theorem 6].

[8] A *context-free grammar* is a 4-tuple $G = (V, \Sigma_1, P, \sigma)$, where V is a finite set of symbols, $\Sigma_1 \subseteq V$, σ is in $V - \Sigma_1$, and P is a finite set of ordered pairs (ξ, y), with ξ in $V - \Sigma_1$ and y in V^*. Each pair (ξ, y) is written $\xi \to y$. A context-free grammar is *right linear* if each production in it is of the form $\xi \to z$ or $\xi \to z\mu$, where z is in $\Sigma_1{}^*$ and μ is in $V - \Sigma_1$.

[9] For each context-free grammar $G = (V, \Sigma_1, P, \sigma)$, write $u \Rightarrow v$ if there exist $u_1\ u_2, u_3$, and ξ such that $u = u_1 \xi u_2$, $v = u_1 u_3 u_2$, and $\xi \to u_3$ is in P. Let $\Rightarrow^*$ be the reflexive transitive closure of $\Rightarrow$. Let $L(G) = \{w \text{ in } \Sigma_1{}^* / \sigma \Rightarrow^* w\}$. $L(G)$ is called the *context-free language generated* by G or by P.

LEMMA 2.3. *$L(D) = \bar{h}_T(L_T \cap R_D)$ for each $D = (K_1, T, \delta, q_0, F_1)$ in $\mathscr{D}$, and if D is quasi-realtime, then $\bar{h}_T$ is ϵ-limited on $L_T \cap R_D$.*

Proof. Informally, $L_{\mathscr{D}}$ contains all sequences $\epsilon, u_1, \gamma_1, ..., u_n, \gamma_n = \epsilon$ in the encoded form $h_2(\epsilon)\, h_1(u_1)\, h_2(\gamma_1) \cdots h_1(u_n)\, h_2(\gamma_n)$, where

$$\epsilon, F^1(\epsilon, u_1), ..., F^{n-1}(\epsilon, u_1, ..., u_{n-1}), F^n(\epsilon, u_1, ..., u_n) = \epsilon$$

is a sequence of auxiliary storage configurations for some possible accepting computation of some acceptor of $\mathscr{D}$ and γ_i is in $g(F^i(\epsilon, u_1, ..., u_i))$ for each i. Also, L_T is $L_{\mathscr{D}}$ with all possible input strings of T^* inserted at all points. The set R_D consists of all words

$$a_1 h_2(\gamma_0)\, h_1(u_1) \cdots a_n h_2(\gamma_{n-1})\, h_1(u_n)\, h_2(\gamma_n),$$

where $\gamma_n = \epsilon$ and D has a sequence of states $q_0, ..., q_n$, q_n an accepting state, with (q_i, u_i) in $\delta(q_{i-1}, a_i, \gamma_{i-1})$ for each i. Thus $L_T \cap R_D$ combines both auxiliary storage and state transition restrictions. The homomorphism $\bar{h}_T$ then erases all symbols not the input of a computation, leaving $L(D)$.

Formally, we first prove that $L(D) \subseteq \bar{h}_T(L_T \cap R_D)$. To this end, let w be in $L(D)$. Suppose $w = \epsilon$ and q_0 is in F_1. Then $\sigma \Rightarrow q_0 \Rightarrow h_2(\epsilon)$, so that $h_2(\epsilon)$ is in R_D. Since $h_2(\epsilon)$ is in $L_{\mathscr{D}}$, it is in L_T, thus in $L_T \cap R_D$. Then ϵ is in $\bar{h}_T(h_2(\epsilon))$. Now suppose either $w = \epsilon$ and q_0 is not in F_1, or else $w \neq \epsilon$. Then there exist $n \geqslant 1$, $a_1, ..., a_n$ in $T \cup \{\epsilon\}$, $p_1, ..., p_n$ in K_1, with p_n in F_1, and $\gamma_1, ..., \gamma_{n-1}$ in Γ^* such that

$$\begin{aligned}(q_0, a_1 \cdots a_n, \epsilon) &\vdash (p_1, a_2 \cdots a_n, \gamma_1)\\ &\cdots\\ &\vdash (p_{n-1}, a_n, \gamma_{n-1})\\ &\vdash (p_n, \epsilon, \epsilon)\end{aligned}$$

and $w = a_1 \cdots a_n$. Hence there exist $u_1, ..., u_n$ in I and $y_1, ..., y_{n-1}$ in G_D such that $f(\epsilon, u_1) = \gamma_1$, $f(\gamma_i, u_{i+1}) = \gamma_{i+1}$ for $1 \leqslant i \leqslant n-2$, $f(\gamma_{n-1}, u_n) = \epsilon$, y_i in $g(\gamma_i)$ for $1 \leqslant i \leqslant n-1$, (p_1, u_1) in $\delta(q_0, a_1, \epsilon)$, and (p_{i+1}, u_{i+1}) in $\delta(p_i, a_{i+1}, y_i)$ for $1 \leqslant i \leqslant n-1$. Since $\gamma_i = F^i(\epsilon, u_1, ..., u_i)$ for $1 \leqslant i \leqslant n-1$ and $\epsilon = F^n(\epsilon, u_1, ..., u_n)$, it follows that y_i is in $g(F^i(\epsilon, u_1, ..., u_i))$ for $1 \leqslant i \leqslant n-1$ and $\{\epsilon\} = g(F^n(\epsilon, u_1, ..., u_n))$. Therefore

$$h_2(\epsilon)\, h_1(u_1)\, h_2(y_1) \cdots h_1(u_n)\, h_2(\epsilon)$$

is in $L_{\mathscr{D}}$. Hence

$$w' = a_1 h_2(\epsilon)\, h_1(u_1)\, a_2 h_2(y_1)\, h_1(u_2) \cdots a_n h_2(y_{n-1})\, h_1(u_n)\, h_2(\epsilon)$$

is in L_T. Obviously w' is in R_D, so that w' is in $L_{\mathscr{D}} \cap R_D$, and $w = \bar{h}_T(w')$. Thus $L(D) \subseteq \bar{h}_T(L_T \cap R_D)$.

To see the reverse inclusion, let w be in $\bar{h}_T(L_T \cap R_D)$. Suppose $w = \epsilon$ and $h_2(\epsilon)$

in is R_D. Then q_0 is in F_1, so that ϵ is in $L(D)$. Suppose either $w = \epsilon$ and $h_2(\epsilon)$ is not in R_D, or else $w \neq \epsilon$. Then there exist $n \geqslant 1$, $u_1, \ldots, u_n$ in I, $y_1, \ldots, y_{n-1}$ in $g(\Gamma^*)$, and $a_1, \ldots, a_n$ in $T \cup \{\epsilon\}$ such that

$$w' = a_1 h_2(\epsilon)\, h_1(u_1)\, a_2 h_2(y_1) \cdots h_1(u_n)\, h_2(\epsilon),$$

$w = a_1 \cdots a_n = \bar{h}_T(w')$, and w' is in $L_T \cap R_D$. Since w' is in L_T, there exist $\gamma_1, \ldots, \gamma_{n-1}$ such that $\gamma_1 = f(\epsilon, u_1)$, $\gamma_i = F^i(\epsilon, u_1, \ldots, u_i)$ for $1 \leqslant i \leqslant n-1$, $\epsilon = F^n(\epsilon, u_1, \ldots, u_n)$, and y_i is in $g(\gamma_i)$ for $1 \leqslant i \leqslant n-1$. Since w' is in R_D, there exist $p_1, \ldots, p_n$ such that $q_0 \to a_1 h_2(\epsilon)\, h_1(u_1)\, p_1$, $p_i \to a_{i+1} h_2(y_i)\, h_1(u_{i+1})\, p_{i+1}$ for $1 \leqslant i \leqslant n-1$, and $p_n \to h_2(\epsilon)$. It follows from the definition of R_D that p_n is in F_1, (p_1, u_1) is in $\delta(q_0, a_1, \epsilon)$, and (p_{i+1}, u_{i+1}) is in $\delta(p_i, a_{i+1}, y_i)$ for $1 \leqslant i \leqslant n-1$. Hence

$$\begin{aligned}(q_0, a_1 \cdots a_n, \epsilon) &\vdash (p_1, a_2 \cdots a_n, \gamma_1)\\ &\cdots\\ &\vdash (p_{n-1}, a_n, \gamma_{n-1})\\ &\vdash (p_n, \epsilon, \epsilon)\end{aligned}$$

so that $w = a_1 \cdots a_n$ is in $L(D)$. Thus $\bar{h}_T(L_T \cap R_D) \subseteq L(D)$, whence $L(D) = \bar{h}_T(L_T \cap R_D)$.

Suppose that D is quasi-realtime. Then there exists $k \geqslant 0$ such that D has at most k consecutive ϵ-moves. Thus, for $w = a_1 \cdots a_n$ in $L(D)$, at most k consecutive a_i can be ϵ. Hence $\bar{h}_T$ maps at most $2(k+1)$ consecutive symbols of w' into ϵ. Therefore $\bar{h}_T$ is ϵ-limited on $L_T \cap R_D$, completing the proof.

Lemma 2.3 states that a language recognized by a member D of an AFA having I and $g(\Gamma^*)$ finite can be expressed as the homomorphic image of the intersection of (a) a regular set which encodes the finite state control of D, and (b) a member of a fixed family of languages, each of which encodes the action of the AFA on the auxiliary storage. Thus Lemma 2.3 resembles the Chomsky-Schutzenberger characterization of the pushdown acceptor (= context-free) languages given in Example 1.1. In particular, it can be shown that each context-free language L is of the form $h_2(h_1^{-1}(K_2) \cap R)$, where h_1 and h_2 are homomorphisms, R is a regular set, and $K_2 \subseteq \{a_1, a_1^E, a_2, a_2^E\}^*$ is the context-free language generated by the rules $\xi \to a_1 \xi a_1^E$, $\xi \to a_2 \xi a_2^E$, $\xi \to \xi\xi$, and $\xi \to \epsilon$. In passing, we remark that for many well-known AFA the proof of Lemma 2.3 can be modified to exhibit an "intuitively obvious" language as a generator for the AFL associated with the AFA. Among other results, we can modify the argument in Lemma 2.3 to show that (1) the Dyck set on two letters is a (full) generator for the context-free languages, and (2) the Dyck set on one letter is a (full) generator for the 1-counter languages. [Also, see example 2.2.] We shall give the details elsewhere.

We are now ready to characterize (full) principal AFL in terms of AFA.

THEOREM 2.1. *Let* $(\Omega, \mathscr{D})$ *be an* AFA. *Then* $\mathscr{L}(\mathscr{D})$ $(\mathscr{L}^t(\mathscr{D}))$ *is full principal (principal) if and only if* $(\Omega, \mathscr{D})$ *is finitely (t-) encodable.*

Proof. By Lemma 2.1 it suffices to show the "if."

(a) Suppose $(\Omega, \mathscr{D})$ is finitely encodable. Then $(\Omega, \mathscr{D})$ has a sub-AFA $(\Omega', \mathscr{D}')$, where $\Omega' = (K', \Sigma, \Gamma', I', f', g')$, with I' and $g'(\Gamma'^*)$ finite, such that $\mathscr{L}(\mathscr{D}) = \mathscr{L}(\mathscr{D}')$. Consider $L_{\mathscr{D}'}$ for some appropriate functions h_1 and h_2. Thus, $L_{\mathscr{D}'} \subseteq (\Sigma_1 \cup \Sigma_2)^*$. By Lemma 2.2, $L_{\mathscr{D}'}$ is in $\mathscr{L}^t(\mathscr{D}') \subseteq \mathscr{L}(\mathscr{D}')$. Since $\mathscr{L}(\mathscr{D}')$ is a full AFL [4], $\hat{\mathscr{F}}(L_{\mathscr{D}'}) \subseteq \mathscr{L}(\mathscr{D}') = \mathscr{L}(\mathscr{D})$. We shall show the reverse inequality.

Let L be in $\mathscr{L}(\mathscr{D}')$, with $L \subseteq T^*$, $T \subseteq \Sigma$ finite. For each a in T, let $\bar{a}$ be a distinct element in $\Sigma - (\Sigma_1 \cup \Sigma_2 \cup T)$. Let $\bar{T} = \{\bar{a}/a \text{ in } T\}$ and let α_T be the homomorphism from T^* into $\bar{T}^*$ defined by $\alpha_T(a) = \bar{a}$ for each a in T. Then $(\Sigma_1 \cup \Sigma_2) \cap \bar{T} = \phi$ and $\bar{L} = \alpha_T(L)$ is in $\mathscr{L}(\mathscr{D}')$. Let $D = (K_1, \bar{T}, \delta, q_0, F_1)$ in $\mathscr{D}'$ be such that $\bar{L} = L(D)$. By Lemma 2.3, $\bar{L} = \bar{h}_T(h_T^{-1}(L_{\mathscr{D}'}) \cap R_D)$. Since R_D is regular, $\bar{L}$ is in $\hat{\mathscr{F}}(L_{\mathscr{D}'})$. Hence $L = \alpha_T^{-1}(\bar{L})$ is in $\hat{\mathscr{F}}(L_{\mathscr{D}'})$, so that $\hat{\mathscr{F}}(L_{\mathscr{D}'}) = \mathscr{L}(\mathscr{D})$.

(b) Suppose $(\Omega, \mathscr{D})$ is finitely t-encodable. Then $(\Omega, \mathscr{D})$ contains a sub-AFA $(\Omega', \mathscr{D}')$, where $\Omega' = (K', \Sigma, \Gamma', I', f', g')$, with I' and $g'(\Gamma'^*)$ finite, such that $\mathscr{L}^t(\mathscr{D}) = \mathscr{L}^t(\mathscr{D}')$. It readily follows that $\mathscr{F}(L_{\mathscr{D}'}) \subseteq \mathscr{L}^t(\mathscr{D}')$. To see the reverse inequality, we use the notation in (a). Let L be in $\mathscr{L}^t(\mathscr{D}')$, with $L \subseteq T^*$, $T \subseteq \Sigma$ finite. Since α_T is an ϵ-free homomorphism, $\bar{L} = \alpha_T(L)$ is in $\mathscr{L}^t(\mathscr{D}')$ and $\bar{h}_T$ is ϵ-limited on $h_T^{-1}(L_{\mathscr{D}'} \cap R_D)$. Thus $\bar{L}$, hence L, is in $\mathscr{F}(L_{\mathscr{D}'})$. Therefore $\mathscr{L}^t(\mathscr{D}') = \mathscr{F}(L_{\mathscr{D}'})$, completing the proof.

Since $\mathscr{L}$ is a full AFL ($\mathscr{L}$ is an AFL containing $\{\epsilon\}$) if and only if $\mathscr{L} = \mathscr{L}(\mathscr{D})$ ($\mathscr{L} = \mathscr{L}^t(\mathscr{D})$) for some AFA $\mathscr{D}$, we have

COROLLARY 1. *An* AFL $\mathscr{L}$ *(containing* $\{\epsilon\}$*) is full principal (principal) if and only if* $\mathscr{D}$ *is finitely (t-) encodable for every* AFA $\mathscr{D}$ *such that* $\mathscr{L} = \mathscr{L}(\mathscr{D})$ $(\mathscr{L} = \mathscr{L}^t(\mathscr{D}))$.

COROLLARY 2. *An* AFL $\mathscr{L}$ *(containing* $\{\epsilon\}$*) is full principal (principal) if and only if* $\mathscr{L} = \mathscr{L}(\mathscr{D})$ $(\mathscr{L} = \mathscr{L}^t(\mathscr{D}))$ *for some finitely (t-) encodable* AFA $\mathscr{D}$.

Now for each AFA $\mathscr{D}$ and each D in $\mathscr{D}$, there exists a homomorphism h and *a* D' in $\mathscr{D}_0{}^t$ such that $h(L(D')) = L(D)$ [4]. Thus $\mathscr{L}(\mathscr{D})$ is the closure of $\mathscr{L}^t(\mathscr{D})$ under arbitrary homomorphism. This yields

COROLLARY 3. *A full* AFL $\mathscr{L}$ *is principal if and only if there exists a finitely t-encodable* AFA $\mathscr{D}$ *such that* $\mathscr{L} = \mathscr{L}(\mathscr{D}) = \mathscr{L}^t(\mathscr{D})$.

We conclude the section with several applications of Theorem 2.1 to known AFL.

EXAMPLE 2.1. Let $\mathscr{D}_p$ be the AFA formalization for pda. Since the context-free languages form a full principal AFL, $\mathscr{D}_p$ is finitely encodable by Theorem 2.1. In particular, using the notation[10] of Example 1 of [5], let $\mathscr{D}'$ be the sub-AFA of $\mathscr{D}_p$,

[10] We frequently present instructions as words over some alphabet, instead of as abstract symbols. This is done to stay as close as possible to the customarily used symbolism of certain acceptors. If an instruction is given as a word w in V^* for some alphabet V, then we may formally conceive of the instruction as the abstract symbol i_w.

with $\Gamma' = \{A, B\}$, $H = \{A, B, \epsilon\}$, $I' = \Gamma^2 \cup \Gamma \cup \{\epsilon\}$, $g'(yA) = \{A\}$, $g'(yB) = \{B\}$, $g'(\epsilon) = \{\epsilon\}$, $f'(\epsilon, u) = u$, and $f'(yA, u) = yu = f'(yB, u)$ for all y in Γ^* and u in I. Now every context-free language is accepted by a pda which (α) has at most two storage symbols, and (β) never increases the length of the auxiliary storage by more than one. Thus $\mathscr{L}(\mathscr{D}') = \mathscr{L}(\mathscr{D}_p)$. Since I' and $g'(\Gamma'^*)$ are finite, $\mathscr{D}_p$ is finitely encodable via $\mathscr{D}'$.

It is known [8] that every context-free language is accepted by an ϵ-free pda, thus by a pda $\mathscr{D}_p{}^t$. Therefore $\mathscr{L}(\mathscr{D}_p) = \mathscr{L}^t(\mathscr{D}_p)$. Moreover, for every quasi-realtime pda D there clearly is a quasi-realtime pda D' in $\mathscr{D}'$ such that $L(D) = L(D')$. Thus $\mathscr{D}_p$ is finitely t-encodable. By Theorem 2.1, the context-free languages form a principal AFL. It can be shown, although not done here, that the Dyck set K_2 is a generator, as well as a full generator, for the context-free languages.

EXAMPLE 2.2. Let $\mathscr{D}$ be the family of one-way (nonerasing) stack acceptors as defined in Example 4 of [4]. Let $\mathscr{D}'$ be the sub-AFA of $\mathscr{D}$ in which $\Gamma' = \{A, B, \uparrow\}$, $H = \{A, B, \epsilon\}$, and $I' = (\Gamma'')^2 \cup \Gamma'' \cup \{\epsilon, +, -, 0\}$, with $\Gamma'' = \{A, B\}$. It is not difficult to see that for every (nonerasing) one-way stack acceptor D there is a one-way (nonerasing) stack acceptor D' satisfying (α) and (β) in Example 2.1 and such that $L(D') = L(D)$. Furthermore, D' is quasi-realtime if D is. Hence $\mathscr{L}(\mathscr{D}) = \mathscr{L}(\mathscr{D}')$ and $\mathscr{L}^t(\mathscr{D}) = \mathscr{L}^t(\mathscr{D}')$, so that $\mathscr{D}$ is finitely encodable and finitely t-encodable. Therefore the one-way (nonerasing) stack languages form a full principal AFL and the one-way (nonerasing) quasi-realtime stack languages form a principal AFL (which is not full by the corollary to Theorem 1.1 of [10]).

In connection with the remark made prior to Theorem 2.1, the proof of Lemma 2.3 can be modified to exhibit an "intuitively obvious" language as a full generator (generator) for the (quasi-realtime) one-way stack acceptor languages. This full generator (generator) is the set $L_0 \subseteq \{a, a^E, a^L, a^R, b, b^E, b^L, b^R\}^*$ which consists of all words $a_1 \cdots a_k$ where, interpreting x as "add x to the stack," x^E as "erase x from the stack," x^L as "move the pointer from the right of x to the left of x," and x^R as "move the pointer from the left of x to the right of x," x in $\{a, b\}$, $F^k(e, a_1, ..., a_k) = e$, that is, the word $a_1 \cdots a_k$ describes the action on the auxiliary storage as an acceptor makes a sequence of moves, starting and ending with empty storage. For example $aaa^Ebb^Lb^Rb^Ea^E$ is in L_0. We shall present the details elsewhere.

EXAMPLE 2.3. Let $\mathscr{D}$ be the AFA defined as follows. Let $\bar{\Gamma}$ be an infinite set such that $\bar{\Gamma} \cap \{\uparrow, ¢, \$, -1, 0, 1\} = \phi$, and let

$$\Gamma = \bar{\Gamma} \cup \{\uparrow, ¢, \$\} \qquad \text{and} \qquad I = \bar{\Gamma}^*\$ \cup \{\epsilon\} \cup \{-1, 0, 1\} \cup ¢\bar{\Gamma}^*\$.$$

Let g be defined by $g(\epsilon) = \{\epsilon\}$, $g(xZ\uparrow y) = \{Z\}$, and $g(xZ\$\uparrow y) = \{Z\$\}$ for all Z in $\bar{\Gamma} \cup \{¢\}$ and x, y in $(\Gamma - \{\uparrow\})^*$. Let f be defined as follows (for all xy in $(\Gamma - \{\uparrow\})^*$, $x \neq \epsilon$, and u in $\bar{\Gamma}^*$):

(a) $f(\epsilon, ¢u\$) = ¢u\$\urcorner$ and $f(xZ\urcorner y, ¢u\$) = x¢u\$\urcorner Zy$ for all Z in $\bar{\Gamma} \cup \bar{\Gamma}\$$.
(b) $f(xZ\$\urcorner y, u\$) = xu\$\urcorner y$ for all Z in $\bar{\Gamma}$.
(c) $f(x¢\$\urcorner Zy, \epsilon) = xZ\urcorner y$ and $f(¢\$\urcorner, \epsilon) = \epsilon$.
(d) $f(xZ\$\urcorner y, -1) = x\urcorner Z\$y$ and $f(xZ\urcorner y, -1) = x\urcorner Zy$ for Z in $\bar{\Gamma} \cup \{¢\}$.
(e) $f(xZ\urcorner y, 0) = xZ\urcorner y$ for Z in Γ.
(f) $f(y'Y\urcorner Zy, 1) = y'YZ\urcorner y$ and $f(y'Y\urcorner Z\$y, 1) = y'YZ\$\urcorner y$ for all Y, Z in $\bar{\Gamma} \cup \{¢\}$ and y' in $(\Gamma - \{\urcorner\})^*$.

The family $\mathscr{D}$ is the AFA of nested stack acceptors (nsa) defined in [1].[11]
Let $\mathscr{D}'$ be the sub-AFA where

$$\Gamma' = \{A, B, ¢, \$, \urcorner, -1, 0, 1\}, \qquad H = \{A, B, ¢, \epsilon, A\$, B\$, ¢\$\},$$

and[12]

$$I' = \{¢u\$/ |u| \leqslant 2, u \text{ in } \{A, B\}^*\} \cup \{\epsilon, -1, 0, 1\} \cup \{u\$/ |u| \leqslant 2, u \text{ in } \{A, B\}^*\}.$$

It can be shown that $\mathscr{L}(\mathscr{D}') = \mathscr{L}(\mathscr{D})$ and $\mathscr{L}^t(\mathscr{D}') = \mathscr{L}^t(\mathscr{D})$. Consequently the (quasi-realtime) nsa languages form a full principal (principal) AFL.

EXAMPLE 2.4. Since each Turing acceptor is simulated by one whose auxiliary storage contains just the blank symbol, the pointer symbol, and two other symbols, the r.e. sets form a full principal AFL. Consider the problem of whether the r.e. sets form a principal AFL. Let $\Sigma = \{a_1, ..., a_i, ...\}$ and let h be the homomorphism on Σ^* defined by $h(a_i) = a_1{}^i a_2$ for each i. Let $\Sigma_n = \{a_1, ..., a_n\}$ and let $L_1, ..., L_i, ...$ be an enumeration of the r.e. sets such that $L_n \subseteq \Sigma_n{}^*$ for each n. For each n, let $\gamma(n)$ be a Godel number assigned L_n. Then $L = \{a_1^{\gamma(n)} a_2 h(w)/w \text{ in } L_n, n \geqslant 1\} \cup \{\epsilon\}$ is an r.e. set in $\{a_1, a_2\}^*$. For each n let M_n be a gsm, ϵ-limited on[13] $\{a_1, a_2\}^*$, that maps $a_1^{\gamma(n)} a_2$ into ϵ and then decodes $a_1{}^i a_2$ as a_i for $1 \leqslant i \leqslant n$. Clearly $L_n = M_n(L \cap a_1^{\gamma(n)} a_2 \Sigma_2{}^*)$. Then $L \cap a_1^{\gamma(n)} a_2 \Sigma_2{}^*$ is in $\mathscr{F}(L)$. Since an ϵ-limited gsm maps each language in an AFL containing $\{\epsilon\}$ into a language in the AFL by Corollary 4 of Theorem 2.1 of [4], L_n is in $\mathscr{F}(L)$. Therefore $\mathscr{F}(L)$ is the family of r.e. sets, i.e., the r.e. sets form a principal AFL.

SECTION 3. REPRESENTATION THEOREMS

By definition, $\mathscr{F}(L)$ and $\hat{\mathscr{F}}(L)$ are the smallest AFL and full AFL, resp., containing L. As such, each set L' in $\mathscr{F}(L)$ ($\hat{\mathscr{F}}(L)$) is obtained from L by a finite number of applications of the closure operations of a (full) AFL. For many purposes, this method

[11] This definition differs trivially from the one in [1] mainly because AFA acceptors start with empty storage whereas the acceptors in [1] start with $\#Z_0\$\urcorner$ for some fixed Z_0.
[12] For each word x, $|x|$ denotes its length.
[13] See [4].

of representing an element in $\mathscr{F}(L)$ ($\hat{\mathscr{F}}(L)$) is extremely awkward to apply. In the present section, we shall give two representations for each such language. Specifically, we shall show that each set L' in $\mathscr{F}(L)$ ($\hat{\mathscr{F}}(L)$) can be represented in the form (i) $M((Lc)^*)$ for some appropriate "transducer" M, and in the form (ii) $h_2(h_1^{-1}((Lc)^*) \cap R)$, where R is a regular set, c is a symbol not in Σ_L, and h_1 and h_2 are appropriate homomorphisms.

In order to present the first representation theorem, we recall some notions about a-transducers.

DEFINITION. An *a-transducer* is a 6-tuple $M = (K, \Sigma_1, \Sigma_2, H, p_0, F)$, where

(1) K, Σ_1, and Σ_2 are finite sets (of *states*, *inputs*, and *outputs*, resp.).
(2) H is a finite subset of $K \times \Sigma_1{}^* \times \Sigma_2{}^* \times K$ (the *moves*).
(3) p_0 is in K (the *start* state).
(4) $F \subseteq K$ (set of *accepting* states).

If $H \subseteq K \times \Sigma_1{}^* \times \Sigma_2{}^+ \times K$, the M is called *ϵ-free*.

The moves of an a-transducer are described by the following symbolism:

Notation. Let $\vdash$ and $\overset{*}{\vdash}$ be the relations on $K \times \Sigma_1{}^* \times \Sigma_2{}^*$ defined as follows: Let $(p, xw, z_1) \vdash (q, w, z_2)$ if (p, x, y, q) is in H and $z_2 = z_1 y$. Let $\overset{*}{\vdash}$ be the reflexive transitive closure of $\vdash$.

In particular, $(p, w, z) \overset{*}{\vdash} (p, w, z)$ for all (p, w, z) in $K \times \Sigma_1{}^* \times \Sigma_2{}^*$.

The triple (p, w, z) represents the fact that M is in state p, with w the input still to be read, and z the accumulated output. $(p, w, z) \overset{*}{\vdash} (p', w', z')$ means that M can go from (p, w, z) to (p', w', z') by a sequence of zero or more elementary moves.

The a-transducer effects an operation as follows:

DEFINITION. Let $M = (K, \Sigma_1, \Sigma_2, H, p_0, F)$ be an a-transducer. For each word w in $\Sigma_1{}^*$, let $M(w) = \{z/(p_0, w, \epsilon) \overset{*}{\vdash} (p, \epsilon, z)$ for some p in $F\}$. For every $W \subseteq \Sigma_1{}^*$, let $M(W) = \bigcup_{w \text{ in } W} M(w)$. The mapping M from $2^{\Sigma_1{}^*}$ into $2^{\Sigma_2{}^*}$ so defined is called an *a-transducer mapping*.

Notation. For each family $\mathscr{L}$ of languages let $\hat{\mathscr{M}}(\mathscr{L})$ $[\mathscr{M}(\mathscr{L})]$ be the family of all sets $M(L)$, where M is an [ϵ-free] a-transducer and L is in $\mathscr{L}$. If $\mathscr{L} = \{L\}$, then we write $\mathscr{M}(L)$ $[\hat{\mathscr{M}}(L)]$ instead of $\mathscr{M}(\mathscr{L})$ $[\hat{\mathscr{M}}(\mathscr{L})]$.

Note that $\mathscr{M}(\mathscr{L})$ is undefined if $\mathscr{L}$ contains just the language ϕ.

We shall need the following result which, in essence, has been proved elsewhere.

PROPOSITION 3.1. *For each family $\mathscr{L}$ of languages,*

$$\hat{\mathscr{M}}(\mathscr{L})\ [\mathscr{M}(\mathscr{L})] = \{h_2(h_1^{-1}(L) \cap R)/R \textit{ regular}, L \textit{ in } \mathscr{L}, h_1 \textit{ and } h_2 \textit{ homomorphisms (with } h_2\ \epsilon\textit{-free)}\},$$

and is the smallest family containing $\mathscr{L}$ and closed under (ϵ-free) homomorphism, inverse homomorphism, and intersection with regular sets.

Proposition 3.1 follows from Remark 2 on page 8 of [4] (union with $\{\epsilon\}$ is not needed here) and from the fact that the composition of (ϵ-free) a-transducers is an (ϵ-free) a-transducer.

We now turn to the first representation theorem of a principal AFL. We need three lemmas.

LEMMA 3.1. *For each nonempty language L, $\mathscr{M}(L)$ ($\hat{\mathscr{M}}(L)$) is closed under union.*

Proof. It suffices to show the $\mathscr{M}(\mathscr{L})$ case only. Let $M_1 = (K_1, \Sigma_1, \Sigma_2, H_1, p_0, F_1)$ and $M_2 = (K_2, \Sigma_1, \Sigma_2, H_2, q_0, F_2)$ be ϵ-free a-transducers. Without loss, we may assume $K_1 \cap K_2 = \phi$. Let r_0 be a new symbol. Let $M_3 = (K_3, \Sigma_1, \Sigma_2, H_3, r_0, F_3)$, where $K_3 = K_1 \cup K_2 \cup \{r_0\}$, $F_3 = F_1 \cup F_2 \cup \{r_0/p_0 \text{ in } F_1 \text{ or } q_0 \text{ in } F_2\}$, and $H_3 = H_1 \cup H_2 \cup \{(r_0, u, v, p)/(p_0, u, v, p) \text{ in } H_1\} \cup \{(r_0, u, v, q)/(q_0, u, v, q) \text{ in } H_2\}$. Then M_3 is an ϵ-free a-transducer and $M_3(L) = M_1(L) \cup M_2(L)$. Thus $\mathscr{M}(L)$ is closed under union.

We need to consider certain kinds of a-transducers.

DEFINITION. An a-transducer $M = (K_1, \Sigma_1, \Sigma_2, H, p_0, F)$ is said to be *ϵ-output bounded* if there exists $k \geqslant 0$ such that for each sequence

$$(p_1, w_1, \epsilon) \vdash \cdots \vdash (p_{r+1}, w_{r+1}, \epsilon),$$

$r \leqslant k$. M is said to be *1-bounded* if $H \subseteq K_1 \times (\Sigma_1 \cup \{\epsilon\}) \times (\Sigma_2 \cup \{\epsilon\}) \times K_1$.

Thus M is ϵ-output bounded if there exists $k \geqslant 0$ so that M never has $k+1$ consecutive ϵ-output moves. M is 1-bounded if it only reads and outputs words of length at most one.

LEMMA 3.2. *For each (ϵ-free) a-transducer M, there exists a 1-bounded a-transducer (which is ϵ-output bounded) $M_2 = (K_2, \Sigma_1, \Sigma_2, H_2, q_0, F_2)$ such that $M_2(L) = M_1(L)$ for every language L. Furthermore, there is no sequence of elements $(q_0, u_1, v_1, q_1), \ldots, (q_t, u_{t+1}, v_{t+1}, (*)\ q_{t+1})$ in H_2 such that $v_i = \epsilon$ for each i and q_{t+1} is in F_2.*

Conversely, for every ϵ-output bounded a-transducer M_2 satisfying (*), there exists an ϵ-free a-transducer M_1 such that $M_1(L) = M_2(L)$ for every language L.

The proof of the first part of Lemma 3.2 follows by adding additional states, in the obvious manner, so that in a move only elements of $\Sigma_1 \cup \{\epsilon\}$ are read and elements of $\Sigma_2 \cup \{\epsilon\}$ are output. The proof of the last part follows by generalizing, in a straightforward manner, the argument in Corollary 4 on page 7 of [4]. We omit the details.

LEMMA 3.3. *For each language L and each symbol c not in* Σ_L, $\mathscr{M}((Lc)^*)$ *and* $\hat{\mathscr{M}}((Lc)^*)$ *are closed under* $+$; *and if* $L \neq \phi$, *then* $\mathscr{M}((Lc)^+)$ *is closed under* $+$.

Proof. It suffices to give the proof for $\mathscr{M}((Lc)^*)$.

Let $M_1 = (K_1, \Sigma_1, \Sigma_2, H_1, p_0, F_1)$ be an ϵ-free a-transducer. Let $M_2 = (K_2, \Sigma_1, \Sigma_2, H_2, q_0, F_2)$ be a 1-bounded, ϵ-output bounded a-transducer satisfying (*) in Lemma 3.2 and such that $M_2(U) = M_1(U)$ for every language U. We shall construct a 1-bounded, ϵ-output bounded, a-tranducer M_3 satisfying (*) of Lemma 3.2 such that $M_3((Lc)^*) = [M_1((Lc)^*)]^+$. Hence there will exist an ϵ-free a-tranducer M_4 such that $M_4((Lc)^*) = M_3((Lc)^*)$, so that $M_4((Lc)^*) = M_1((Lc)^*)$ will be in $\mathscr{M}((Lc)^*)$.

To do this let $\bar{q}$ be a new symbol for each q in K_2. Let

$$M_3 = (K_3, \Sigma_1, \Sigma_2, H_3, q_0, F_2),$$

where $K_3 = K_2 \cup \{\bar{q}/q \text{ in } K_2\}$ and H_3 contains all 4-tuples of the following form:

(1) (q_1, u, v, q_2) if (q_1, u, v, q_2) is in H_2.
(2) (q_1, c, v, q_0) if (q_1, c, v, q_2) is in H_2 for some q_2 in F_2.
(3) $(q_1, c, v, \bar{q}_2)$ if (q_1, c, v, q_2) is in H_2 and q_2 is not in F_2.
(4) $(\bar{q}_1, \epsilon, v, \bar{q}_2)$ if (q_1, ϵ, v, q_2) is in H_2.
(r) $(\bar{q}_1, \epsilon, v, q_0)$ if (q_1, ϵ, v, q_2) is in H_2 and q_2 is in F_2.

Informally, the a-transducer M_3 operates as follows. On reading the last c in a word of the form $w_1cw_2c \cdots w_rc$, M_3 either simulates M_2 (by (1)), or resets at q_0 if M_2 goes to an accepting state (by (2)), or simulates M_2 in marked states so that if ϵ-moves take M_2 to an accepting state, then M_3 resets to q_0 (by (4) and (5)). Clearly M_3 is 1-bounded ϵ-output bounded, $M_3((Lc)^*) = M_2((Lc)^*)^+$, and M_3 satisfies (*) of Lemma 3.2.

We are now ready for the first representation result.

THEOREM 3.1. *Let L be a language and c a symbol not in* Σ_L. *Then*

$$\mathscr{F}(L \cup \{\epsilon\}) = \mathscr{M}((Lc)^*) \quad \text{and} \quad \hat{\mathscr{F}}(L) = \hat{\mathscr{M}}((Lc)^*).$$

If $L \neq \phi$, *then* $\mathscr{F}(L - \{\epsilon\}) = \mathscr{M}((Lc)^+)$.

Proof. It suffices to consider the argument for $\mathscr{F}(L \cup \{\epsilon\})$. Since $(Lc)^*$ is in $\mathscr{F}(L \cup \{\epsilon\})$, $\mathscr{M}((Lc)^*) \subseteq \mathscr{F}(L \cup \{\epsilon\})$. Consider the reverse containment. Obviously there exists an ϵ-free a-transducer M_1 such that $M_1((Lc)^*) = L \cup \{\epsilon\}$. Hence $\mathscr{M}(L \cup \{\epsilon\}) \subseteq \mathscr{M}((Lc)^*)$. Now $\mathscr{M}((Lc)^*)$ is closed under union by Lemma 3.1 and under $+$ by Lemma 3.3. By Proposition 3.1, $\mathscr{M}((Lc)^*)$ is closed under ϵ-free homomorphism, inverse homomorphism, and intersection with regular sets. By Lemma 1 of [12], $\mathscr{M}((Lc)^*)$ is an AFL, so that $\mathscr{F}(L \cup \{\epsilon\}) \subseteq \mathscr{M}((Lc)^*)$. Thus $\mathscr{F}(L \cup \{\epsilon\}) = \mathscr{M}((Lc)^*)$.

We now give a second representation for $\mathscr{F}(L)$ ($\hat{\mathscr{F}}(L)$), this in terms of homomorphism, inverse homomorphism, and intersection with regular sets.

Definition. A homomorphism h on Σ_1^* is *decreasing* if $|h(a)| \leqslant 1$ for each a in Σ_1.

Notation. Let $\mathscr{R}(\mathscr{R}_0)$ denote the family of (ϵ-free) regular sets.

Lemma 3.4. *For each nonempty language L,*

(a) $\hat{\mathscr{M}}(L \cup \{\epsilon\}) = \hat{\mathscr{M}}(L) = \{h_2(h_1^{-1}(L) \cap R)/R$ *in* $\mathscr{R}$, h_1 *and* h_2 *decreasing homomorphisms*$\}$.

(b) $\mathscr{M}(L \cup \{\epsilon\}) = \{h_2(h_1^{-1}(L) \cap R)/R$ *in* $\mathscr{R}$, h_1 *and* h_2 *decreasing homomorphisms,* h_2 ϵ*-limited on* $R\}$.

(c) *if* $L - \{\epsilon\} \neq \phi$, *then*

$$\mathscr{M}(L - \{\epsilon\}) = \{h_2(h_1^{-1}(L) \cap R)/R \text{ in } \mathscr{R},\ h_1 \text{ and } h_2 \text{ decreasing homomorphisms},$$
$$h_2\ \epsilon\text{-limited on } R,\ h_2^{-1}(\epsilon) \cap R = \phi\}.$$

Proof. We shall only prove (b) and (c), the proof of (a) being similar to that of (b).

Consider (b). Let $\mathscr{C}(L)$ be the family of all sets of the form $h_2(h_1^{-1}(L) \cap R)$, where R is in $\mathscr{R}$, h_1 is a decreasing homomorphism, and h_2 a decreasing homomorphism which is ϵ-limited on R. By Proposition 3.1, $\mathscr{M}(L \cup \{\epsilon\})$ is closed under ϵ-free homomorphism, inverse homomorphism, and intersection with regular sets. By Corollary 5, page 7 of [4], and Remark 2, page 8 of [4], $\mathscr{M}(L \cup \{\epsilon\})$ is closed under ϵ-limited homomorphism. Hence $\mathscr{M}(L \cup \{\epsilon\})$ contains $h_2(h_1^{-1}(L) \cap R)$ for each R in $\mathscr{R}$, each decreasing homomorphism h_1, and each decreasing homomorphism h_2 ϵ-limited on R, thus on $h_1^{-1}(L) \cap R$.

Now let M_1 be an ϵ-free a-transducer. We consider $M_1(L \cup \{\epsilon\}) = M_1(L) \cup M_1(\epsilon)$. Since $M_1(\epsilon)$ is regular, $M_1(\epsilon) - \{\epsilon\}$ is in $\mathscr{R}_0$. Since $L \neq \phi$, $\mathscr{M}(L)$ contains $\mathscr{R}_0$ by Theorem 1.1 of [4] and Remark 2, page 8 of [4]. Then $\mathscr{M}(L)$ contains $M_1(L) \cup (M_1(\epsilon) - \{\epsilon\})$ by Lemma 3.1. Hence $M_1(L) \cup (M_1(\epsilon) - \{\epsilon\}) = M_2(L)$ for some ϵ-free a-transducer M_2. Let $M = (K_1, \Sigma_1, \Sigma_2, H, p_0, F_1)$ be a 1-bounded, ϵ-output bounded a-transducer such that $M(L) = M_2(L)$. For each (p, u, v, p') in H, let $\overline{(p, u, v, p')}$ be a new symbol and Σ_3 the set of all such symbols. Let h_1 and h_2 be the homomorphisms from Σ_3^* into Σ_1^* and Σ_3^* into Σ_2^*, resp., defined by $h_1(\overline{(p, u, v, p')}) = u$ and $h_2(\overline{(p, u, v, p')}) = v$ for all $\overline{(p, u, v, p')}$ in Σ_3. Since M is 1-bounded, h_1 and h_2 are decreasing homomorphisms. Let $i_0 = 0$ and R_1 be the set

$$\{\overline{(p_{i_0}\ u_{i_1}\ v_{i_1}\ p_{i_1})} \cdots \overline{(p_{i_{n-1}}, u_{i_n}, v_{i_n}, p_{i_n})}/\overline{(p_{i_j}, u_{i_{j+1}}, v_{i_{j+1}}, p_{i_{j+1}})} \text{ in } \Sigma_3,$$
$$n \geqslant 1,\ 1 \leqslant j < n,\ p_{i_n} \text{ in } F_1\}.$$

Let $R_2 = R_1 \cup \{\epsilon \mid \epsilon$ in L, p_0 in $F_1\}$. Then R_1 and R_2 are regular sets and $M(L) = h_2(h_1^{-1}(L) \cap R_2)$. Since M is ϵ-output bounded, h_2 is ϵ-limited on R_2. Suppose

$M_1(L \cup \{\epsilon\})$ is ϵ-free. Then $M_1(L \cup \{\epsilon\}) = M_1(L) \cup (M_1(\epsilon) - \{\epsilon\}) = h_2(h_1^{-1}(L) \cap R_2)$, and $M_1(L \cup \{\epsilon\})$ is in $\mathscr{C}(L)$. Suppose $M_1(L \cup \{\epsilon\})$ contains ϵ. Then

$$M_1(L \cup \{\epsilon\}) = M_1(L) \cup (M_1(\epsilon) - \{\epsilon\}) \cup \{\epsilon\} = h_2(h_1^{-1}(L) \cap R_2) \cup \{\epsilon\}.$$

Two cases arise. Suppose L contains ϵ. Then $M_1(L \cup \{\epsilon\}) = h_2(h_1^{-1}(L) \cap (R_2 \cup \{\epsilon\}))$, so that $M_1(L \cup \{\epsilon\})$ is in $\mathscr{C}(L)$. Suppose L is ϵ-free. Since $L \neq \phi$, there exists $a_1 \cdots a_k$ in L, with $k \geqslant 1$ and each a_i in Σ_1. Let $\bar{a}_1, ..., \bar{a}_k$ be new symbols and $\Sigma_4 = \Sigma_3 \cup \{\bar{a}_i \mid 1 \leqslant i \leqslant k\}$. Let $R_3 = R_2 \cup \{\bar{a}_1, ..., \bar{a}_k\}$. Let h_3 and h_4 be the homomorphism on $\Sigma_4{}^*$ defined by $h_3(x) = h_1(x)$, $h_4(x) = h_2(x)$, $h_3(\bar{a}_i) = a_i$, and $h_4(\bar{a}_i) = \epsilon$ for all x in Σ_3 and all $\bar{a}_i$. Then $M_1(L \cup \{\epsilon\}) = h_4[h_3^{-1}(L) \cap R_3]$, with h_3 and h_4 decreasing homomorphisms, and h_4 ϵ-limited on R_3. Again $M_1(L \cup \{\epsilon\})$ is in $\mathscr{C}(L)$. In all cases, therefore, $M_1(L \cup \{\epsilon\})$ is in $\mathscr{C}(L)$.

Consider (c). Let $L - \{\epsilon\} \neq \phi$ and let $\mathscr{C}^-(L)$ be the family of all sets of the form $h_2[h_1^{-1}(L) \cap R]$, where R is a regular set, and h_1 and h_2 are decreasing homomorphisms, with h_2 ϵ-limited on R and $h_2^{-1}(\epsilon) \cap R = \phi$. By (b), $\mathscr{C}^-(L) \subseteq \mathscr{M}(L \cup \{\epsilon\})$. By definition, each language in $\mathscr{C}^-(L)$ is ϵ-free. Hence $\mathscr{C}^-(L) \subseteq \{L_1/L_1 \text{ in } \mathscr{M}(L \cup \{\epsilon\}), L_1\ \epsilon\text{-free}\}$. Consider the reverse containment. Suppose L_1 is ϵ-free and in $\mathscr{M}(L \cup \{\epsilon\})$. By (b), $L_1 = h_2[h_1^{-1}(L) \cap R]$, with R in $\mathscr{R}$, and h_1 and h_2 decreasing homomorphisms, with h_2 ϵ-limited on R. Since L_1 is ϵ-free,

$$h_2^{-1}(\epsilon) \cap [h^{-1}(L) \cap R] = \phi \qquad \text{and} \qquad L_1 = h_2[h^{-1}(L) \cap (R - (R \cap h_2^{-1}(\epsilon)))].$$

Since R is regular, $R - (R \cap h_2^{-1}(\epsilon))$ is regular. Since h_2 is ϵ-limited on R, it is ϵ-limited on $R - (R \cap h_2^{-1}(\epsilon))$. Obviously $h_2^{-1}(\epsilon) \cap [R - (R \cap h_2^{-1}(\epsilon))] = \phi$. Hence L_1 is in $\mathscr{C}^-(L)$, so that $\mathscr{C}^-(L) = \{L_1/L_1 \text{ in } \mathscr{M}(L \cup \{\epsilon\}), L_1\ \epsilon\text{-free}\}$.

Using Theorem 3.1 and Lemma 3.4, we now prove

THEOREM 3.2. *Let L be a language and c a symbol not in Σ_L. Then*

(a) $\hat{\mathscr{F}}(L) = \{h_2(h_1^{-1}((Lc)^*) \cap R)/R$ *in* $\mathscr{R}$, h_1 *and* h_2 *decreasing homomorphisms*$\}$.

(b) $\mathscr{F}(L \cup \{\epsilon\}) = \{h_2(h_1^{-1}((Lc)^*) \cap R)/R$ *in* $\mathscr{R}$, h_1 *and* h_2 *decreasing homomorphisms,* h_2 ϵ*-limited on* $R\}$.

(c) $\mathscr{F}(L - \{\epsilon\}) = \{h_2(h_1^{-1}((Lc)^*) \cap R)/R$ *in* $\mathscr{R}$, h_1 *and* h_2 *decreasing homomorphisms,* h_2 ϵ*-limited on* R, $h_2^{-1}(\epsilon) \cap R = \phi\}$,

and, if $L \neq \phi$,

$$\mathscr{F}(L - \{\epsilon\}) = \{h_2(h_1^{-1}((Lc)^+) \cap R)/R \text{ in } \mathscr{R}, h_1 \text{ and } h_2 \text{ decreasing homomorphisms, } h_2\ \epsilon\text{-limited on } R, h_2^{-1}(\epsilon) \cap R = \phi\}.$$

Proof. (a) and (b) follow immediately from Theorem 3.1 and Lemma 3.4.

Consider (c). If $L = \{\epsilon\}$, then (c) holds since $\mathscr{F}(L - \{\epsilon\})$ and the other two sets in the equations are all $\mathscr{R}_0$. If $L = \phi$, then the two sets in the first equation are also $\mathscr{R}_0$. Thus suppose $L - \{\epsilon\} \neq \phi$. The second equation then follows from Theorem 3.1 and Lemma 3.4. Consider the first equation. Let $\mathscr{E}(L)$ be the right side of the first equation. From (b), $\mathscr{E}(L) \subseteq \mathscr{F}(L \cup \{\epsilon\})$. Since

$$\mathscr{F}(L - \{\epsilon\}) = \{L'/L' \text{ in } \mathscr{F}(L \cup \{\epsilon\}), L' \text{ is } \epsilon\text{-free}\}$$

and each language in $\mathscr{E}(L)$ is ϵ-free, $\mathscr{E}(L) \subseteq \mathscr{F}(L - \{\epsilon\})$. To see the reverse containment, let L' be in $\mathscr{F}(L - \{\epsilon\})$. Since $\mathscr{F}(L - \{\epsilon\}) \subseteq \mathscr{F}(L \cup \{\epsilon\})$, $L' = h_2[h_1^{-1}((Lc)^*) \cap R]$ for some regular set R and some decreasing homomorphisms h_1 and h_2, where h_2 is ϵ-limited on R. Let $R' = R - (h_2^{-1}(\epsilon) \cap R)$. Clearly R' is regular, h_2 is ϵ-limited on R', and $h_2^{-1}(\epsilon) \cap R' = \phi$. Then $L' = h_2[h_1^{-1}((Lc)^*) \cap R] = h_2[h_1^{-1}((Lc)^*) \cap R'] \cup L''$, where $L'' = h_2[h_1^{-1}((Lc)^*) \cap (h_2^{-1}(\epsilon) \cap R)]$. Since L'' is either $\{\epsilon\}$ or ϕ and L' is ϵ-free, $L'' = \phi$. Hence $L' = h_2[h_1^{-1}((Lc)^*) \cap R']$, so that L' is in $\mathscr{E}(L)$, i.e., $\mathscr{F}(L - \{\epsilon\}) \subseteq \mathscr{E}(L)$, whence equality.

Section 4. Operators

We now discuss several operators on principal AFL that yield principal AFL. Now from Theorem 2.1 of Section 2 and from Theorem 2.1 and Lemma 1.1 of [11], it follows that (a) if $\mathscr{L}_1$ and $\mathscr{L}_2$ are (full) principal AFL, then so is the smallest (full) AFL containing the family $\{L_1 \cap L_2/L_1 \text{ in } \mathscr{L}_1, L_2 \text{ in } \mathscr{L}_2\}$. Similarly, from Theorem 2.1 of Section 2 and from Theorem 4.2 and Lemma 4.1 of [11], it follows that (b) if $\mathscr{L}_1$ and $\mathscr{L}_2$ are (full) principal AFL, then so is the family of sets obtained by substituting ϵ-free (arbitrary) languages of $\mathscr{L}_2$ into languages of $\mathscr{L}_1$. In this section we offer algebraic proofs of (a) and (b) instead of AFA dependent proofs. Furthermore, for each of the (full) AFL given by the conclusions of (a) and (b), we exhibit a (full) generator which depends on the given (full) generators for $\mathscr{L}_1$ and $\mathscr{L}_2$ in a reasonably simple manner.

Notation. For all families of languages $\mathscr{L}_1, \ldots, \mathscr{L}_n$, let

$$\mathscr{L}_1 \wedge \cdots \wedge \mathscr{L}_n = \{L_1 \cap \cdots \cap L_n/\text{each } L_i \text{ in } \mathscr{L}_i\}.$$

Notation. For all families $\mathscr{L}$ of languages, let

$$H(\mathscr{L}) = \{h(L)/L \text{ in } \mathscr{L}, h \text{ an } \epsilon\text{-free homomorphism on } \Sigma_L^*\},$$

and

$$\hat{H}(\mathscr{L}) = \{h(L)/L \text{ in } \mathscr{L}, h \text{ an arbitrary homomorphism on } \Sigma_L^*\}.$$

If $\mathscr{L}_1$ and $\mathscr{L}_2$ are AFL, then by Theorem 2.1 of [5] so is $H(\mathscr{L}_1 \wedge \mathscr{L}_2)$. Our first major results in this section are algebraic proofs that $H(\mathscr{L}_1 \wedge \mathscr{L}_2)$ is principal if $\mathscr{L}_1$ and $\mathscr{L}_2$ are, and $\hat{H}(\mathscr{L}_1 \wedge \mathscr{L}_2)$ is full principal if $\mathscr{L}_1$ and $\mathscr{L}_2$ are. In addition, we shall display (full) generators for $H(\mathscr{L}_1 \wedge \mathscr{L}_2)$ $(\hat{H}(\mathscr{L}_1 \wedge \mathscr{L}_2))$ from (full) generators for $\mathscr{L}_1$ and $\mathscr{L}_2$.

Notation. For all languages L_1 and L_2 let

$$\mathrm{Shuff}(L_1, L_2) = \{w_1 y_1 \cdots w_n y_n / w_1 \cdots w_n \text{ in } L_1, y_1 \cdots y_n \text{ in } L_2\}.$$

LEMMA 4.1. *Let L_1 and L_2 be languages such that $\Sigma_{L_1} \cap \Sigma_{L_2} = \phi$. Then for* $L = \mathrm{Shuff}(L_1, L_2)$,

(a) $\hat{\mathscr{M}}(L_1) \wedge \hat{\mathscr{M}}(L_2) \subseteq \hat{\mathscr{F}}(L)$ *if* $L_1 \neq \phi$ *and* $L_2 \neq \phi$,

(b) $\mathscr{M}(L_1 \cup \{\epsilon\}) \wedge \mathscr{M}(L_2 \cup \{\epsilon\}) \subseteq \mathscr{F}(L \cup \{\epsilon\})$, *and*

(c) $\mathscr{M}(L_1 - \{\epsilon\}) \wedge \mathscr{M}(L_2 - \{\epsilon\}) \subseteq \mathscr{F}(L - \{\epsilon\})$ *if* $L_1 - \{\epsilon\} \neq \phi$ *and* $L_2 - \{\epsilon\} \neq \phi$.

Proof. We shall show (c), the proofs for (a) and (b) being similar.

Suppose $L_1 - \{\epsilon\} \neq \phi$ and $L_2 - \{\epsilon\} \neq \phi$. Then L_1 and L_2 both contain a non-ϵ word. Let $L' = \mathrm{Shuff}(L_1 - \{\epsilon\}, L_2 - \{\epsilon\})$. As is easily seen,

(1) $\mathscr{F}(L') = \mathscr{F}(L - \{\epsilon\})$.

Now note that

(2) $h_1^{-1}(L_1 - \{\epsilon\}) \cap h_2^{-1}(L_2 - \{\epsilon\})$ is in $\mathscr{F}(L')$ for all homomorphisms h_1 and h_2.
For let h_1 and h_2 be homomorphisms of $\Sigma_1{}^*$ into $\Sigma_{L_1}^*$ and $\Sigma_2{}^*$ into $\Sigma_{L_2}^*$, resp. Let h_3 be the homomorphism on $(\Sigma_1 \cap \Sigma_2)^*$ defined by $h_3(a) = h_1(a)\, h_2(a)$ for each a in $\Sigma_1 \cap \Sigma_2$. Consider $h_1^{-1}(L_1 - \{\epsilon\}) \cap h_2^{-1}(L_2 - \{\epsilon\})$ and $h_3^{-1}(L')$. Clearly ϵ is not in

$$h_1^{-1}(L_1 - \{\epsilon\}) \cup h_2^{-1}(L_2 - \{\epsilon\}) \cup h_3^{-1}(L').$$

Let $a_1 \cdots a_k$ be arbitrary, with $k \geqslant 1$ and each a_i in $\Sigma_1 \cap \Sigma_2$. Then $a_1 \cdots a_k$ is in $h_1^{-1}(L_1 - \{\epsilon\}) \cap h_2^{-1}(L_2 - \{\epsilon\})$ if and only if $h_1(a_1) \cdots h_1(a_k)$ is in $L_1 - \{\epsilon\}$ and $h_2(a_1 \cdots a_k)$ is in $L_2 - \{\epsilon\}$. Since $\Sigma_{L_1} \cap \Sigma_{L_2} = \phi$, this occurs if and only if

$$h_1(a_1)\, h_2(a_1) \cdots h_1(a_k)\, h_2(a_k) = h_3(a_1) \cdots h_3(a_k)$$

is in L', and hence $a_1 \cdots a_k$ is in $h_3^{-1}(L')$. Thus $h_1^{-1}(L_1 - \{\epsilon\}) \cap h_2(L_2 - \{\epsilon\}) = h_3^{-1}(L')$ and thus is in $\mathscr{F}(L')$.

We next observe that

(3) if L_3 is in $\mathscr{M}(L_1 - \{\epsilon\})$, then $L_3 \cap h_3^{-1}(L_2 - \{\epsilon\})$ is in $\mathscr{F}(L - \{\epsilon\})$ for every homomorphism h_3.

For let $L_3 = M(L_1 - \{\epsilon\})$ for some ϵ-free a-transducer M. By Proposition 3.1,

$M(L_1 - \{\epsilon\}) = h_2(h_1^{-1}(L_1 - \{\epsilon\}) \cap R)$ for some R in $\mathscr{R}$, homomorphism h_1, and ϵ-free homomorphism h_2. Let $h_4 = h_3 h_2$. Then

$$\begin{aligned} L_3 \cap h_3^{-1}(L_2 - \{\epsilon\}) &= h_2(h_1^{-1}(L_1 - \{\epsilon\}) \cap R) \cap h_3^{-1}(L_2 - \{\epsilon\}) \\ &= h_2[h_1^{-1}(L_1 - \{\epsilon\}) \cap R \cap h_2^{-1}h_3^{-1}(L_2 - \{\epsilon\})] \\ &= h_2[h_1^{-1}(L_1 - \{\epsilon\}) \cap h_4^{-1}(L_2 - \{\epsilon\}) \cap R]. \end{aligned}$$

By (2) and (1), $h_1^{-1}(L_1 - \{\epsilon\}) \cap h_4^{-1}(L_2 - \{\epsilon\})$ is in $\mathscr{F}(L') = \mathscr{F}(L - \{\epsilon\})$. Thus $h_2[h_1^{-1}(L_1 - \{\epsilon\}) \cap h_4^{-1}(L_2 - \{\epsilon\}) \cap R] = L_3 \cap h_3^{-1}(L_2 - \{\epsilon\})$ is in $\mathscr{F}(L - \{\epsilon\})$.

We are now ready to consider $\mathscr{M}(L_1 - \{\epsilon\}) \cap \mathscr{M}(L_2 - \{\epsilon\})$. Let L_4 be in $\mathscr{M}(L_1 - \{\epsilon\})$ and L_5 in $\mathscr{M}(L_2 - \{\epsilon\})$. By Proposition 3.1, $L_5 = h_6[h_5^{-1}(L_2 - \{\epsilon\}) \cap R_1]$, where h_5 and h_6 are homomorphisms, with h_6 ϵ-free and R_1 in $\mathscr{R}$. Now

$$\begin{aligned} L_4 \cap L_5 &= L_4 \cap h_6[h_5^{-1}(L_2 - \{\epsilon\}) \cap R_1] \\ &= h_6[h_6^{-1}(L_4) \cap h_5^{-1}(L_2 - \{\epsilon\}) \cap R_1]. \end{aligned}$$

Since $\mathscr{M}(L_1 - \{\epsilon\})$ is closed under inverse homomorphism and intersection with regular sets, $h_6^{-1}(L_4) \cap R_1$ is in $\mathscr{M}(L_1 - \{\epsilon\})$. By (3), $(h_6^{-1}(L_4) \cap R_1) \cap h_5^{-1}(L_2 - \{\epsilon\})$ is in $\mathscr{F}(L - \{\epsilon\})$, whence $h_6[h_6^{-1}(L_4) \cap R_1 \cap h_5^{-1}(L_2 - \{\epsilon\})] = L_4 \cap L_5$ is in $\mathscr{F}(L - \{\epsilon\})$.

THEOREM 4.1. *Let L_1 and L_2 be nonempty languages. Let h and h' be one to one homomorphisms on L_1 and L_2, resp., such that $\Sigma_{h(L_1)} \cap \Sigma_{h'(L_2)} = \phi$. Let c_1 and c_2 be two symbols not in $\Sigma_{h(L_1)} \cup \Sigma_{h'(L_2)}$. Then*

(a) $H(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2)) = \mathscr{F}(\text{Shuff}((h(L_1)\, c_1)^+, (h'(L_2)\, c_2)^+))$ *if L_1 or L_2 is ϵ-free,*

(b) $H(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2)) = \mathscr{F}(\text{Shuff}((h(L_1)\, c_1)^*, (h'(L_2)\, c_2)^*))$ *if ϵ is in $L_1 \cap L_2$,*

and

(c) $\hat{H}(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2)) = \hat{\mathscr{F}}(\text{Shuff}((h(L_1)\, c_1)^*, (h'(L_2)\, c_2)^*))$.

Proof. Since $\mathscr{F}(L) = \mathscr{F}(h''(L))$ and $\hat{\mathscr{F}}(L) \subseteq \hat{\mathscr{F}}(h''(L))$ for each one to one homomorphism h'' on L, it suffices to assume that h and h' are the identity functions and $\Sigma_{L_1} \cap \Sigma_{L_2} = \phi$.

(a) Let $L = \text{Shuff}((L_1c_1)^+, (L_2c_2)^+)$. Suppose L_i is ϵ-free for some i in $\{1, 2\}$, say L_1 is ϵ-free. Then $H(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2)) = H(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2 - \{\epsilon\}))$. If $L_2 = \{\epsilon\}$, then (a) is easily seen to hold. Suppose $L_2 - \{\epsilon\} \neq \phi$. It is straightforward to verify that

$$\mathscr{F}(\text{Shuff}((L_1c_1)^+, (L_2c_2)^+) = \mathscr{F}(\text{Shuff}((L_1c_1)^+, ((L_2 - \{\epsilon\})\, c_2)^+)).$$

An analogous result holds if L_2 is ϵ-free. Thus we may assume that L_1 and L_2 are ϵ-free.

By Theorem 3.1, $\mathscr{M}(L_ic_i)^+) = \mathscr{F}(L_i - \{\epsilon\}) = \mathscr{F}(L_i)$ for each i. Then

$$\begin{aligned}\mathscr{F}(L_1) \wedge \mathscr{F}(L_2) &= \mathscr{M}((L_1c_1)^+) \wedge \mathscr{M}((L_2c_2)^+)\\ &\subseteq \mathscr{F}(L - \{\epsilon\}), \quad \text{by (c) of Lemma 4.1,}\\ &= \mathscr{F}(L), \quad \text{since } L \text{ is } \epsilon\text{-free.}\end{aligned}$$

Hence $H(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2)) \subseteq \mathscr{F}(L)$.

Consider the reverse containment. For $i = 1, 2$, let h_i be the homomorphism on $\Sigma_{L_1} \cup \Sigma_{L_2} \cup \{c_1, c_2\}$ defined by $h_i(a) = a$ for a in $\Sigma_{L_i} \cup \{c_i\}$ and $h_i(a) = \epsilon$ otherwise. Then $L = h_1^{-1}((L_1c_1)^+) \cap h_2^{-1}((L_2c_2)^+)$. Hence L is in $\mathscr{F}(L_1) \wedge \mathscr{F}(L_2) \subseteq H(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2))$. Therefore $\mathscr{F}(L) \subseteq H(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2))$, whence equality.

(b) An argument similar to that in the second part of (a) shows that

$$\mathscr{F}(\text{Shuff}((L_1c_1)^*, (L_2c_2)^*) \subseteq H(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2)).$$

Consider the reverse containment. By Theorem 5.3.1, $\mathscr{M}((L_ic_i)^*) = \mathscr{F}(L_i)$ for each i. Then

$$\begin{aligned}\mathscr{F}(L_1) \wedge \mathscr{F}(L_2) &= \mathscr{M}((L_1c_1)^*) \wedge \mathscr{M}((L_2c_2)^*)\\ &\subseteq \mathscr{F}(\text{Shuff}((L_1c_1)^*, (L_2c_2)^*), \quad \text{by (b) of Lemma 4.1.}\end{aligned}$$

(c) If L_1 and L_2 both contain ϵ, then

$$\begin{aligned}\hat{H}(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2)) &= \hat{H}(H(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2)))\\ &= \hat{H}(\mathscr{F}(\text{Shuff}((L_1c_1)^*, (L_2c_2)^*))), \quad \text{by (b),}\\ &= \hat{\mathscr{F}}(\text{Shuff}((L_1c_1)^*, (L_2c_2)^*)).\end{aligned}$$

Suppose L_1 or L_2 does not contain ϵ. Then

$$\begin{aligned}\hat{H}(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2)) &= \hat{H}(H(\mathscr{F}(L_1) \wedge \mathscr{F}(L_2)))\\ &= \hat{H}(\mathscr{F}(\text{Shuff}((L_1c_1)^+, (L_2c_2)^+))), \quad \text{by (a),}\\ &= \hat{\mathscr{F}}(\text{Shuff}((L_1c_1)^+, (L_2c_2)^+)).\end{aligned}$$

Since

$$\text{Shuff}((L_1c_1)^*, (L_2c_2)^*) = (L_1c_1)^+ \cup (L_2c_2)^+ \cup \{\epsilon\} \cup \text{Shuff}((L_1c_1)^+, (L_2c_2)^+)$$

and

$$\begin{aligned}\text{Shuff}((L_1c_1)^+, (L_2c_2)^+) = \text{Shuff}((L_1c_1)^*, (L_2c_2)^*) \cap [(\Sigma_{L_1} \cup \Sigma_{L_2} \cup \{c_1, c_2\})^*\\ - ((\Sigma_{L_1}^*c_1)^* \cup (\Sigma_{L_2}^*c_2)^*)],\end{aligned}$$

$$\hat{\mathscr{F}}(\text{Shuff}((L_1c_1)^+, (L_2c_2)^+)) = \hat{\mathscr{F}}(\text{Shuff}((L_1c_1)^*, (L_2c_2)^*)),$$

completing the proof.

COROLLARY 1. *If* $\mathscr{L}_1$ *and* $\mathscr{L}_2$ *are (full) principal* AFL, *then so is* $H(\mathscr{L}_1 \wedge \mathscr{L}_2)$ $(\hat{H}(\mathscr{L}_1 \wedge \mathscr{L}_2))$.

From the corollary to Theorem 1.2 in [11],

$$H(\mathscr{L}_1 \wedge \cdots \wedge \mathscr{L}_n) = H(H(\mathscr{L}_1 \wedge \cdots \wedge \mathscr{L}_{n-1}) \wedge \mathscr{L}_n)$$

for $n \geqslant 3$ AFL $\mathscr{L}_1 ,..., \mathscr{L}_n$, and

$$\hat{H}(\mathscr{L}_1 \wedge \cdots \wedge \mathscr{L}_n) = \hat{H}(\hat{H}(\mathscr{L}_1 \wedge \cdots \wedge \mathscr{L}_{n-1}) \wedge \mathscr{L}_n)$$

for $n \geqslant 3$ full AFL $\mathscr{L}_1 ,..., \mathscr{L}_n$. Thus we have

COROLLARY 2. *If* $\mathscr{L}_1 ,..., \mathscr{L}_n$ *are (full) principal* AFL, *then so is* $H(\mathscr{L}_1 \wedge \cdots \wedge \mathscr{L}_n)$ $(\hat{H}(\mathscr{L}_1 \wedge \cdots \wedge \mathscr{L}_n))$.

From Theorem 4.1 there follows

COROLLARY 3. *An* AFL $\mathscr{L}$ *is closed under intersection if and only if it is closed under Shuff, i.e.,* $\text{Shuff}(L_1, L_2)$ *is in* $\mathscr{L}$ *for each* L_1, L_2 *in* $\mathscr{L}$.

The second operator to be discussed in this section is substitution.

Notation. For all families of languages $\mathscr{L}_1$ and $\mathscr{L}_2$, let $\mathscr{L}_1 \hat{\sigma} \mathscr{L}_2$ ($\mathscr{L}_1 \sigma \mathscr{L}_2$) be the family of all $\tau(L)$, where L is in $\mathscr{L}_1$ and τ is a (ϵ-free) substitution[14] such that $\tau(a)$ is in $\mathscr{L}_2$ for each a in Σ_L.

If $\mathscr{L}_1$ and $\mathscr{L}_2$ are AFL, then so are $\mathscr{L}_1 \sigma \mathscr{L}_2$ ([11], Corollary 1 of Theorem 4.2) and $\mathscr{L}_1 \hat{\sigma} \mathscr{L}_2$ [7]. If $\mathscr{L}_1$ and $\mathscr{L}_2$ are full AFL, then $\mathscr{L}_1 \sigma \mathscr{L}_2 = \mathscr{L}_1 \hat{\sigma} \mathscr{L}_2$ is a full AFL by Remark 4 after Theorem 4.1 of [11].

We now turn to an algebraic proof that $\mathscr{F}(L_1) \sigma \mathscr{F}(L_2)$ $(\hat{\mathscr{F}}(L_1) \hat{\sigma} \hat{\mathscr{F}}(L_2))$ is a (full) principal AFL. First though, we need two lemmas.

LEMMA 4.2. *Let* L *be an* ϵ*-free language and* $M = (K_1, \Sigma_1, \Sigma_2, H, p_0, F)$ *a* 1*-bounded,* ϵ*-output bounded a-transducer such that* H *contains no sequence of elements* $(p_0, u_1, v_1, p_1),..., (p_t, u_{t+1}, v_{t+1}, p_{t+1})$ *with the properties that* $v_i = \epsilon$ *for each* i *and* p_{t+1} *is in* F. *Then there exists a* 1*-bounded,* ϵ*-output bounded a-transducer* $M' = (K', \Sigma_1, \Sigma_2, H', q_0, F)$ *such that* $M'(L) = M(L)$, q_0 *is not in* F, H' *contains no element of the form* (q_0, ϵ, v, q), *and* H' *contains no sequence of elements* $(q_0, u_1, v_1, q_1),..., (q_t, u_{t+1}, v_{t+1}, q_{t+1})$ *with the properties that* $v_i = \epsilon$ *for each* i *and* q_{t+1} *is in* F.

[14] Let L be a language and for each a in Σ_L let L_a be a language. Let τ be the function defined on $\Sigma_L{}^*$ by $\tau(\epsilon) = \{\epsilon\}$, $\tau(a) = L_a$ for each a in Σ_L, and $\tau(a_1 \cdots a_n) = \tau(a_1) \cdots \tau(a_n)$ for each a_i in Σ_L and $n \geqslant 1$. Then τ is called a *substitution.* τ is extended to $2^{\Sigma_L^*}$ by defining $\tau(X) = \bigcup_{x \text{ in } X} \tau(x)$ for all $X \subseteq \Sigma_L{}^*$. τ is called ϵ*-free* if $\tau(a)$ is ϵ-free for each a in Σ_L.

Proof. First assume that there exists a in Σ_1 such that each word in L begins with a. For each p in K_1 let p' be a new symbol. Let q_0 be a new symbol. Let $K' = K_1 \cup \{q_0, p'/p \text{ in } K_1\}$ and let H' consist of the following 4-tuples:

(1) (p_1, u, v, p_2) for all (p_1, u, v, p_2) in H.

(2) (q_0, a, v, p) for all (p_0, a, v, p) in H.

(3) $(p_1', \epsilon, v, p_2')$ for all (p_1, ϵ, v, p_2) in H.

(4) (q_0, a, v, p') for all (p_0, ϵ, v, p) in H.

(5) (p_1', ϵ, v, p_2) for all (p_1, a, v, p_2) in H.

Let $M' = (K', \Sigma_1, \Sigma_2, H', q_0, F)$. Clearly M' satisfies the conclusion of the lemma.

Now let L be any ϵ-free language. For each a in Σ_1, let $L_a = a\Sigma_1{}^* \cap L$. Clearly $M(L) = \bigcup_{L_a \neq \phi} M(L_a)$. By the previous paragraph, for each a, $L_a \neq \phi$, there exists an a-transducer $M_a' = (K_a', \Sigma_1, \Sigma_2, H_a', q_0, F_a)$ satisfying the conclusion of the lemma for L_a. Since q_0 is not in F_a, we may assume that $K_a' \cap K_b' = \{q_0\}$ for each a and b in Σ_1, $a \neq b$. Then $M' = (\cup K_a', \Sigma_1, \Sigma_2, \cup H_a', \{q_0\} \cup F_a)$ satisfies the conclusion of the lemma for L.

For the next lemma we need a particular kind of substitution, first defined in [10].

Notation. Given languages L_1 and L_2, with $\Sigma_{L_1} \cap \Sigma_{L_2} = \phi$, let τ_{L_2} be the substitution on $\Sigma_{L_1}^*$ defined by $\tau_{L_2}(a) = aL_2$ for each a in Σ_{L_1}.

LEMMA 4.3. *Let L_1 and L_2 be nonempty languages such that L_1 is ϵ-free and $\Sigma_{L_1} \cap \Sigma_{L_2} = \phi$. Let c be a symbol not in $\Sigma_{L_1} \cup \Sigma_{L_2}$. Then*

$$\mathscr{M}(L_1)\ \sigma\ \mathscr{M}(L_2) \subseteq \mathscr{M}(\tau_{(L_2c)^+}(L_1)).$$

Proof. Let L be in $\mathscr{M}(L_1)\ \sigma\ \mathscr{M}(L_2)$. Then $L = \tau(L_3)$, where L_3 is in $\mathscr{M}(L_1)$ and, for each b in Σ_{L_3}, $\tau(b)$ is an ϵ-free set in $\mathscr{M}(L_2)$. Since L_1 is ϵ-free, so is L_3. By Lemmas 3.2 and 4.2, we may assume that $L_3 = M_1(L_1)$, where $M_1 = (K_1, \Sigma_{L_1}, \Sigma_{L_3}, H_1, p_0, F_1)$ is a 1-bounded, ϵ-output bounded a-transducer such that p_0 is not in F_1, H_1 contains no element of the form (p_0, ϵ, v, p), and H_1 contains no sequence of elements $(p_0, u_1, v_1, p_1), \ldots, (p_t, u_{t+1}, v_{t+1}, p_{t+1})$ with the properties that $v_i = \epsilon$ for each i and p_{t+1} is in F_1. For each b in Σ_{L_3}, there exists an ϵ-free a-transducer $M_b = (K_b, \Sigma_{L_2}, H_b, q_b, F_b)$ such that q_b is not in F_b and $\tau(b) = M_b(L_2)$. Clearly we may assume that all sets of states are pairwise disjoint.

Let w be a specific element in L_2. Let

$$M = \Big(K_M, \Sigma_{L_1} \cup \Sigma_{L_2} \cup \{c\}, \bigcup_{b \text{ in } \Sigma_{L_3}} \Sigma_b, H, \bar{p}_0, F\Big)$$

be the a-transducer where

$$K_M = (K_1 \times \{\epsilon\}) \cup \Big(K_1 \times \bigcup_{b \text{ in } \Sigma_{L_3}} K_b\Big), \qquad \bar{p}_0 = (p_0, \epsilon), \qquad F = F_1 \times \{\epsilon\},$$

and H contains the following 4-tuples:

(1) $((p, \epsilon), awc, \epsilon, (p', \epsilon))$ if (p, a, ϵ, p') is in H_1, with a in $\Sigma_{L_1} \cup \{\epsilon\}$.

(2) $((p, \epsilon), a, \epsilon, (p', q_b))$ if (p, a, b, p') is in H_1, with b in Σ_{L_3} and a in $\Sigma_{L_1} \cup \{\epsilon\}$.

(3) $((p, q), u, v, (p, q'))$, for each p in K_1, if (q, u, v, q') is in H_b.

(4) $((p, q), c, \epsilon, (p, \epsilon))$ for all p in K_1, b in Σ_{L_3}, and q in F_b.

Intuitively, M simulates the operation of M_1 on the first coordinate and M_b on L_2 on the second coordinate. The idea of the simulation is as follows. While M_1 reads a_i, a_i in $\Sigma_{L_1} \cup \{\epsilon\}$, and outputs b_i, b_i in $\Sigma_{L_3} \cup \{\epsilon\}$, M reads a_iL_2c and outputs $\tau(b_i) = M_{b_i}(L_2)$. The start and end of this subroutine always occurs at states in $K_1 \times \{\epsilon\}$. [Initially, M_1 reads a symbol a_1 in Σ_{L_1} since H_1 contains no rule of the form (p_0, ϵ, v, p). Hence $a_1(L_2c)^+$ supplies an appropriate number of occurrences of L_2c.] If M_1 reads a_i in $\Sigma_{L_1} \cup \{\epsilon\}$ and outputs ϵ, then by (1), M reads exactly one word, a_iwc, in a_iL_2c, and outputs $\{\epsilon\} = \tau(\epsilon)$. If M_1 reads a_i in $\Sigma_{L_1} \cup \{\epsilon\}$ and outputs b_i in Σ_{L_3}, then by (2), (3) and (4), M reads a_iL_2c and outputs $M_{b_i}(L_2) = \tau(b_i)$. In particular, by (2), M reads a_i, outputs ϵ, and goes to the start state of M_b. By (3), M reads all words of L_2 and outputs all words of $M_b(L_2)$. By (4), M is in an accepting state of M_b and, under c, gets ready to simulate the processing of M_1 on a_{i+1}.

More formally, we now show that $M(\tau_{(L_2c)^+}(L_1)) = \tau(M_1(L_1))$. Suppose $(p_0, a_1, b_1, p_1), \ldots, (p_{k-1}, a_k, b_k, p_k)$ is a sequence of elements in H_1, with p_k in F_1 and $a_1 \cdots a_k$ in L_1. Let $m(1) < \cdots < m(t)$ be the indices for which a_i is in Σ_{L_1}. Note that $m(1) = 1$ since H_1 has no elements of the form (p_0, ϵ, v, p). Let U be the set of all words $z_{m(1)} \cdots z_{m(t)}$, where for $1 \leqslant i < t$,

$$z_{m(i)} = a_{m(i)}w_{m(i)}c_{m(i)}w_{m(i)+1}c_{m(i)+1} \cdots w_{m(i+1)-1}c_{m(i+1)-1}z_{m(t)}$$

$$= a_{m(t)}w_{m(t)}c_{m(t)} \cdots w_kc_k,$$

and for all j, $1 \leqslant j \leqslant k$, $c_j = c$ and w_j is obtained as follows. If $b_j = \epsilon$ then $w_j = w$. If b_j is in Σ_{L_3} then w_j is in L_2. Obviously $U \subseteq a_{m(1)}(L_2c)^+ \cdots a_{m(t)}(L_2c)^+$. From (1), (2), (3), and (4), $\tau(b_1) \cdots \tau(b_k) \subseteq M(U)$. Therefore $\tau M_1(L_1) \subseteq M(\tau_{(L_2c)^+}(L_1))$.

To see the reverse inclusion, let v be in $M(\tau_{(L_2c)^+}(L_1))$. Then there exist u in $\tau_{(L_2c)^+}(L_1)$ and a sequence

(5) $(r_1, u_1, v_1, r_2), \ldots, (r_k, u_k, v_k, r_{k+1})$

of elements in H such that $r_1 = (p_0, \epsilon)$, r_{k+1} is in F, $u = u_1 \cdots u_k$, and $v = v_1 \cdots v_k$.

Let $m(1) < \cdots < m(t)$ be those integers i such that $r(i)$ is in $K_1 \times \{\epsilon\}$. For each j, let $r_{m(j)} = (p_j', \epsilon)$. Then $m(1) = 1$ and $m(t) = k + 1$. For each j, $1 \leqslant j < t$, consider

$$(6) \quad (r_{m(j)}, u_{m(j)} \cdots u_{m(j+1)-1}, \epsilon) \vdash \cdots \vdash (r_{m(j+1)}, \epsilon, v_{m(j)} \cdots v_{m(j+1)-1})$$

as obtained from (5). Two possibilities arise.

(α) $m(j + 1) - m(j) = 1$. Rules (2) and (4) imply that (6) is obtained by an application of a type 1 rule. Thus $v_{m(j)} = \epsilon$, $u_{m(j)}$ is of the form $a_{m(j)}wc$, and $(p_j', a_{m(j)}, b_{m(j)}, p_{j+1}')$ is in H_1, with $b_{m(j)} = \epsilon$. Thus $v_{m(j)}$ is in $\tau(b_{m(j)})$.

(β) $m(j + 1) - m(j) > 1$. Then a type 1 rule is not used. Thus type 2, 3, and 4 rules only are used, with exactly one occurrence of a type 2 rule and exactly one occurrence of a type 4 rule. Since a state of the form (p, ϵ) can be entered from another state only on reading (in part) an occurrence of c (rules (1) and (4)), $u_{m(j)} \cdots u_{m(j+1)-1}$ is in $u_{m(j)}L_2c$, with $u_{m(j)+1} \cdots u_{m(j+1)-1}$ in L_2c. Since rule (2) is used, $(p_m', a_{m(j)}, b_{m(j)}, p_{m+1}')$ is in H_1 for some $b_{m(j)}$ in Σ_{L_3}, with $a_{m(j)} = u_{m(j)}$. From rules (3) and (4), $v_{m(j)} \cdots v_{m(j+1)-1}$ is in $M_{b_{m(j)}}(a_{m(j)}L_2c) = \tau(b_{m(j)})$. Since $u_1 \cdots u_t$ is in $\tau_{(L_2c)^+}(L_1)$, with u_i in $\Sigma_{L_2} \cup \{\epsilon\}$ if i is not of the form $m(j)$ for some j, and $u_{m(j)}$ is either $a_{m(j)}$ in $\Sigma_{L_1} \cup \{\epsilon\}$ or $u_{m(j)} = a_{m(j)}wc$, $a_{m(j)}$ in $\Sigma_{L_1} \cup \{\epsilon\}$, it follows that $a_{m(1)} \cdots a_{m(k)}$ is in L_1. Then

$$(p_1', a_{m(1)}, b_{m(1)}, p_2') \cdots (p_{k-1}', a_{m(k)}, b_{m(k)}, p_{k+1}')$$

is a sequence of elements in H_1, with $p_1' = p_0$, p_k' in F_1, and $a_{m(1)} \cdots a_{m(k)}$ in L_1. Then $b_{m(1)} \cdots b_{m(k)}$ is in $M_1(a_{m(1)} \cdots a_{m(k)})$ and $v_1 \cdots v_t$ is in $\tau(b_{m(1)} \cdots b_{m(k)})$, i.e., v is in $\tau M_1(L_1)$. Thus $M(\tau_{(L_2c)^+}(L_1)) \subseteq \tau M_1(L_1)$, whence equality.

In view of (1) and the hypothesis on H_1, H contains no sequence of elements of the form $(r_0, u_1, v_1, r_1),\ldots, (r_t, u_{t+1}, v_{t+1}, r_{t+1})$ such that $v_i = \epsilon$ for all i, $r_0 = \bar{p}_0$, and r_{t+1} is in $F = F_1 \times \{\epsilon\}$. In view of Lemma 3.2, in order to complete the proof of the lemma it suffices to show that M is ϵ-output bounded. Since M_1 is ϵ-output bounded, there exists k_1 such that M_1 has at most k_1 consecutive moves with ϵ as output. Each type 1 rule simulates an ϵ-rule in M_1. Types 2 and 4 rules add ϵ-output rules to H. Since each M_b is ϵ-output free, no type 3 rule is an ϵ-output rule. Thus M has at most $k_1 + 2$ consecutive ϵ-output rules, namely one of type 4, followed by k_1 of type 1, followed by one of type 2. Hence M is ϵ-output bounded.

We also need the following result which, in essense, has been proved elsewhere.

PROPOSITION 4.1. *For each family $\mathscr{L}$ of languages, $\mathscr{F}(\mathscr{L}) = \mathscr{R}_0 \hat{\sigma} \mathscr{M}(\mathscr{L})$ and $\hat{\mathscr{F}}(\mathscr{L}) = \mathscr{R} \hat{\sigma} \hat{\mathscr{M}}(\mathscr{L}) = \mathscr{R}_0 \hat{\sigma} \hat{\mathscr{M}}(\mathscr{L})$.*

Proposition 4.1 follows from Corollary 2 of Theorem 4.2 of [4], with slight modification due to ϵ, and from the corollary to Theorem 4.1 of [4].

We are now ready for the second major result of this section.

THEOREM 4.2. *Let L_1 and L_2 be nonempty languages, with $\Sigma_{L_1} \cap \Sigma_{L_2} = \phi$. For new symbols c_1 and c_2*

(a) $\mathscr{F}(L_1)\, \sigma\, \mathscr{F}(L_2) = \mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^+))$ *if L_1 is ϵ-free,*
(b) $\mathscr{F}(L_1)\, \sigma\, \mathscr{F}(L_2) = \mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^*))$ *if L_1 contains ϵ, and*
(c) $\hat{\mathscr{F}}(L_1)\, \sigma\, \hat{\mathscr{F}}(L_2) = \hat{\mathscr{F}}(L_1)\, \hat{\sigma}\, \hat{\mathscr{F}}(L_2) = \hat{\mathscr{F}}(\tau_{(L_2c_2)^+}((L_1c_1)^+))$.

Proof. (a) Suppose L_1 is ϵ-free. Clearly $\tau_{(L_2c_2)^+}((L_1c_1)^+)$ is in $\mathscr{F}(L_1)\, \sigma\, \mathscr{F}(L_2)$. By Corollary 1 of Theorem 4.2 of [11], $\mathscr{L}_1\, \sigma\, \mathscr{L}_2$ is an AFL if $\mathscr{L}_1$ and $\mathscr{L}_2$ are AFL. Thus $\mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^+)) \subseteq \mathscr{F}(L_1)\, \sigma\, \mathscr{F}(L_2)$.

Consider the reverse containment. Note that

$$\mathscr{M}(L_1)\, \sigma\, \mathscr{M}(L_2) \subseteq \mathscr{M}(\tau_{(L_2c_2)^+}(L_1)), \qquad \text{by Lemma 4.3,}$$
$$\subseteq \mathscr{F}(\tau_{(L_2c_2)^+}(L_1)).$$

Let h be the homomorphism on $(\Sigma_{L_1} \cup \Sigma_{L_2} \cup \{c_1, c_2\})^*$ defined by $h(c_1) = \epsilon$ and $h(x) = x$ for x in $\Sigma_{L_1} \cup \Sigma_{L_2} \cup \{c_2\}$. Let

$$U = \tau_{(L_2c_2)^+}((L_1c_1)^+) \cap ((\Sigma_{L_1} \cup \Sigma_{L_2} \cup \{:\})^+ c_1).$$

Then U is in $\mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^+))$. Since L_1 is ϵ-free, $\tau_{(L_2c_2)^+}(L_1) = h(U) = h(U) - \{\epsilon\}$. Since h is ϵ-limited on U, $\tau_{(L_2c_2)^+}(L_1) = h(U) - \{\epsilon\}$ is in $\mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^+))$ by Corollary 5 of Theorem 2.1 of [4]. Thus

$$\mathscr{M}(L_1)\, \sigma\, \mathscr{M}(L_2) \subseteq \mathscr{F}(\tau_{(L_2c_2)^+}(L_1))$$
$$\subseteq \mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^+)).$$

As noted in [4], each AFL is closed under substitution into ϵ-free regular sets. Thus

$$\mathscr{R}_0\, \sigma\, (\mathscr{M}(L_1)\, \sigma\, \mathscr{M}(L_2)) \subseteq \mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^+)).$$

Finally, note that

$$\begin{aligned}
\mathscr{F}(L_1)\, \sigma\, \mathscr{F}(L_2) &= \mathscr{F}(L_1)\, \sigma\, (\mathscr{R}_0\, \sigma\, \mathscr{M}(L_2)), && \text{by Proposition 4.1,}\\
&= (\mathscr{F}(L_1)\, \sigma\, \mathscr{R}_0)\, \sigma\, \mathscr{M}(L_2), && \text{by associativity of } \sigma,^{15}\\
&= \mathscr{F}(L_1)\, \sigma\, \mathscr{M}(L_2), && \text{since each AFL is closed under substitution by } \epsilon\text{-free regular sets,}\\
&= \mathscr{R}_0\, \sigma\, (\mathscr{M}(L_1)\, \sigma\, \mathscr{M}(L_2)), && \text{by Proposition 4.1 and associativity of } \sigma.
\end{aligned}$$

Hence $\mathscr{F}(L_1)\, \sigma\, \mathscr{F}(L_2) \subseteq \mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^+))$, whence equality.

[15] It was shown in [7] that $\hat{\sigma}$ is associative on families of languages closed under isomorphism, i.e., $(\mathscr{L}_1 \hat{\sigma} \mathscr{L}_2)\, \hat{\sigma} \mathscr{L}_3 = \mathscr{L}_1 \hat{\sigma} (\mathscr{L}_2 \hat{\sigma} \mathscr{L}_3)$ if $\mathscr{L}_1$, $\mathscr{L}_2$, and $\mathscr{L}_3$ are closed under isomorphism. The same proof shows that σ is associative on such families.

(b) Suppose L_1 contains ϵ. Since (b) is true if $L_1 = \{\epsilon\}$, we may assume that $L_1 \neq \{\epsilon\}$. Then

$$\begin{aligned}\mathscr{F}(L_1)\,\sigma\,\mathscr{F}(L_2) &= \mathscr{F}(L_1 - \{\epsilon\})\,\sigma\,\mathscr{F}(L_2) \cup \{L \cup \{\epsilon\}/L \text{ in } \mathscr{F}(L_1 - \{\epsilon\})\,\sigma\,\mathscr{F}(L_2)\}\\ &= \mathscr{F}(\tau_{(L_2c_2)^+}(((L_1 - \{\epsilon\})\,c_1)^+))\\ &\quad \cup \{L \cup \{\epsilon\}/L \text{ in } \mathscr{F}(\tau_{(L_2c_2)^+}(((L_1 - \{\epsilon\})\,c_1)^+))\}.\end{aligned}$$

As is easily seen,

$$\mathscr{F}(\tau_{(L_2c_2)^+}(((L_1 - \{\epsilon\})\,c_1)^+)) = \mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^+)).$$

Thus

$$\begin{aligned}\mathscr{F}(L_1)\,\sigma\,\mathscr{F}(L_2) &= \mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^+)) \cup \{L \cup \{\epsilon\}/L \text{ in } \mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^+))\}\\ &= \mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^*)).\end{aligned}$$

(c) Since (c) is obviously true if $L_1 = \{\epsilon\}$, we may assume that $L_1 \neq \{\epsilon\}$. Then

$$\begin{aligned}\hat{\mathscr{F}}(L_1)\,\sigma\,\hat{\mathscr{F}}(L_2) &= \hat{\mathscr{F}}(\mathscr{F}(L_1 - \{\epsilon\}))\,\sigma\,\hat{\mathscr{F}}(\mathscr{F}(L_2))\\ &= \hat{\mathscr{F}}(\mathscr{F}(L_1 - \{\epsilon\})\,\sigma\,\mathscr{F}(L_2)), \quad \text{by Corollary 5 to Theorem 4.2 of [11],}\\ &= \hat{\mathscr{F}}(\mathscr{F}(\tau_{(L_2c_2)^+}(((L_1 - \{\epsilon\})\,c_1)^+))), \quad \text{by (a),}\\ &= \hat{\mathscr{F}}(\mathscr{F}(\tau_{(L_2c_2)^+}((L_1c_1)^+)))\\ &= \hat{\mathscr{F}}(\tau_{(L_2c_2)^+}((L_1c_1)^+)).\end{aligned}$$

COROLLARY 1. *If $\mathscr{L}_1$ and $\mathscr{L}_2$ are (full) principal* AFL, *then so is $\mathscr{L}_1\,\sigma\,\mathscr{L}_2$ ($\mathscr{L}_1\,\hat{\sigma}\,\mathscr{L}_2 = \mathscr{L}_1\,\sigma\,\mathscr{L}_2$).*

Proof. Let $\mathscr{L}_1 = \mathscr{F}(L_1)$ and $\mathscr{L}_2 = \mathscr{F}(L_2)$ ($\mathscr{L}_1 = \hat{\mathscr{F}}(L_1)$) and $\mathscr{L}_2 = \hat{\mathscr{F}}(L_2)$). If $L_1 \neq \phi$ and $L_2 \neq \phi$, then the corollary follows immediately from Theorem 4.2. If $L = \phi$, then $\mathscr{F}(L_1) = \mathscr{F}(\{a\})$ and $\hat{\mathscr{F}}(L_1) = \hat{\mathscr{F}}(\{a\})$. If $L_2 = \phi$, then $\mathscr{F}(L_2) = \mathscr{F}(\{a\})$ and $\hat{\mathscr{F}}(L_2) = \hat{\mathscr{F}}(\{a\})$. Thus the corollary is also true if $L_1 = \phi$ or $L_2 = \phi$.

By induction, we get

COROLLARY 2. *If $\mathscr{L}_1, \ldots, \mathscr{L}_n$ are (full) principal* AFL, *then so is $\mathscr{L}_1\,\sigma\,\mathscr{L}_2\,\sigma \cdots \sigma\,\mathscr{L}_n$ ($\mathscr{L}_1\,\hat{\sigma} \cdots \hat{\sigma}\,\mathscr{L}_n = \mathscr{L}_1\,\sigma \cdots \sigma\,\mathscr{L}_n$).*

COROLLARY 3. *An* AFL *$\mathscr{L}$ is closed under substitution if and only if $\mathscr{L}$ is closed under all substitutions of the form $\tau_{L_2}(L_1)$ for L_1 and L_2 in $\mathscr{L}$.*

References

1. A. V. Aho, "Nested Stack Automata," *J. Assoc. Comput. Mach.* **19** (1969), 383–406.
2. N. Chomsky, "On Certain Formal Properties of Grammars," *Information and Control* **2** (1959), 137–167.
3. N. Chomsky, *Context-Free Grammars and Pushdown Storage*, MIT Research Laboratory of Electronics, Quarterly Progress Report 65, 1962, pp. 187–194.
4. S. Ginsburg and S. A. Greibach, "Abstract Families of Languages," Studies in Abstract Families of Languages, *Mem. Amer. Math. Soc. Memoir* **87** (1969), 1–32.
5. S. Ginsburg, S. A. Greibach, and J. Hopcroft, "Pre-AFL," Studies in Abstract Families of Languages, *Mem. Amer. Math. Soc., Memoir* **87** (1969), 41–51.
6. S. Ginsburg and E. H. Spanier, "Finite-Turn Pushdown Automata," *SIAM J. of Control* **4** (1966), 429–453.
7. S. Ginsburg and E. H. Spanier, "Substitution in Families of Languages," *Information Sciences* **2** (1970), 83–110.
8. S. A. Greibach, "A New Normal-Form Theorem for Context-Free Phrase Structure Grammars," *Journal of the Association for Computing Machinery* **12** (1965), 42–52.
9. S. A. Greibach, "An Infinite Hierarchy of Context-Free Grammars," *J. Assoc. Comput. Mach.* **16** (1969), 91–106.
10. S. A. Greibach, "Checking Automata and One-Way Stack Languages," *J. Comput. System Sci.* **3**, 1969, 196–217.
11. S. A. Greibach and S. Ginsburg, *Multitape AFA*, SDC document TM-738/051/00, January, 1969.
12. S. A. Greibach and J. Hopcroft, "Independence of AFL Operations," Studies in Abstract Families of Languages, *Mem. Amer. Math. Soc. Memoir* **87** (1969), 33–40.
13. R. E. Stearns, J. Hartmanis, and P. M. Lewis II, "Hierarchies of Memory Limited Computations," *Proceedings of the Sixth Annual Symposium on Switching Circuit Theory and Logical Design*, 1965, pp. 179–190.