One-Way Stack Automata

SEYMOUR GINSBURG

System Development Corporation, Santa Monica, California

SHEILA A. GREIBACH

Harvard University, Cambridge, Massachusetts

AND

MICHAEL A. HARRISON

University of California, Berkeley, California

ABSTRACT. A number of operations which either preserve sets accepted by one-way stack automata or preserve sets accepted by deterministic one-way stack automata are presented. For example, sequential transduction preserves the former; set complementation, the latter. Several solvability questions are also considered.

Introduction

In [5], the notion of a stack automaton, both deterministic and nondeterministic, is defined. This device embodies many features used in the recognition aspect of currently used compilers. It is less powerful than a Turing machine but more potent than a pushdown automaton. Specifically, a stack automaton allows reading, but not writing, in the interior of its stack. (This occurs, for example, in the reading of symbol tables.) It permits writing and erasing only on a last-in first-out basis. The stack automaton also permits reading the input many times (technically, a two-way read which corresponds in one sense to a multipass compiler). In the present paper, we discuss the important case when the stack automaton reads the input tape from left to right only, as in a single-pass compiler. This device, a generalization of the pushdown automaton, is called a "one-way stack automaton." Of special interest are the sets of words recognized by one-way stack automata, hereafter called "languages."

Our motivation in studying one-way stack automata is twofold. First, it is a natural specialization of the stack automata. And second, more of ALGOL can be recognized by this type of device than by a pushdown automaton. Thus sets accepted by one-way stack automata may be better approximations to currently used programming languages than sets recognized by pushdown automata, i.e., context-free languages.

The paper is divided into five sections. In Section 1, one-way stack automata and languages are introduced. In Section 2, various operations which preserve the family of languages are considered. For example, intersection with a regular set preserves

The research reported in this paper was sponsored in part by the Air Force Cambridge Research Laboratories, Office of Aerospace Research, under Contract AF 19(628)-5166, CRL—Algorithmic Languages Program.

languages, as does transformation by a sequential transducer and word reversal. In Section 3, it is shown that languages are not closed under complementation. In Section 4, some closure properties of D-languages (languages accepted by deterministic one-way stack automata) are presented. By a lengthy argument, it is established that D-languages are closed under complementation. D-languages are also closed under transformation by an inverse gsm (generalized sequential machine). Deletion of a word on the left, or on the right, also preserves D-languages. Decision problems are considered in Section 5. In particular, it is recursively solvable to determine if a language is empty or if a D-language equals a particular regular set. It is recursively unsolvable to determine if an arbitrary language is context-free or a D-language.

Since a one-way stack automaton is a complicated device, it is to be expected that the arguments are often quite involved and messy. They frequently require construction of one-way stack automata with special properties. We have used our discretion and, whenever feasible, have either omitted or outlined the justification that a particular one-way stack automaton recognizes exactly a certain prescribed set. The methods employed also indicate alternative (but not necessarily simpler) proofs of known results about context-free languages.

1. *Preliminaries*

In this section, the basic objects with which we are concerned in this paper, namely, the one-way stack automata and the sets recognized or accepted by them, are defined.

Roughly speaking, a one-way stack automaton consists of a "finite-state control," an "input" sequence or tape, and a "stack." The device advances the input tape at most one symbol per move. The stack is a "last-in first-out" store, i.e., it may be written or erased from the right end in the conventional way. In addition, the interior part of the stack may be read but not rewritten.

A one-way stack automaton operates in the following manner. If the device is in a state, reading both an input symbol and a stack symbol, then it simultaneously

(i) goes to another state;

(ii) moves at most one symbol to the right of the input symbol just read;

(iii) does exactly one of two alternatives: (a) it may move its stack pointer (≡ read-write head) one symbol to the left or to the right, or keep it stationary, or (b) if it is reading the rightmost symbol on the stack, then it may write a (possibly empty) finite sequence of symbols onto the stack, simultaneously erasing the symbol just read.

The reader is referred to [5] for a further discussion of a stack automaton as well as the motivation for its definition.

We now formalize the above intuitive description.

Definition. A *one-way stack automaton* (abbreviated "one-way sa") is a 9-tuple $A = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$ satisfying the following conditions:

(1) K is a finite nonempty set (of *states*).

(2) Σ is a finite nonempty set (of *inputs*).

(3) ¢ and \$ are two elements not in Σ (the left and right *endmarkers* for the input).

(4) Γ is a finite nonempty set (of *stack symbols*).

(5) Z_0 is in Γ (the *initial* stack symbol).
(6) δ is a function from $K \times (\Sigma \cup \{\cent, \$\}) \times \Gamma$ into the set[1] of finite subsets[2] of $\{0, 1\} \times K \times \{-1, 0, 1\} \times \Gamma^*$ having the following property: (*) If (d, q', e, w) is in $\delta(q, a, Z)$ and $w \neq Z$, then $e = 0$.
(7) q_0 is in K (the *start* state).
(8) $F \subseteq K$ (the set of *final* states).

The formalism[3] (d, q', e, Z) is in $\delta(q, a, Z)$ means the following: Suppose the one-way sa A is in state q, reading a on the input tape and Z on the stack. Then A may[4]
(i) go to state q';
(ii) move right on the input tape if $d = 1$ and remain stationary if $d = 0$;
(iii) move left on the stack if $e = -1$, move right if $e = 1$, and stay stationary if $e = 0$.

The formalism $(d, q', 0, w)$ is in $\delta(q, a, Z)$ means that if A is in configuration (q, a, Z) of $K \times (\Sigma \cup \{\cent, \$\}) \times \Gamma$, then A may
(i) move on the input as specified by d;
(ii) go to state q';
(iii) write w in place of Z.

(Later in this paper the "write" command is restricted so that it is applicable only when A scans the rightmost stack symbol.)

The symbolism introduced so far allows only a single move of the device. The formalism is now expanded to allow discussion of sequences of basic operations of the device.

Definition. An *instantaneous description* (abbreviated ID) of a one-way sa is any element of $K \times (\Sigma \cup \{\cent, \$\})^* \times (\Gamma \cup \{\upharpoonleft\})^*$, where[5] $\upharpoonleft$ is a symbol not in Γ.

The ID $(q, a_i \cdots a_k, Z_1 \cdots Z_j \upharpoonleft Z_{j+1} \cdots Z_l)$ denotes the fact that A is in state q, reading input a_i, with $Z_1 \cdots Z_l$ on the stack and A scanning Z_j. $\upharpoonleft$ is referred to as the "stack pointer."

Notation. Given a one-way sa $A = (K, \Sigma, \cent, \$, \Gamma, \delta, q_0, Z_0, Z_0, F)$, the relation $\vdash$ between ID's is defined as follows: Let $i, k, l \geq 1$; $a_1, \cdots, a_k$ in $\Sigma \cup \{\cent, \$\}$; $Z_1, \cdots, Z_l$ in Γ; y in Γ^*; and Z in Γ.

(1) If (d, q', e, Z_j) is in $\delta(q, a_i, Z_j)$, where $1 \leq i \leq k$ and $1 \leq j \leq l$, and (i) $e \geq 0$ if $j = 1$, and (ii) $e \leq 0$ if $j = l$; then

$$(q, a_i \cdots a_k, Z_1 \cdots Z_j \upharpoonleft \cdots Z_l) \vdash (q', a_{i+d} \cdots a_k, Z_1 \cdots Z_{j+e} \upharpoonleft \cdots Z_l).^6$$

(2) If $(d, q', 0, w)$ is in $\delta(q, a_i, Z)$ for some $1 \leq i \leq k$, then

$$(q, a_i \cdots a_k, yZ\upharpoonleft) \vdash (q', a_{i+d} \cdots a_k, yw\upharpoonleft).$$

[1] For sets of words X and Y, the (*complex*) *product* of X and Y, written XY, is the set $\{xy \mid x \text{ in } X,\ y \text{ in } Y\}$, where xy is the concatenation of x and y. Let $X^0 = \{\epsilon\}$, where ϵ is the empty word. For $i \geq 0$, let $X^{i+1} = X^i X$ and $X^* = \bigcup_{i=0}^{\infty} X^i$. Thus Σ^* is the free semigroup with identity generated by Σ.

[2] Because of a difference in point of view between [5] and the present paper, a one-way sa now means a nondeterministic device. In [5], a one-way sa is taken as a deterministic device.

[3] To avoid long strings of quantifiers, unless stated otherwise q and q' are in K, a is in $\Sigma \cup \{\cent, \$\}$, Z is in Γ, d is in $\{0, 1\}$, and e is in $\{-1, 0, 1\}$.

[4] A is nondeterministic and thus may have other choices.

[5] As in [5], for technical convenience many elements are called instantaneous descriptions even though they do not correspond to actual configurations of a one-way sa, e.g., any ID with two occurrences of $\upharpoonleft$.

[6] $a_{k+1} \cdots a_k$ is to be interpreted as ϵ.

The relation $\vdash$ completely describes the atomic acts of a one-way sa. Condition (1) allows one-way motion on the input tape and two-way reading of the stack. Restrictions (i) and (ii) prevent A from going off either end of the stack. The condition $i + d = k + 1$ is allowed, and it means that the automaton has left the right end of the input tape. Condition (2) permits the device to write on the right end of the stack, where writing ϵ is actually erasing.

Note that A is unable to write in the interior of the stack, and it "blocks" if the stack becomes empty.

The notation for describing a sequence of movements of A is now given.

Definition. For each x, y in $(\Sigma \cup \{¢, \$\})^*$, and w_1, w_2, w_1', w_2' in Γ^*, let

$$(q, xy, w_1 \uparrow w_2) \vdash^* (q', y, w_1' \uparrow w_2')$$

if there exist $k \geq 0$, f_i, g_i, and h_i for $0 \leq i \leq k$ such that $f_0 = q$, $g_0 = xy$, $h_0 = w_1 \uparrow w_2$, $f_k = q'$, $g_k = y$, $h_k = w_1' \uparrow w_2'$, and $(f_i, g_i, h_i) \vdash (f_{i+1}, g_{i+1}, h_{i+1})$ for $0 \leq i < k$.

The final states in a one-way sa are used to "accept" a set of words by the following procedure.

Definition. A word x in Σ^* is *accepted* by a one-way sa A if

$$(q_0, ¢x\$, Z_0 \uparrow) \vdash^* (q, \epsilon, w_1 \uparrow w_2)$$

for some q in F and some w_1, w_2 in Γ^*. The set of words accepted by A, denoted by $T(A)$, is called a *one-way sa language* (abbreviated *language*).

We now specialize the model to be "deterministic" in nature, i.e., for each (q, a, Z) in $K \times (\Sigma \cup \{¢, \$\}) \times \Gamma$, there is to be one and only one "next move" which is possible.

Definition. A *deterministic one-way sa* is a one-way sa $A = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$ with the following properties:

(α) $\#(\delta(q, a, Z)) = 1$,[7] for each (q, a, Z) in $K \times (\Sigma \cup \{¢, \$\}) \times \Gamma$.

(β) If[8] $\delta(q, a, Z_0) = (d, q', e, y)$, then $y = Z_0 w$ for some w in Γ^*.

Condition (β) prevents the stack from being emptied. (The leftmost stack symbol is always Z_0).

Definiton. A language L is called a *determinististic language* (abbreviated D-language), if there is a deterministic one-way sa A such that $T(A) = L$.

An important specialization of a one-way sa is now noted. By restricting the stack to function as a pushdown store, i.e., to be read only at the right end, we obtain a pushdown automaton. Specifically, a *pushdown automaton* (pda) is a one-way sa with the restriction that (d, q', e, w) in $\delta(q, a, Z)$ implies $e = 0$.[9]

To obtain the family of deterministic pushdown automata, we need only start from a one-way deterministic sa. Specifically, a *deterministic pushdown automaton* is a one-way sa in which $\delta(q, a, Z) = (d, q', e, w)$ implies $e = 0$.[9]

In this paper, we are concerned with one-way sa and languages. We were led to

[7] For any set E, $\#(E)$ is the number of elements in E.

[8] We write $\delta(q, a, Z) = (d, q', e, y)$ instead of $\delta(q, a, Z) = \{(d, q', e, y)\}$.

[9] The comparison is not quite obvious, since a one-way sa has endmarkers while pushdown automata customarily do not. Furthermore, there is a slight distinction between "ϵ-moves" in pushdown automata and moves in one-way sa in which $d = 0$. It is not difficult to prove that the families of languages accepted by these two types of device are identical.

these concepts by specializing to the one-way case the stack automata and languages accepted by stack automata as discussed in [5]. A practical reason for studying one-way sa is that more of ALGOL can be recognized by these devices than by pushdown automata. (As an example of this, consider the role of declaration statements. As noted in [5], a (two-way) stack automaton can search symbol tables and thus can recognize declaration statements. A (nondeterministic) one-way sa also can search symbol tables as follows: It guesses that it is reading an identifier on the input tape, it has its stack pointer move left into the stack, and then it guesses that it has found the proper entry in the proper symbol table. Bad guesses lead to unfruitful computations of the sa.) As is well known, ALGOL with the constraint that identifiers must be declared cannot be entirely recognized by a pushdown automaton. Thus sets accepted by one-way sa may be better approximations to currently used programming languages than sets recognized by pushdown automata. Unfortunately, since ALGOL does not require declaration of all identifiers before their use, it appears that one-way sa cannot accept all ALGOL programs.

2. *Closure Properties of Languages*

There are a number of operations which have proved to be important in the theory of context-free languages.[10] In this section, many of these operations are shown to preserve languages.

For technical reasons it is convenient to introduce two variations of one-way sa, namely, the "one-way sa without left endmarker" and the "one-way sa without endmarkers." The latter device is useful in a number of proofs about languages.

Definition. A *one-way sa without left endmarker* is an 8-tuple $A = (K, \Sigma, \$, \Gamma, \delta, q_0, Z_0, F)$, where $K, \Sigma, \$, \Gamma, q_0, Z_0, F$ are the same as in a one-way sa, and δ is a function from $K \times (\Sigma \cup \{\$\}) \times \Gamma$ into the set of finite subsets of $\{0, 1\} \times K \times \{-1, 0, 1\} \times \Gamma^*$ satisfying $(*)$ in the definition of a one-way sa.

The definitions of ID, $\vdash$, $\vdash^*$, and acceptance for a one-way sa without left endmarker are the same (with obvious modifications) as for a one-way sa. For example:

Definition. A word w in Σ^* is *accepted* by a one-way sa without left endmarker $A = (K, \Sigma, \$, \Gamma, \delta, q_0, Z_0, F)$ if $(q_0, w\$, Z_0\uparrow) \vdash^* (q, \epsilon, y_1\uparrow y_2)$ for some q in F, y_1, y_2 in Γ^*.

A "one-way sa $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ without endmarkers" is defined analogously. The definitions for ID, $\vdash$, $\vdash^*$ and acceptance are also defined analogously.

We shall show (Lemma 2.3) that the sets of words accepted by the two variations of one-way sa coincide with the languages. This permits us to simplify certain arguments.

LEMMA 2.1. *If $L = T(A)$ for some one-way sa A, then $L = T(B)$ for some one-way sa B without left endmarker.*

PROOF. Let $L = T(A)$ for the one-way sa $A = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$. Let $B = (K_B, \Sigma, \$, \Gamma, \delta_B, \bar{q}_0, Z_0, F_B)$, where $K_B = K \cup \{\bar{q} \mid q \text{ in } K\}$, each $\bar{q}$ a new symbol, $F_B = F$ if ϵ is not in $T(A)$, $F_B = F \cup \{\bar{q}_0\}$ if ϵ is in $T(A)$, and δ_B is defined as follows for each (q, a, b, Z) in $K \times \Sigma \times (\Sigma \cup \{\$\}) \times \Gamma$:

(1) $\delta_B(\bar{q}_0, \$, Z) = \{(1, \bar{q}_0, 0, Z_0)\}$.

[10] Context-free languages may be characterized as those sets accepted by pushdown automata.

(2) $\delta_B(\bar{q}, a, Z) = \{(0, \bar{q}', e, y) \mid (0, q', e, y) \text{ in } \delta(q, ¢, Z)\}$
$\cup \{(0, q', e, y) \mid (1, q', e, y) \text{ in } \delta(q, ¢, Z)\}$.

(3) $\delta_B(q, b, Z) = \delta(q, b, Z)$.[11]

Intuitively, B operates as follows. Rule (1) ensures that ϵ is in $T(B)$ if and only if ϵ is in $T(A)$. Given a non-ϵ input word w, B first simulates the action of ¢ in A (rule (2)). When A moves to the right of ¢, B records this by changing to states q and then mimicking A (rules (2) and (3)). Formally, for a in Σ and w in Σ^*,

$$\begin{aligned}(q_0, ¢aw\$, Z_0\rceil) &\vdash_A^* (q_1, ¢aw\$, y_1\rceil y_1')\\ &\vdash_A (q_2, aw\$, y_2\rceil y_2')\\ &\vdash_A^* (q_3, \epsilon, y_3, \rceil y_3')\end{aligned}$$

if and only if

$$\begin{aligned}(\bar{q}_0, aw\$, Z_0\rceil) &\vdash_B^* (\bar{q}_1, aw\$, y_1\rceil y_1')\\ &\vdash_B (q_2, aw\$, y_2\rceil y_2')\\ &\vdash_B^* (q_3, \epsilon, y_3\rceil y_3').\end{aligned}$$

Thus $T(A) = T(B)$.

LEMMA 2.2. *If $L = T(A)$ for some one-way sa A without left endmarker, then $L = T(B)$ for some one-way sa B without endmarkers.*

PROOF. Let $A = (K, \Sigma, \$, \Gamma, \delta, q_0, Z_0, F)$. Let $B = (K_B, \Sigma, \Gamma, \delta_B, \bar{q}_0, Z_0, F_B)$, where $\bar{q}_0$ and $\bar{q}_1$ are new symbols, $K_B = (K \times \Sigma) \cup K \cup \{\bar{q}_0, \bar{q}_1\}$, $F_B = \{\bar{q}_0, \bar{q}_1\}$ if ϵ is in $T(A)$, $F_B = \{\bar{q}_1\}$ if ϵ is not in $T(A)$, and δ_B is defined as follows:

(1) $\delta_B(\bar{q}_0, a, Z) = \delta(q_0, a, Z) \cup \{(0, (q', a), e, y) \mid (1, q', e, y) \text{ in } \delta(q_0, a, Z)\}$.
(2) $\delta_B(q, a, Z) = \delta(q, a, Z) \cup \{(0, (q', a), e, y) \mid (1, q', e, y) \text{ in } \delta(q, a, Z)\}$.
(3) $\delta_B((q, a), a, Z) = \{(0, (q', a), e, y) \mid (0, q', e, y) \text{ in } \delta(q, \$, Z)\}$
$\cup \{(1, \bar{q}_1, e, y) \mid (1, q', e, y) \text{ in } \delta(q, \$, Z) \text{ and } q' \text{ in } F\}$.

Intuitively, B starts out by imitating A. If the next input in A is to be \$, then B goes to a state (q', a) and simulates the movement of A under \$ while B still scans a. Formally, for w in Σ^* and a in Σ,

$$\begin{aligned}(q_0, wa\$, Z_0\rceil) &\vdash_A^* (q_1, a\$, y_1\rceil y_1') \vdash_A (q_2, \$, y_2\rceil y_2')\\ &\vdash_A^* (q_3, \epsilon, y_3\rceil y_3') \text{ for some } q_3 \text{ in } F\end{aligned}$$

if and only if

$$(\bar{q}_0, wa, Z_0\rceil) \vdash_B^* (q_1, a, y_1\rceil y_1') \vdash_B ((q_2, a), a, y_2\rceil y_2') \vdash_B^* (\bar{q}_1, \epsilon, y_3\rceil y_3').$$

Furthermore, ϵ is in $T(A)$ if and only if ϵ is in $T(B)$. Thus $T(A) = T(B)$.

We are now able to show that the families of sets accepted by the various kinds of one-way sa coincide.

LEMMA 2.3. *The following three statements are equivalent for a set $L \subseteq \Sigma^*$:*
(1) $L = T(A)$ for some one-way sa A.
(2) $L = T(A')$ for some one-way sa A' without left endmarker.
(3) $L = T(A'')$ for some one-way sa A'' without endmarkers.

[11] $\delta(q, a, Z)$ is always to be ϕ unless otherwise stated.

PROOF. By Lemmas 2.1 and 2.2, (1) implies (2) and (2) implies (3). It thus suffices to show that (3) implies (1). Therefore suppose that $A'' = (K'', \Sigma, \Gamma, \delta'', q_0, Z_0, F)$ is a one-way sa without endmarkers. Let $A = (K'', \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$, where $\delta(q, a, Z) = \bar{\delta}(q, a, Z)$ for each (q, a, Z) in $K'' \times \Sigma \times \Gamma$ and $\delta(q, ¢, Z) = \delta(q, \$, Z) = \{(1, q, 0, Z)\}$. Clearly $T(A) = T(A'')$.

We summarize some obvious properties about one-way sa and one-way sa without endmarkers in the following two lemmas. These lemmas assert that any language may be accepted:

(a) by some one-way sa (without endmarkers) in which the initial state occurs only at the beginning of the computation;

(b) by some one-way sa (without endmarkers) in which the longest "write instruction" has length at most two;

(c) by some one-way sa (without endmarkers) with a leftmost symbol on the stack which (i) is never erased, and (ii) appears only as the leftmost symbol on the stack;

(d) by some one-way sa A (without endmarkers) with the following properties: (i) A has a unique final state, and (ii) A accepts a word if and only if A ends in the final state with exactly Z_0 on the stack;

(e) by some one-way sa (without endmarkers) with properties (a)–(d).

The above properties are formalized for one-way sa in the following lemma.

LEMMA 2.4. *Let $A = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$ be a one-way sa. Then:*

(*a*) *There exists a one-way sa $A_1 = (K_1, \Sigma, ¢, \$, \Gamma, \delta_1, \bar{q}_0, Z_0, F)$ such that $T(A_1) = T(A)$ and for each q in K_1, w_1w_2 in $¢\Sigma^*\$$, $w_1 \neq \epsilon$,*

$$(\bar{q}_0, w_1w_2, Z_0\uparrow) \vdash^*_{A_1} (q, w_2, y_1\uparrow y_2)$$

implies $q \neq q_0$.

(*b*) *There exists a one-way sa $A_2 = (K_2, \Sigma, ¢, \$, \Gamma, \delta_2, q_0, Z_0, F)$ such that $T(A_2) = T(A)$ and (d, q', e, y) in $\delta_2(q, a, Z)$ implies $|y| \leq 2$.*[12]

(*c*) *There exists a one-way sa $A_3 = (K_3, \Sigma, ¢, \$, \Gamma_3, \delta_3, \bar{q}_0, \bar{Z}_0, F)$ such that $T(A_3) = T(A)$ and*

$$(\bar{q}_0, w_1w_2, \bar{Z}_0\uparrow) \vdash^*_{A_3} (q, w_2, y_1\uparrow y_2)$$

implies y_1y_2 is in $\bar{Z}_0(\Gamma - \{\bar{Z}_0\})^*$.

(*d*) *There exists a one-way sa $A_4 = (K_4, \Sigma, ¢, \$, \Gamma_4, \delta_4, \bar{q}_0, \bar{Z}_0, \{f\})$ such that* (i) *$T(A_4) = T(A)$,* (ii) *w is in $T(A_4)$ if and only if*

$$(q_0, ¢w\$, \bar{Z}_0\uparrow) \vdash^*_{A_4} (f, \epsilon, \bar{Z}_0\uparrow),$$

(iii) *$\delta_4(f, a, Z) = \phi$ for all a and Z, and* (iv) *(d, f, e, w) in $\delta_4(q, a, Z)$ implies $d = 1$, $e = 0$,* and *$w = Z = \bar{Z}_0$.*

(*e*) *There exists a one-way sa $A_5 = (K_5, \Sigma, ¢, \$, \Gamma_5, \delta_5, \bar{q}_0, \bar{Z}_0, F_5)$ such that $T(A_5) = T(A)$ and A_5 satisfies (a)–(d).*

PROOF. We give the construction of each one-way sa. The proof that each device has the desired properties is clear and is omitted.

(a) Let $K_1 = K \cup \{\bar{q}_0\}$, where $\bar{q}_0$ is a new symbol. Let $\delta_1(q, a, Z) = \delta(q, a, Z)$ and $\delta_1(\bar{q}_0, a, Z_0) = \{(0, q_0, 0, Z_0)\}$ for all q in K, a in $\Sigma \cup \{¢, \$\}$, and Z in Γ.

[12] For each word y, $|y|$ denotes the length of y.

(b) For each $(d, q', e, Z_1 \cdots Z_r)$ in $\delta(q, a, Z)$, (i) if $r \leq 2$, let $(d, q', e, Z_1 \cdots Z_r)$ be in $\delta_2(q, a, Z)$; (ii) if $r > 2$, let q_i, $1 \leq i \leq r - 2$, be new symbols and let $(0, q_1, 0, Z_1Z_2)$ be in $\delta_2(q, a, Z)$, $(0, q_{i+1}, 0, Z_{i+1}Z_{i+2})$ be in $\delta_2(q_i, a, Z_{i+1})(1 \leq i \leq r - 3)$, and $(d, q', e, Z_{r-1}Z_r)$ be in $\delta_2(q_{r-2}, a, Z_{r-1})$.

Let $K_2 = K \cup \{q_i \mid q_i \text{ defined above}\}$.

(c) Let $K_3 = K \cup \{\bar{q}_0\}$ and $\Gamma_3 = \Gamma \cup \{\bar{Z}_0\}$, where $\bar{q}_0$ and $\bar{Z}_0$ are new symbols. Let $\delta_3(\bar{q}_0, a, Z_0) = \{(0, q_0, 0, \bar{Z}_0Z_0)\}$ and $\delta_3(q, a, Z) = \delta(q, a, Z)$ for each q in K, a in $\Sigma \cup \{¢, \$\}$, and Z in Γ.

(d) Without loss of generality we may assume that A satisfies (c). Let $K_4 = K_3 \cup \{\bar{q}, f\}$, and $\Gamma_4 = \Gamma_3$, where $\bar{q}$ and f are new symbols. For each q in K and a in $\Sigma \cup \{¢, \$\}$, let (i) $\delta_4(q, a, Z) = \delta_3(q, a, Z) \cup \{(0, \bar{q}, e, y) \mid (1, q', e, y) \text{ in } \delta_3(q, a, Z)$ and q' is in $F\}$ for each Z in Γ; (ii) $\delta_4(\bar{q}, a, Z) = \{(0, \bar{q}, 1, Z), (0, \bar{q}, 0, \epsilon)\}$ for each Z in $\Gamma - \{\bar{Z}_0\}$; (iii) $\delta_4(\bar{q}, a, \bar{Z}_0) = \{(1, f, 0, \bar{Z}_0)\}$.

(e) The proof of (e) follows from the fact that each of the constructions in (a)–(d) may be carried out without destroying any of the other properties.

The analogue to Lemma 2.4 for one-way sa without endmarkers is now given.

LEMMA 2.5. *Let $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a one-way sa without endmarkers. Then:*

(*a*) *There exists a one-way sa $A_1 = (K_1, \Sigma, \Gamma, \delta_1, \bar{q}_0, Z_0, F_1)$ without endmarkers such that $T(A_1) = T(A)$ and for each q in K_1, $w_1 \neq \epsilon$, w_1w_2 in Σ^*,*

$$(\bar{q}_0, w_1w_2, Z_0\uparrow) \vdash^*_{A_1} (q, w_2, y_1\uparrow y_2)$$

implies $q \neq \bar{q}_0$.

(*b*) *There exists a one-way sa $A_2 = (K_2, \Sigma, \Gamma, \delta_2, q_0, Z_0, F)$ without endmarkers such that $T(A_2) = T(A)$ and (d, q', e, y) in $\delta_2(q, a, Z)$ implies $|y| \leq 2$.*

(*c*) *There exists a one-way sa $A_3 = (K_3, \Sigma, \Gamma_3, \delta_3, \bar{q}_0, \bar{Z}_0, F)$ without endmarkers such that $T(A_3) = T(A)$ and*

$$(\bar{q}_0, w_1w_2, \bar{Z}_0\uparrow) \vdash^*_{A_3} (q, w_2, y_1\uparrow y_2)$$

implies y_1y_2 is in $\bar{Z}_0(\Gamma - \{\bar{Z}_0\})^*$.

(*d*) *There exists a one-way sa $A_4 = (K_4, \Sigma, \Gamma_4, \delta_4, \bar{q}_0, \bar{Z}_0, \{f\})$ without endmarkers such that* (i) *$T(A_4) = T(A) - \{\epsilon\}$,* (ii) *$w$ is in $T(A_4)$ if and only if*

$$(\bar{q}_0, w, \bar{Z}_0\uparrow) \vdash^*_{A_4} (f, \epsilon, \bar{Z}_0\uparrow),$$

and (iii) *$\delta_4(f, a, Z) = \phi$ for all a and Z.*[13]

(*e*) *There exists a one-way sa $A_5 = (K_5, \Sigma, \Gamma_5, \delta_5, \bar{q}_0, \bar{Z}_0, F_5)$ without endmarkers such that $T(A_5) = T(A) - \{\epsilon\}$ and A_5 satisfies (a)–(d).*[13]

PROOF. The proof is a trivial modification of Lemma 2.4 and is omitted.

The first result on operations concerns the intersection of a language and a regular set.[14]

[13] Note that ϵ cannot be in $T(A_4)$ since $(\bar{q}_0, \epsilon, \bar{Z}_0\uparrow) \vdash^*(q, \epsilon, y_1\uparrow y_2)$ only if $q = \bar{q}_0$, $y_1 = \bar{Z}_0$, and $y_2 = \epsilon$. Similarly ϵ cannot be in $T(A_5)$.

[14] A *finite-state automaton* (abbreviated fsa) is a 5-tuple $A = (K, \Sigma, \delta, p_0, F)$, where K and Σ are finite nonempty sets (of *states* and *inputs* respectively), δ is a function from $K \times \Sigma$ into K, p_0 is in K, and $F \subseteq K$. The function δ is extended to $K \times \Sigma^*$ by defining $\delta(q, \epsilon) = q$ and $\delta(q, xa) = \delta[\delta(q, x), a]$ for each (q, x, a) in $K \times \Sigma^* \times \Sigma$. A set $R \subseteq \Sigma^*$ is said to be *regular* if there exists an fsa $A = (K, \Sigma, \delta, p_0, F)$ such that $R = T(A)$, where $T(A) = \{x \mid \delta(p_0, x) \text{ in } F\}$.

THEOREM 2.1. *If L is a language and R is a regular set, then $L \cap R$ is a language.*

PROOF. Let $L = T(A)$, where $A = (K_1, \Sigma, ¢, \$, \Gamma, \delta_1, q_0, Z_0, F_1)$ is a one-way sa. Let $R = T(B)$, where $B = (K_2, \Sigma, \delta_2, p_0, F_2)$ is an fsa. Extend δ_2 to a mapping of $K_2 \times (\Sigma \cup \{¢, \$\})$ by the condition $\delta_2(q, ¢) = \delta_2(q, \$) = q$ for each q in K_2. Define $C = (K_1 \times K_2, \Sigma, ¢, \$, \Gamma, \delta, (q_0, p_0), Z_0, F_1 \times F_2)$, where δ is defined as follows: For each (q, a, Z) in $K_1 \times (\Sigma \cup \{¢, \$\}) \times \Gamma$ and each p in K_2,

$$\delta((q, p), a, Z) = \{(0, (q', p), e, y) | (0, q', e, y) \text{ in } \delta_1(q, a, Z)\} \cup \{(1, (q', \delta_2(p, a)), e, y) | (1, q', e, y) \text{ in } \delta_1(q, a, Z)\}.$$

Clearly $T(C) = T(A) \cap T(B) = L \cap R$.[15]

COROLLARY 1. *$L - R$ is a language for each language L and each regular set R.*

PROOF. Since R is regular, $\Sigma^* - R$ is regular [6]. Then $L - R = L \cap (\Sigma^* - R)$ is a language by the theorem.

COROLLARY 2. *If L is a D-language and R is a regular set, then $L \cap R$ and $L - R$ are D-languages.*

PROOF. Let A and C be as in the proof of Theorem 2.1. If L is a D-language, then A is a deterministic one-way sa. Clearly C is deterministic, so that $L \cap R$ is a D-language. Since the complement of a regular set is regular [6], $L - R = L \cap (\Sigma^* - R)$ is also a D-language.

We now consider the basic operations of union, product, and $*$. First, however, we prove a result involving context-free languages.

THEOREM 2.2. *Let $L \subseteq \Sigma^*$ be a context-free language. For each a in Σ, let Σ_a be a finite set and $L_a \subseteq \Sigma_a{}^*$ be a language. Then*

$$H = \{x_1 \cdots x_n \,|\, n \geq 0, \quad a_1, \cdots, a_n \text{ in } \Sigma, \quad x_j \text{ in } L_{a_j}, \quad a_1 \cdots a_n \text{ in } L\}$$

is a language.

PROOF. Let $L = T(A)$, where $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is a pda. Without loss of generality, we may assume that there is no element a in Σ such that ϵ is in L_a. (For otherwise, let $\tau(a) = \{a\}$ if ϵ is not in L_a and $\tau(a) = \{\epsilon, a\}$ if a is in L_a. Then $\tau(L) = \bigcup_{a_1 \cdots a_k \text{ in } L} \tau(a_1) \cdots \tau(a_k)$ is context-free and

$$H = \{x_1 \cdots x_n \,|\, n \geq 0, \quad a_1, \cdots, a_n \text{ in } \Sigma, \quad x_j \text{ in } L_{a_j} - \{\epsilon\}, \quad a_1 \cdots a_n \text{ in } \tau(L)\}.)$$

Then, for each a in Σ, $L_a = T(A_a)$, where $A_a = (K_a, \Sigma_a, \Gamma_a, \delta_a, q_{0a}, Z_{0a}, \{f_a\})$ satisfies Lemma 2.5(e), and all the K_a, K, Γ_a, Γ are pairwise disjoint. Let $\bar{q}_0$ be a symbol not in $K \cup \bigcup_{a \text{ in } \Sigma} (K_a \times K) \cup (\Sigma \times K)$. Let $\bar{A}$ be the one-way sa without left endmarker $(\bar{K}, \bar{\Sigma}, \$, \bar{\Gamma}, \bar{\delta}, \bar{q}_0, Z_0, \bar{F})$, where $\bar{K} = \{\bar{q}_0\} \cup K \cup \bigcup_a (K_a \times K) \cup (\Sigma \times K)$, $\bar{\Gamma} = \Gamma \cup \bigcup_a \Gamma_a$, $\bar{\Sigma} = \bigcup_a \Sigma_a$, $\bar{F} = F \cup \{\bar{q}_0\}$ if ϵ is in L and $\bar{F} = F$ if ϵ is not in L, and $\bar{\delta}$ is defined as follows for (a, b, q, Z) in $\Sigma \times \bar{\Sigma} \times K \times \Gamma$:

(i) (a) $\bar{\delta}(\bar{q}_0, b, Z_0) = \{(0, (q_{0c}, q_0), 0, Z_0 Z_{0c}) |\, c \text{ in } \Sigma\}$.

(b) $\bar{\delta}(\bar{q}_0, \$, Z_0) = \{(1, \bar{q}_0, 0, Z_0)\}$.

(ii) For Z_a in Γ_a and q_a in K_a,

$$\bar{\delta}((q_a, q), b, Z_a) = \{(d, (q_a', q), e, w) | (d, q_a', e, w) \text{ in } \delta_a(q_a, b, Z_a)\} \cup \{(1, (a, q), 0, \epsilon) | (1, f_a, 0, Z_a) \text{ in } \delta_a(q_a, b, Z_a)\}.$$

[15] The proof could be simplified slightly by considering one-way sa without endmarkers. The present form is given in order to obtain Corollary 2 to Theorem 2.1.

(iii) $\bar{\delta}((a, q), b, Z) = \{(0, (a, q'), 0, w) | (0, q', 0, w) \text{ in } \delta(q, a, Z)\}$
$\cup \{(0, (q_{0c}, q'), 0, wZ_{0c}) | (1, q', 0, w) \text{ in } \delta(q, a, Z), c \text{ in } \Sigma\}$.
(iv) $\bar{\delta}((a, q), \$, Z) = \{(0, (a, q'), 0, w) | (0, q', 0, w) \text{ in } \delta(q, a, Z)\}$
$\cup \{(1, q', 0, w) | (1, q', 0, w) \text{ in } \delta(q, a, Z)\}$.

Intuitively, $\bar{A}$ starts by (i), simulates A_a, searching for w in L_a, by (ii), and simulates A on a by (iii) and (iv). Formally, the construction does the following. By (i-b), ϵ is in $T(\bar{A})$ if and only if ϵ is in H. By (i-a),

$$(\bar{q}_0, x\$, Z_0\uparrow) \vdash_{\bar{A}} ((q_{0a}, q_0), x\$, Z_0Z_{0a}\uparrow)$$

for all a in Σ and $x \neq \epsilon$ in $\bar{\Sigma}$. By (ii), when $x \neq \epsilon$,

$$((q_{0a}, q), xy\$, uZ_{0a}\uparrow) \vdash_{\bar{A}}^{*} ((a, q), y\$, u\uparrow)$$

if and only if x is in L_a. By (iii) and (iv), when $x \neq \epsilon$,

$$((a, q), x\$, u\uparrow) \vdash_{\bar{A}(\text{iii})}^{*} ((q_{0c}, q'), x\$, vZ_{0c}\uparrow)$$

if and only if

$$(q, a, u\uparrow) \vdash_{\bar{A}}^{*} (q', \epsilon, v\uparrow)$$

if and only if

$$((a, q), \$, u\uparrow) \vdash_{\bar{A}(\text{iv})}^{*} (q', \epsilon, v\uparrow).$$

For $k \geq 1$ and each x_i in $\bar{\Sigma}^*$,

$$\begin{aligned}(\bar{q}_0, x_1 \cdots x_k\$, Z_0\uparrow) &\vdash_{\bar{A}}^{*} ((a_1, q_0), x_2 \cdots x_k\$, Z_0\uparrow)\\ &\vdash_{\bar{A}}^{*} ((a_2, q_1), x_3 \cdots x_k\$, u_1\uparrow)\\ &\cdots\\ &\vdash_{\bar{A}}^{*} ((a_k, q_{k-1}), \$, u_{k-1}\uparrow)\\ &\vdash_{\bar{A}}^{*} (q_k, \epsilon, u_k\uparrow),\end{aligned}$$

where the states (a_k, q_{k-1}) are precisely those states in $\Sigma \times K$ which follow steps of type (ii), if and only if

$$\begin{aligned}(q_0, a_1 \cdots a_k\$, Z_0\uparrow) &\vdash_{A}^{*} (q_1, a_2 \cdots a_k\$, u_1\uparrow)\\ &\vdash_{A}^{*} (q_2, a_3 \cdots a_k\$, u_2\uparrow)\\ &\cdots\\ &\vdash_{A}^{*} (q_{k-1}, a_k\$, u_{k-1}\uparrow)\\ &\vdash_{A}^{*} (q_k, \epsilon, u_k\uparrow)\end{aligned}$$

and x_i is in L_{a_i} for each i, $1 \leq i \leq k$. Thus $T(\bar{A}) = H$, so that H is a language.

Since $\{a\} \cup \{b\}$, $\{ab\}$, and a^* are context-free languages for $\Sigma = \{a, b\}$, we obtain as a corollary:

THEOREM 2.3. *The family of languages is closed under union, product, and* $*$.

We now introduce a transformation device which preserves languages.

Definition. A *sequential transducer* is a 5-tuple $S = (K, \Sigma, \Delta, H, s_0)$ where

(i) K, Σ, and Δ are finite nonempty sets (of *states*, *inputs*, and *outputs* respectively);

(ii) s_0 is in K (the *start* state);

(iii) H is a finite subset of $K \times \Sigma^* \times \Delta^* \times K$.

The sequential transducer transforms words as follows:

Definition. Let $S = (K, \Sigma, \Delta, H, s_0)$ be a sequential transducer. For each u in Σ^*, let $S(u)$ be the set of words v with the property that there exist $k \geq 1$ and words $u_1, \cdots, u_k$ in Σ^*, $v_1, \cdots, v_k$ in Δ^*, and $s_1, \cdots, s_k$ in K such that $u = u_1 \cdots u_k$, $v = v_1 \cdots v_k$, and $(s_i, u_{i+1}, v_{i+1}, s_{i+1})$ is in H for each $0 \leq i < k$. For each $U \subseteq \Sigma^*$, let $S(U) = \bigcup_{u \text{ in } U} S(u)$.

Given a sequential transducer $S = (K, \Sigma, \Delta, H, s_0)$, there exists a sequential transducer $T = (\bar{K}, \Sigma, \Delta, \bar{H}, \bar{s}_0)$ such that (i) $S(U) = T(U)$ for all $U \subseteq \Sigma^*$ and (ii) (s, u, v, s') in $\bar{H}$ implies $s' \neq \bar{s}_0$. (For let $\bar{s}_0$ be a new symbol, $\bar{K} = K \cup \{\bar{s}_0\}$, and $\bar{H} = H \cup \{(\bar{s}_0, u, v, s) | (s_0, u, v, s) \text{ in } H\}$.)

THEOREM 2.4. *$S(L)$ is a language for each sequential transducer S and each language L.*

PROOF. Let L be a language. Then $L = T(A)$ for some one-way sa $A = (K_1, \Sigma, \Gamma, \delta_1, q_0, Z_0, F_1)$ without endmarkers. Let $S = (K_2, \Sigma, \Delta, H, s_0)$ be a sequential transducer. As noted above, there is no loss of generality in assuming that (s, u, v, s') in H implies $s' \neq s_0$. We may also assume that \$ is not in Δ and that A satisfies (c) of Lemma 2.5. Let

$$m = \max\{1, |u|, |v| \,|(s, u, v, s') \text{ in } H \text{ for some } s, s' \text{ in } K\}.$$

Let $\Sigma_m = \bigcup_{i=0}^{m} \Sigma^i$ and $\Delta_m = \bigcup_{i=0}^{m} \Delta^i$. Let $B = (\bar{K}, \Delta, \$, \Gamma, \delta, (q_0, s_0, \epsilon, \epsilon), Z_0, \bar{F})$ be the one-way sa without left endmarker with $\bar{K} = K_1 \times K_2 \times \Sigma_m \times \Delta_m$, $\bar{F} = F_1 \times (K_2 - \{s_0\}) \times \{\epsilon\} \times \{\epsilon\}$, and δ defined as follows:

(i) For each (q, a, Z) in $K_1 \times (\Delta \cup \{\$\}) \times \Gamma$ and each s in K_2,

$$\delta((q, s, \epsilon, \epsilon), a, Z) = \{(0, (q, s', u, v), 0, Z) | (s, u, v, s') \text{ in } H\}.$$

(ii) For each (q, s, a, c) in $K_1 \times K_2 \times \Sigma \times (\Delta \cup \{\$\})$ and for each (x, y) in $\Sigma_m \times \Delta_m$ such that $|x| < m$ and $|y| \leq m$,

$$\delta((q, s, ax, y), c, Z) = \{(0, (q', s, ax, y), e, w) | (0, q', e, w) \text{ in } \delta_1(q, a, Z)\} \cup \{(0, (q', s, x, y), e, w) | (1, q', e, w) \text{ in } \delta_1(q, a, Z)\}.$$

(iii) For each (q, s, b) in $K_1 \times K_2 \times \Delta$ and each y in Δ_m, with $|y| < m$,

$$\delta((q, s, \epsilon, by), b, Z) = \{(1, (q, s, \epsilon, y), 0, Z)\}.$$

(iv) For each (q, s, Z) in $K_1 \times K_2 \times \Gamma$,

$$\delta((q, s, \epsilon, \epsilon), \$, Z) = \{(1, (q, s, \epsilon, \epsilon), 0, Z)\}.$$

Intuitively, B operates as follows. B simulates the action at state s (i.e., the state of B is $(q, s, \epsilon, \epsilon)$) by guessing that S takes a word u into v (formally, (s, u, v, s') is in H, i.e., by (i)). By (ii), B simulates the action that A might take on u without advancing the input tape. In (iii), B advances the input tape over $S(u)$. The cycle is repeated until \$ is reached, at which time (iv) is applied. Formally, v is in $S(L)$ if and only if there exist $u_1, \cdots, u_k$, $v_1, \cdots, v_k$, $s_1, \cdots, s_k$, $q_1, \cdots, q_k$, $y_0, \cdots, y_k$, $y_0', \cdots, y_k'$ with the following properties: $y_0 = Z_0$; $y_0' = \epsilon$; q_k is in F_1; each $y_i y_i'$ is in $\Gamma^* - \{\epsilon\}$ (by (c) of Lemma 2.5);

$$(q_i, u_{i+1} \cdots u_k, y_i \uparrow y_i') \vdash_A^* (q_{i+1}, u_{i+2} \cdots u_k, y_{i+1} \uparrow y_{i+1}') \quad \text{for} \quad 0 \leq i < k$$

(so that $u_1 \cdots u_k$ is in L);

$$(s_i, u_{i+1}, v_{i+1}, s_{i+1}) \quad \text{is in} \quad H \quad \text{for} \quad 0 \leq i < k$$

(so that $v_1 \cdots v_k$ is in $S(u_1 \cdots u_k)$; and $v = v_1 \cdots v_k$. The latter occurs if and only if there exist $u_1, \cdots, u_k$, $v_1, \cdots, v_k$, $s_1, \cdots, s_k$, $q_1, \cdots, q_k$, $y_0, \cdots, y_k$, $y_0', \cdots, y_k'$ with the following properties: $y_0 = Z_0$; $y_0' = \epsilon$; q_k is in F_1; each $y_i y_i'$ is in $\Gamma^* - \{\epsilon\}$;

$$\begin{aligned}((q_i, s_i, \epsilon, \epsilon), v_{i+1} \cdots v_k\$, y_i \uparrow y_i') &\vdash_B ((q_i, s_{i+1}, u_{i+1}, v_{i+1}), v_{i+1} \cdots v_k\$, y_i \uparrow y_i') \\ &\vdash_B^* ((q_{i+1}, s_{i+1}, \epsilon, v_{i+1}), v_{i+1} \cdots v_k\$, y_{i+1} \uparrow y_{i+1}') \\ &\vdash_B^* ((q_{i+1}, s_{i+1}, \epsilon, \epsilon), v_{i+2} \cdots v_k\$, y_{i+1} \uparrow y_{i+1}')\end{aligned}$$

for $0 \leq i < k$; and

$$((q_k, s_k, \epsilon, \epsilon), \$, y_k \uparrow y_k') \vdash_B ((q_k, s_k, \epsilon, \epsilon), \epsilon, y_k \uparrow y_k'),^{16}$$

which, as is easily seen, occurs if and only if v is in $T(B)$.

A number of important special cases follow from the theorem on sequential transducers.

Definition. A *generalized sequential machine* (abbreviated gsm) is a 6-tuple $S = (K, \Sigma, \Delta, \delta, \lambda, q_0)$, where

(i) K, Σ, and Δ are finite nonempty sets (of *states, inputs,* and *outputs* respectively);

(ii) δ is a function from $K \times \Sigma$ into K (*next state* function);

(iii) λ is a function from $K \times \Sigma$ into Δ^* (*output* function);

(iv) q_0 is in K (*start* state).

δ and λ are extended to $K \times \Sigma^*$ by letting $\delta(q, \epsilon) = q$, $\lambda(q, \epsilon) = \epsilon$, $\delta(q, xa) = \delta[\delta(q, x), a]$, and $\lambda(q, xa) = \lambda(q, x)\lambda(\delta(q, x), a)$ for each (q, x, a) in $K \times \Sigma^* \times \Sigma$.

THEOREM 2.5. *If $S = (K, \Sigma, \Delta, \delta, \lambda, q_0)$ is a gsm and L is a language, then $S(L) = \{\lambda(q_0, x) \mid x \text{ in } L\}$ is a language.*

PROOF. Let S' be the sequential transducer $(K, \Sigma, \Delta, H, q_0)$, where $H = \{(q, a, \lambda(q, a), \delta(q, a)) \mid (q, a) \text{ in } K \times \Sigma\} \cup \{(q, \epsilon, \epsilon, q) \mid q \text{ in } K\}$. For each x in Σ^*, $S(x) = S'(x)$. The result then follows from Theorem 2.4.

Languages are also preserved under inverse gsm.

THEOREM 2.6. *If $S = (K, \Sigma, \Delta, \delta, \lambda, q_0)$ is a gsm and L is a language, then*

$$S^{-1}(L) = \{x \quad in \quad \Sigma^* \mid \lambda(q_0, x) \quad in \quad L\}$$

is also a language.

PROOF. Let S' be the sequential transducer $(K, \Sigma, \Delta, H, q_0)$, where

$$H = \{(q, \lambda(q, a), a, \delta(q, a)) \mid (q, a) \quad \text{in} \quad K \times \Sigma\} \cup \{(q, \epsilon, \epsilon, q) \mid q \quad \text{in} \quad K\}.$$

Clearly $S'(x) = S^{-1}(x)$ for each x in Σ^*. The result then follows from Theorem 2.4.

From Theorems 2.5 and 2.6 there immediately follows:

COROLLARY. *If h is a homomorphism*[17] *and L is a language, then both $h(L)$ and $h^{-1}(L)$ are languages.*

[16] Note that \$ is needed in order to serve as input in case $v_{k-j} \cdots v_k = \epsilon$ for some j.

[17] A *homomorphism* is a mapping h from Σ^* into Δ^* such that $h(\epsilon) = \epsilon$ and $h(a_1 \cdots a_k) = h(a_1) \cdots h(a_k)$, each a_i in Σ.

Another interesting operation which preserves languages is "quotient by a regular set."

THEOREM 2.7. *Let L be a language and R a regular set. Then*

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

and

$$R\backslash L = \{w \mid yw \text{ in } L \text{ for some } y \text{ in } R\}$$

are languages.

PROOF. Let $L \subseteq \Sigma^*$ be a language and c a new symbol. Let h be the homomorphism defined by $h(a) = a$ for each a in Σ, and $h(c) = \epsilon$. Consider the gsm $S = (\{q_0, q_1\}, \Sigma \cup \{c\}, \Delta, \delta, \lambda, q_0)$, where for each a in Σ, $\delta(q_0, a) = q_0$, $\delta(q_0, c) = q_1$, $\delta(q_1, a) = \delta(q_1, c) = q_1$, $\lambda(q_0, a) = a$, and $\lambda(q_0, c) = \lambda(q_1, a) = \lambda(q_1, c) = \epsilon$. Clearly $S(x) = x$ and $S(xcy) = x$ for each x in Σ^* and y in $(\Sigma \cup \{c\})^*$. Since $L' = h^{-1}(L) \cap \Sigma^* cR$ is a language, $L/R = S(L')$ is also a language.

The proof that $R\backslash L$ is a language is similar.

COROLLARY. *If L is a language, then so are*

$$Init(L) = \{u \mid uv \text{ in } L \text{ for some } v \text{ in } \Sigma^*\},$$

$$Fin(L) = \{v \mid uv \text{ in } L \text{ for some } u \text{ in } \Sigma^*\},$$

and

$$Sub(L) = \{v \mid uvw \text{ in } L \text{ for some } u, v \text{ in } \Sigma^*\}.$$

PROOF. Init $(L) = L/\Sigma^*$, Fin $(L) = \Sigma^*\backslash L$, and Sub $(L) =$ Fin (Init (L)).

The family of languages is closed under reversal.[18] To prove this, we "run" a one-way sa without endmarkers "backwards."

THEOREM 2.8. *If L is a language, then so is* L^R.

PROOF. Let $L = T(A)$, where $A = (K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ is a one-way sa without endmarkers. We may assume that ϵ is not in $T(A)$. (For otherwise $T(A) - \{\epsilon\}$ is a language, we can show that $(T(A) - \{\epsilon\})^R$ is a language, and $T(A)^R = (T(A) - \{\epsilon\})^R \cup \{\epsilon\}$ is a language.) We may also assume that A satisfies (e) of Lemma 2.5. Thus $F = \{f\}$ for some f in K. Let

$$B = (K_B, \Sigma, ¢, \$, \Gamma, \delta_B, (f, ¢, \epsilon, 0), Z_0, \{f_1\}),$$

where f_1 is a new symbol, $\Gamma_2 = \Gamma^0 \cup \Gamma^1 \cup \Gamma^2$,

$$K_B = (K \times (\Sigma \cup \{¢, \$\}) \times \Gamma_2 \times \{0, 1\}) \cup \{f_1\},$$

and δ_B is defined as follows:

(i) Suppose (d, q', e, y) is in $\delta(q, a, Z)$. For each b in $\Sigma \cup \{¢\}$ and Y in Γ:

(a) if y is in Γ, let $(d, (q, a, \epsilon, 0), -e, Z)$ be in $\delta_B((q', b, \epsilon, 0), b, y)$;

(b) if $y = \epsilon$, let $(d, (q, a, \epsilon, 0), -e, YZ)$ be in $\delta_B((q', b, \epsilon, 0), b, Y)$;

(c) if $y = Y_1Y_2$, Y_1 and Y_2 in Γ, let $(0, (q, b, ZY_1, d), 0, \epsilon)$ be in $\delta_B((q', b, \epsilon, 0), b, Y_2)$ and $(d, (q, a, \epsilon, 0), 0, Z)$ in $\delta_B((q, b, ZY_1, d), b, Y_1)$.

(ii) $(1, f_1, 0, Z_0)$ is in $\delta_B((q_0, b, \epsilon, 0), b, Z_0)$.

(iii) $(1, f_1, 0, Z_0)$ is in $\delta_B(f_1, \$, Z_0)$.

[18] Let E be an abstract set. Let $\epsilon^R = \epsilon$ and $(xa)^R = ax^R$, for a in E, and x in E^*. For $U \subseteq E^*$, let $U^R = \{w^R \mid w \text{ in } U\}$. U^R is called the *reversal* of U.

Intuitively, (q, a, X, d) is an analogue of q in A and represents the fact that in A, a is applied at q. (a), (b), and (c) of (i) enable B to simulate "moving backwards" in A. In particular, suppose that at state q, under input a, and reading Z on the stack; A moves to q' and writes y on the stack. Then at $(q', b, \epsilon, 0)$, an analogue to q', under the influence of y on the stack; B goes to an analogue of q which acts only under a. Formally, w is in $T(A)$ if and only if there exist $q_1, \cdots, q_k$, $a_0, \cdots, a_{k-1}$, $w_0, \cdots, w_{k-1}$, $y_0, y_0', \cdots, y_k, y_k'$ such that $q_k = f$, $y_0 = y_k = Z_0$, $y_0' = y_k' = \epsilon$, $a_0w_0 = w$, and

$$\begin{aligned}(q_0, a_0w_0, y_0\uparrow y_0') &\vdash_A (q_1, a_1w_1, y_1\uparrow y_1')\\ &\cdots\\ &\vdash_A (q_{k-1}, a_{k-1}w_{k-1}, y_{k-1}\uparrow y_{k-1}')\\ &\vdash_A (q_k, \epsilon, y_k\uparrow y_k').\end{aligned}$$

This holds if and only if there exist $v_1, \cdots, v_{k-1}$ such that $v_ia_iw_i = w$ for each i and

$$\begin{aligned}((f, ¢, \epsilon, 0), ¢w^R\$, Z_0\uparrow) &\vdash_B^* ((q_{k-1}, a_{k-1}, \epsilon, 0), a_{k-1}v_{k-1}^R\$, y_{k-1}\uparrow y_{k-1}')\\ &\vdash_B^* ((q_{k-2}, a_{k-2}, \epsilon, 0), a_{k-2}v_{k-2}^R\$, y_{k-2}\uparrow y_{k-2}')\\ &\cdots\\ &\vdash_B^* ((q_0, a_0, \epsilon, 0), a_0\$, y_0\uparrow y_0')\\ &\vdash_B (f_1, \$, Z_0\uparrow)\\ &\vdash_B (f_1, \epsilon, Z_0\uparrow),\end{aligned}$$

i.e., w^R is in $T(B)$.

3. *Operations Which Do Not Preserve Languages*

In this section we exhibit certain operations which do not preserve languages. This is done with the help of a representation theorem for recursively enumerable sets.

Definition. A *phrase structure grammar* is a 4-tuple $G = (V, \Sigma, P, \sigma)$, where (i) V is a finite nonempty set, (ii) $\Sigma \subseteq V$, (iii) P is a finite set of ordered pairs (u, v), written $u \to v$, with u in $(V - \Sigma)^* - \{\epsilon\}$ and v in V^*, and (iv) σ is in $V - \Sigma$.

A phrase structure grammar selects a certain set of words as follows.

Notation. Let $G = (V, \Sigma, P, \sigma)$ be a phrase structure grammar. For w, y in V^*, write $w \Rightarrow y$ if there exist z_1, z_2, u, v such that $w = z_1uz_2$, $y = z_1vz_2$, and $u \to v$ is in P. For w, y in V^*, write $w \overset{*}{\Rightarrow} y$ if there exist $r \geq 0$ and $z_0, \cdots, z_r$ such that $z_0 = w$, $z_r = y$, and $z_i \Rightarrow z_{i+1}$ for $0 \leq i < r$. Let $L(G)$, called the set "generated" by G, be the set $\{x \text{ in } \Sigma^* \mid \sigma \overset{*}{\Rightarrow} x\}$.

The following proposition is well known [2]: "For $L \subseteq \Sigma^*$, $L = L(G)$ for some phrase structure grammar G if and only if L is recursively enumerable."[19]

We now associate a set of a special form with each phrase structure grammar.

Notation. Let $G = (V, \Sigma, P, \sigma)$ be a phrase structure grammar. Let c be a new symbol, $P = \{u_i \to v_i \mid 1 \leq i \leq k\}$, and let $d_1, \cdots, d_k$ be k new symbols. Let $L_1'(G)$ be the set

$$L_1'(G) = \{w_1u_id_iw_2cw_2^Rv_i^Rw_1^Rc \mid w_1, w_2 \text{ in } V^*, \\ u_i \to v_i \text{ in } P, \ 1 \leq i \leq k\}^* - \{\epsilon\}.$$

[19] See [3] for a definition and a discussion of recursively enumerable sets.

LEMMA 3.1. *$L_1'(G)$ is a deterministic context-free language*[20] *for each phrase structure grammar G.*

PROOF. For $u_i \to v_i$ in P, $1 \leq i \leq k$, let $u_i = u_{i1} \cdots u_{im(i)}$ and $v_i = v_{i1} \cdots v_{in(i)}$, each u_{ij} and v_{ij} in V.[21] Let $A = (K, \Sigma', \Gamma, \delta, Z_0, q_0, \{f\})$ be the deterministic pda[22] with $\Sigma' = V \cup \{c, d_1, \cdots, d_k\}$,

$$K = \{q_0, q_1, q_2, d, f\} \cup \{s_{ij} \mid 1 \leq i \leq k, \quad 0 \leq j < m(i)\} \cup \{t_{ij} \mid 1 \leq i \leq k, \quad 0 \leq j < n(i)\},$$

$\Gamma = \Sigma' \cup \{Z_0\}$, Z_0 being a new symbol, and δ defined as follows:

(i) $\delta(q_0, a, Z_0) = (q_0, Z_0 a)$ for each a in V.
$\delta(q_0, a, b) = (q_0, ba)$ for each a and b in $V \cup \{d_i \mid 1 \leq i \leq k\}$.
$\delta(q_0, c, b) = (q_1, b)$ for each b in $V \cup \{d_i \mid 1 \leq i \leq k\}$.

(ii) $\delta(q_1, a, a) = (q_1, \epsilon)$ for each a in V.
$\delta(q_1, \epsilon, d_i) = (s_{i0}, \epsilon)$ for each $1 \leq i \leq k$.

(iii) $\delta(s_{ij}, \epsilon, u_{i(m(i)-j)}) = (s_{i(j+1)}, \epsilon)$ for each $1 \leq i \leq k$, $0 \leq j < m(i) - 1$.
$\delta(s_{i(m(i)-1)}, \epsilon, u_{i1}) = (t_{i0}, \epsilon)$ for each $1 \leq i \leq k$.

(iv) $\delta(t_{ij}, v_{i(n(i)-j)}, a) = (t_{i(j+1)}, a)$ for each a in $V \cup \{Z_0\}$, $1 \leq i \leq k$, $0 \leq j < n(i) - 1$.
$\delta(t_{i(n(i)-1)}, v_{i1}, a) = (q_2, a)$ for[23] each $1 \leq i \leq k$, a in $V \cup \{Z_0\}$.

(v) $\delta(q_2, a, a) = (q_2, \epsilon)$ for each a in V.
$\delta(q_2, c, Z_0) = (f, Z_0)$.

(vi) $\delta(f, \epsilon, Z_0) = (q_0, Z_0)$.

(vii) $\delta(q, a, b) = (d, b)$ for all (q, a, b) not previously specified.

The pda A operates as follows:

(1) In (i), A copies the input onto the stack until the symbol c is read.

(2) In (ii), it checks the input against the stack until some d_i is read on the stack. The d_i is then erased.

(3) Without moving the input, A checks in (iii) to see if the word u_i is on the stack (erasing u_i in the process).

(4) In (iv), A sees if v_i^R is on the input while not altering the stack.

(5) In (v), A matches the input against the stack until c is read.

(6) In (v), A goes to an accepting state if c is read on the input and Z_0 on the stack.

(7) In (vi), A then goes to the start state, the cycle (beginning at step (1)) being repeated.

It is a straightforward matter to verify that $T(A) = L_1'(G)$.

The following notation is useful.

Notation. For a in Σ, let $\hat{a}$ be a new symbol. Let $\hat{\epsilon} = \epsilon$ and $\hat{w} = \hat{a}_1 \cdots \hat{a}_k$ for each $w = a_1 \cdots a_k$, all a_i in Σ. For each $U \subseteq \Sigma^*$, let $\hat{U} = \{\hat{u} \mid u \text{ in } U\}$. For each phrase structure grammar G, let $L_1(G) = L_1'(G)(\hat{\Sigma})^*$.

COROLLARY. *$L_1(G)$ is a deterministic context-free language.*

PROOF. $(\hat{\Sigma})^*$ is regular and $L_1'(G)$ is a deterministic context-free language. The

[20] A set L is said to be a *deterministic context-free language* if $L = T(A)$ for some deterministic pda.

[21] By definition of a phrase structure grammar, $m(i) \geq 1$ and $n(i) \geq 0$ for each i.

[22] We use the definition of deterministic pda as given in [4] rather than the specialization of a deterministic one-way sa given in Section 1.

[23] If $n(i) = 0$, this becomes $\delta(t_{i0}, \epsilon, a) = (q_2, a)$.

result then follows from the fact that deterministic context-free languages are closed under product on the right with regular sets [4, Theorem 3.3].

Notation. Let V be a finite set containing σ and let $c, d_1, \cdots, d_k$ be $k+1$ new symbols. Let

$$L(k, V) = \{\sigma d_i c \mid 1 \leq i \leq k\}\{y_1 y_2 c y_2^R d_j y_1^R c \mid 1 \leq j \leq k,\ y_1 \text{ and } y_2 \text{ in } V^*\}^* \{xc\hat{x}^R \mid x \text{ in } V^*\}.$$

LEMMA 3.2. *$L(k, V)$ is a deterministic context-free language.*

PROOF. Let A be the deterministic pda $(K, \Sigma', \Gamma, \delta, q_0, Z_0, \{f_1, f_2\})$, where

$$K = \{q_0, q_1, q_2, d, f_1, f_2\} \cup \{s_i \mid 0 \leq i \leq 4\},$$

$$\Sigma' = V \cup \hat{\Sigma} \cup \{c, d_1, \cdots, d_k\},$$

$$\Gamma = V \cup \hat{V} \cup \{Z_0\},$$

Z_0 being a new symbol, and δ is defined as follows (a and b are arbitrary elements in V):

(i) $\delta(q_0, \sigma, Z_0) = (q_1, Z_0)$
$\delta(q_1, d_i, Z_0) = (q_2, Z_0)$ for $1 \leq i \leq k$.
$\delta(q_2, c, Z_0) = (s_0, Z_0)$.

(ii) $\delta(s_0, a, Z_0) = (s_0, Z_0 a)$.
$\delta(s_0, a, b) = (s_0, ba)$.
$\delta(s_0, c, a) = (s_1, a)$.

(iii) $\delta(s_1, a, a) = (s_2, \epsilon)$.
$\delta(s_1, \hat{a}, a) = (s_3, \epsilon)$.

(iv) $\delta(s_2, a, a) = (s_2, \epsilon)$.
$\delta(s_2, d_j, a) = (s_4, a)$, $\delta(s_2, d_j, Z_0) = (s_4, Z_0)$.

(v) $\delta(s_3, \hat{a}, a) = (s_3, \epsilon)$.
$\delta(s_3, \epsilon, Z_0) = (f_1, Z_0)$.

(vi) $\delta(s_4, a, a) = (s_4, \epsilon)$.
$\delta(s_4, c, Z_0) = (s_0, Z_0)$.

(vii)[24] $\delta(s_0, c, Z_0) = (f_2, Z_0)$.
$\delta(f_2, d_i, Z_0) = (s_4, Z_0)$ for $1 \leq i \leq k$.

(viii) $\delta(q, a', Z) = (d, Z)$ for all (q, a', Z) in $K \times \Sigma' \times \Gamma$ not previously defined.

Intuitively, A operates as follows:

(1) In (i), A checks to see if the first three input symbols are $\sigma d_i c$ for some i.

(2) In (ii), A copies the input symbols onto the stack until the symbol c is read.

(3) In (iii) and (iv), it matches the input against the stack until some d_j is read on the input.

(4) In (iv), the d_j is read on the input without altering the stack.

(5) In (vi), the input is again matched against the stack until the symbol c is read.

(6) A then goes to s_0 and cycles.

(7) If at step (3), a symbol $\hat{a}$ is read, then A goes to s_3 where it compares, in (v), $\hat{a}$ on the input with b on the stack.

(8) A accepts if, after reading the input, the stack is empty except for Z_0.

[24] The rules in (vii) are needed in case $y_1 y_2 = \epsilon$ or $x = \epsilon$.

It is a straightforward matter to formally verify that $T(A) = L(k, V)$. The details are omitted.

We now obtain a representation theorem for recursively enumerable sets.

THEOREM 3.1. *For each recursively enumerable set $E \subseteq \Sigma^*$, there exist deterministic context-free languages L_1 and L_2, and a homomorphism h such that*

$$E = h(L_1 \cap L_2).$$

PROOF. Let $E = L(G)$, where $G = (V, \Sigma, P, \sigma)$ is a phrase structure grammar. Let $L_1 = L_1(G)$ and $L_2 = L(k, V)$, where k is the number of productions in P. Let h be the homomorphism defined by $h(\hat{a}) = a$ for each $\hat{a}$ in $\hat{\Sigma}$ and $h(a) = \epsilon$ for each a in $V \cup \{c, d_1, \cdots, d_k\}$.

Let x be a word in E. Then there exist $w_{j1}, w_{j2}, u_{g(j)}, v_{g(j)}$ for $0 \leq j \leq m$ such that $w_{01} = w_{02} = \epsilon$, $u_{g(0)} = \sigma$, $w_{m1}v_{g(m)}w_{m2} = x$,

$$w_{j1}u_{g(j)}w_{j2} \Longrightarrow w_{j1}v_{g(j)}w_{j2},$$

with $u_{g(j)} \rightarrow v_{g(j)}$ in P, for $0 \leq j \leq m$, and

$$w_{j1}v_{g(j)}w_{j2} = w_{(j+1)1}u_{g(j+1)}w_{(j+1)2}$$

for $1 \leq j \leq m - 1$. For each w_1, i, and w_2, let

$$s(w_1, i, w_2) = w_2^R w_1^R c w_1 d_i w_2 c$$

and $t(w_1, i, w_2) = w_1 u_i d_i w_2 c w_2^R v_i^R w_1^R c$. Then

$$z = t(w_{01}, g(0), w_{02})t(w_{11}, g(1), w_{12}) \cdots t(w_{m1}, g(m), w_{m2})\hat{x}$$

is in $L_1(G)$. However,

$$z = \sigma d_{g(0)} c s(w_{11}u_{g(1)}, g(1), w_{12}) \cdots s(w_{m1}u_{g(m)}, g(m), w_{m2})x^R c\hat{x}.$$

Therefore z is in $L(k, V)$, i.e., z is in $L_1 \cap L_2$. Then $x = h(z)$ is in $h(L_1 \cap L_2)$, i.e., $E \subseteq h(L_1 \cap L_2)$.

Suppose that x is in $h(L_1 \cap L_2)$. Thus $x = h(z)$ for some z in $L_1 \cap L_2$. Then

$$z = t(w_{01}, g(0), w_{02}) \cdots t(w_{m1}, g(m), w_{m2})\hat{x}$$

for some $w_{01}, w_{02}, \cdots, w_{m1}, w_{m2}, g(0), \cdots, g(m)$. Similarly

$$z = \sigma d_{f(0)} c s(y_{11}, f(1), y_{12}) \cdots s(y_{n1}, f(n), y_{n2})x^R c\hat{x}$$

for some $y_{11}, y_{12}, \cdots, y_{n1}, y_{n2}, f(0), \cdots, f(n)$. Then $m = n$ (since there are exactly $2m+2$ and $2n+2$ occurrences of c in z). By observing the occurrences of d_j in z, we see that $f(i) = g(i)$ for $0 \leq i \leq m$. We also see that

$$w_{01} = w_{02} = \epsilon, \quad u_{g(0)} = \sigma, \quad w_{i1}u_{g(i)} = y_{i1}, \quad w_{i2} = y_{i2},$$

$$w_{(i-1)1} v_{g(i-1)} w_{(i-1)2} = y_{i1}y_{i2} = w_{i1}u_{g(i)}w_{i2}$$

for $0 \leq i \leq m$, and $w_{m1}v_{g(m)}w_{m2} = x$. Thus

$$\sigma \Longrightarrow w_{01}v_{g(0)}w_{02} \Longrightarrow w_{11}v_{g(1)}w_{12} \Longrightarrow \cdots \Longrightarrow w_{m1}v_{g(m)}w_{m2} = x,$$

i.e., x is in E. Then $h(L_1 \cap L_2) \subseteq E$ and the proof is complete.

Remark. We mention without proof that using a standard coding technique, Theorem 3.1 can be extended to the following: Given Σ, let $\Sigma_0 = \hat{\Sigma} \cup \{a, b\}$, a and

b being two new symbols. There exists a homomorphism h of Σ_0^* onto Σ^* with the following property. For each recursively enumerable set $E \subseteq \Sigma^*$, there exist deterministic context-free languages $L_1 \subseteq \Sigma_0^*$ and $L_2 \subseteq \Sigma_0^*$ such that $E = h(L_1 \cap L_2)$.

Using the representation theorem, several nonclosure results are now obtained. In Section 4, other nonclosure results are obtained from the representation theorem.

THEOREM 3.2. *(a) The family of languages does not contain the intersection of some pairs of deterministic context-free languages and does not contain the complement of some context-free languages.*

(b) The family of languages is not closed under intersection or complementation.

PROOF. It obviously suffices to only prove (a). To this end, let E be a recursively enumerable, nonrecursive set. Such a set does exist [3]. In particular, E is not a language. By Theorem 3.1, there exist deterministic context-free languages L_1 and L_2 and a homomorphism h such that $h(L_1 \cap L_2) = E$. Suppose the family of languages contains the intersection of each pair of deterministic context-free languages. By the corollary to Theorem 2.6, $h(L_1 \cap L_2) = E$ is a language, a contradiction. Suppose the family of languages contains the complement of each context-free language. Since the family of languages contains the union of context-free languages, it thus contains the intersection of context-free languages, a contradiction.

In Theorem 4.1, we prove that the complement of a D-language is a D-language. By (a) of Theorem 3.2, languages are not closed under complementation. Thus we have:

COROLLARY. *There exists a context-free language, thus a language, which is not a D-language.*

4. *Closure Properties of D-Languages*

We now treat some basic closure properties of D-languages. In particular, we show that D-languages are preserved under (i) complementation, (ii) removal of a fixed initial subword, and (iii) removal of a fixed final subword.

We first turn to the proof of closure under complementation. The argument involves a number of intermediate steps and auxiliary concepts. The principal difficulty is that a deterministic one-way sa may fail to leave the input because it gets into a loop or attempts to write in the middle of the stack.

We first define a new relation $\vdash^s$ on ID's. Intuitively, $\vdash^s$ is an atomic move which leaves the contents of the stack unchanged (although the stack pointer may move).

Notation. Let $A = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$ be a one-way sa. Write

$$(q, ax, y_1 \uparrow y_2) \vdash^s (q', x', y_1' \uparrow y_2')$$

if $(q, ax, y_1 \uparrow y_2) \vdash (q', x', y_1' \uparrow y_2')$ and $y_1 y_2 = y_1' y_2'$. Write

$$(q, ax, y_1 \uparrow y_2) \vdash^{*s} (q', x', y_1' \uparrow y_2')$$

if there exist ID's $z_0, \cdots, z_r$, $r > 0$, such that $z_0 = (q, ax, y_1 \uparrow y_2)$, $z_r = (q', x', y_1' \uparrow y_2')$, and $z_i \vdash^s z_{i+1}$ for $0 \le i < r$.

The first two lemmas involve regular sets and one-way sa.

LEMMA 4.1. *Let $A = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$ be a one-way sa. For each p, q in K, and a in $\Sigma \cup \{¢, \$\}$, the set*

$$R_{pqa} = \{x \text{ in } \Gamma^* \mid (p, a, x\uparrow) \vdash^{*s} (q, a, x\uparrow)\}$$

is regular.

PROOF. Without loss of generality, it may be assumed that $K \cap \Gamma = \phi$. Let (p, q, a) in $K \times K \times (\Sigma \cup \{¢, \$\})$ be given. Let P be the following set of "rules." For each (s, s', Z, Z') in $K \times K \times \Gamma \times \Gamma$, if $(0, s', e, Z)$ is in $\delta(s, a, Z)$

(i) and $e = 0$, let $\mathbf{Q}_1 Z s \mathbf{Q}_2 \rightarrow \mathbf{Q}_1 Z s' \mathbf{Q}_2$ be in P;

(ii) and $e = 1$, let $\mathbf{Q}_1 Z s Z' \mathbf{Q}_2 \rightarrow \mathbf{Q}_1 Z Z' s \mathbf{Q}_2$ be in P;

(iii) and $e = -1$, let $\mathbf{Q}_1 Z s \mathbf{Q}_2 \rightarrow \mathbf{Q}_1 s' Z \mathbf{Q}_2$ be in P.

For w, w', y, y' in $(K \cup \Gamma)^*$ and $\mathbf{Q}_1 y \mathbf{Q}_2 \rightarrow \mathbf{Q}_1 y' \mathbf{Q}_2$ in P, write $wyw' \overset{2}{\Rightarrow} wy'w'$. For w and w' in $(K \cup \Gamma)^*$, write $w \overset{*2}{\Rightarrow} w'$ if there exist $w_0, \cdots, w_r$, $r > 0$, such that $w_0 = w$, $w_r = w'$, and $w_i \overset{2}{\Rightarrow} w_{i+1}$ for $0 \leq i < r$. Clearly $(p, a, x\uparrow) \vdash^{*s} (q, a, x\uparrow)$ if and only if $xp \overset{*2}{\Rightarrow} xq$. By Lemma 3.1 of [5],[25] $\{x \text{ in } \Gamma^* \mid xp \overset{*2}{\Rightarrow} xq\}$ is regular. Therefore R_{pqa} is regular.

The next lemma associates with each one-way sa an equivalent one-way sa that checks the contents of the stack for containment in a finite number of regular sets.

LEMMA 4.2. *Let $A = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$ be a one-way sa and for each i, $1 \leq i \leq n$; let $R_i \subseteq \Gamma^*$ be a regular set. Then there exist a one-way sa $A' = (K, \Sigma, ¢, \$, \Gamma', \delta', q_0, Z_0', F)$, a mapping ψ_i of Γ' into $\{0, 1\}$ for each $1 \leq i \leq n$, and a (length-preserving) homomorphism μ of $(\Gamma')^*$ into Γ^* satisfying the following conditions:*

(1) $T(A) = T(A')$.

(2) *Suppose* $(p, x_1x_2, v_1\uparrow v_2) \vdash^*_{A'} (q, x_2, y_1\uparrow y_2)$. *Then*

(*a*) $(p, x_1x_2, \mu(v_1)\uparrow\mu(v_2)) \vdash^*_A (q, x_2, \mu(y_1)\uparrow\mu(y_2))$, *and*

(*b*) *if* $p = q_0$, $v_1 = Z_0'$, $v_2 = \epsilon$, *and* $y_1y_2 = Z_1 \cdots Z_m$, *each* Z_i *in* Γ';

then for each $1 \leq r \leq m$, $1 \leq i \leq n$, $\psi_i(Z_r) = 1$ *if and only if* $\mu(Z_1 \cdots Z_r)$ *is in* R_i.

(3) *Suppose* $(q, x_1x_2, y_1\uparrow y_2) \vdash^*_A (q', x_2, z_1\uparrow z_2)$. *Then there exist* y_1', y_2', z_1', z_2' *in* $(\Gamma')^*$ *such that*

(*a*) $\mu(y_i') = y_i$, $\mu(z_i') = z_i$ *for* $i = 1, 2$, *and*

(*b*) $(q, x_1x_2, y_1'\uparrow y_2') \vdash^*_{A'} (q', x_2, z_1'\uparrow z_2')$.

(4) *If A is deterministic, then so is A'.*

PROOF. For $1 \leq i \leq n$, let $A_i = (K_i, \Gamma, \delta_i, q_i, F_i)$ be an fsa such that $T(A_i) = R_i$. Without loss of generality, we may assume that $K_i \cap K_j = \phi$ for $i \neq j$ and that $K_i \cap K = \phi$ for each i. Let $\Gamma' = \Gamma \times K_1 \times \cdots \times K_n$ and $Z_0' = (Z_0, q_1, \cdots, q_n)$. Let μ be the (length-preserving) homomorphism of $(\Gamma')^*$ into Γ^* defined by $\mu((Z, t_1, \cdots, t_n)) = Z$ for each $(Z, t_1, \cdots, t_n)$ in Γ'. For each $1 \leq i \leq n$, let ψ_i be the mapping of Γ' into $\{0, 1\}$ defined by $\psi_i((Z, t_1, \cdots, t_n)) = 1$ if $\delta_i(t_i, Z)$ is in F and 0 otherwise.

Let δ' be defined as follows: For each t_i in K_i, $1 \leq i \leq n$, and each (d, q', e, y) in $\delta(q, a, Z)$:

(5) if $y = \epsilon$, let (d, q', e, ϵ) be in $\delta'(q, a, (Z, t_1, \cdots, t_n))$;

(6) if $y = Z_1 \cdots Z_j$, $j \geq 1$, each Z_i in Γ, let

$$(d, q', e, (Z_i, t_1, \cdots, t_n)(Z_2, \delta_1(t_1, Z_1), \cdots, \delta_n(t_n, Z_1)) \cdots (Z_j, \delta_1(t_1, Z_1 \cdots Z_{j-1}), \cdots, \delta_n(t_n, Z_1 \cdots Z_{j-1})))$$

be in $\delta'(q, a, (Z, t_1, \cdots, t_n))$.

Clearly A' is deterministic if A is. Condition (3) holds by construction. Consider

[25] The definition of $\overset{*2}{\Rightarrow}$ in [5] assumed that $r \geq 0$, while we now assume that $r > 0$. It is easily seen that the proof of Lemma 3.1 of [5] is still valid under the condition that $r > 0$.

condition (2). A straightforward induction on the number of moves of A' shows that (2a) holds, and if $p = q_0$, $v_1 = Z_0'$, $v_2 = \epsilon$, and $y_1y_2 = Z_1 \cdots Z_m$, where $Z_j = (Y_j, t_{j1}, \cdots, t_{jn})$ for $j = 1, \cdots, m$; then $t_{1i} = q_i$ for $1 \leq i \leq n$ and $t_{ki} = \delta_i(q_i, Z_1 \cdots Z_{k-1})$ for $1 \leq k \leq m$ and $1 \leq i \leq n$. Now from the definitions of ψ and μ, (2) holds. Condition (1) follows from (2) and (3).

We need three additional concepts in order to prove closure of D-languages under complementation. The first is now presented.

Definition. A one-way sa $A = (K, \Sigma, ¢, \$, \Gamma, q_0, Z_0, F)$ is said to be *continuing* if for each w_1w_2 in $¢\Sigma^*\$$, $w_2 \neq \epsilon$, with $(q_0, w_1w_2, Z_0|) \vdash^* (q, w_2, \gamma)$; there exist q', w_2', γ' such that $(q, w_2, \gamma) \vdash (q', w_2', \gamma')$.

Thus a continuing one-way sa is a one-way sa in which each sequence of ID's from q_0 and Z_0 with a nonzero input in the last ID can be extended.

The next lemma shows that when dealing with deterministic one-way sa, there is no loss of generality in assuming that the device is continuing.

LEMMA 4.3. *For each deterministic one-way sa A, there exists a continuing one-way sa B such that $T(A) = T(B)$.*

PROOF. Let $A = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$ be a one-way deterministic sa. Without loss of generality, it may be assumed that for each (q, a) in $K \times (\Sigma \cup \{¢, \$\})$, $\delta(q, a, Z_0) = (d, q', e, Z_0y)$, with $e \geq 0$. (For otherwise we could consider the deterministic one-way sa $(K \cup \{p_0\}, \Sigma, ¢, \$, \Gamma \cup \{U_0\}, \delta_1, p_0, U_0, F)$, where p_0 and U_0 are new symbols, and δ_1 is defined as follows for each (q, a, Z) in $K \times (\Sigma \cup \{¢, \$\}) \times \Gamma$: $\delta_1(p_0, a, U_0) = (0, q_0, 0, U_0Z_0)$, $\delta_1(p_0, a, Z) = (0, q_0, 0, Z)$, $\delta_1(q, a, U_0) = (0, q, 0, U_0)$, and $\delta_1(q, a, Z) = \delta(q, a, Z)$.) For each q in K and Z in Γ, let $\bar{q}$ and $\bar{Z}$ be new symbols. Let V_0 and h also be new symbols. Let $B = (K_B, \Sigma, ¢, \$, \Gamma_B, \delta_B, q_0, V_0, F_B)$, where $\bar{K} = \{\bar{q} \mid q \text{ in } K\}$, $K_B = K \cup \bar{K} \cup \{h\}$, $\Gamma_B = \Gamma \cup \{\bar{Z} \mid Z \text{ in } \Gamma\} \cup \{V_0\}$, $F_B = F \cup \{\bar{q} \mid q \text{ in } F\}$, and δ_B is defined as follows (for each (q, a, Z) in $K \times (\Sigma \cup \{¢, \$\}) \times \Gamma$):

(i) $\delta_B(h, a, Y) = (1, h, 0, Y)$ for each Y in Γ_B.

(ii) (a) $\delta_B(q_0, ¢, V_0) = (0, q_0, 0, V_0\bar{Z}_0)$.

(b) $\delta_B(p, b, V_0) = (1, h, 0, V_0)$ for each $(p, b) \neq (q_0, ¢)$ in $(K \cup \bar{K}) \times (\Sigma \cup \{¢, \$\})$.

Let $\delta(q, a, Z) = (d, q', e, y)$.

(iii) (a) If $y = Z$, let $\delta_B(q, a, Z) = \delta(q, a, Z)$.

(b) If $y \neq Z$, let $\delta_B(q, a, Z) = (1, h, 0, Z)$.

(iv) (a) If $y = Z$ and $e \leq 0$, let $\delta_B(q, a, \bar{Z}) = (d, q', e, \bar{Z})$.

(b) If $y = Z$ and $e = 1$, let $\delta_B(q, a, \bar{Z}) = (1, h, 0, \bar{Z})$.

(c) If $y \neq Z$ and $y = vY$, Y in Γ, let $\delta_B(q, a, \bar{Z}) = (d, q', e, v\bar{Y})$.

(d) If $y = \epsilon$, let $\delta_B(q, a, \bar{Z}) = (d, \bar{q}', e, \epsilon)$.

(v) $\delta_B(\bar{q}, a, Z) = \delta_B(\bar{q}, a, \bar{Z}) = (0, q, 0, \bar{Z})$.

The intuitive idea for B is as follows. In a computation, B is to simulate A if A does not "block." If A "blocks," then B goes to the dead state h. To do this, a new leftmost stack symbol V_0 is introduced, and the rightmost symbol on the stack is marked. Since $\delta(q, a, Z_0) = (d, q', e, Z_0y)$, with $e \geq 0$, the "blocking" of A occurs if and only if the stack pointer of B is either (α) in the interior of the stack and the instruction in A is to write (this occurs at (iii-b)), or (β) at the rightmost symbol on the stack and the instruction in A is to move the stack pointer right (this occurs at (iv-b)). The simulation of A by B is in (iii), (iv), and (v). In particular, (iv-d) and (v) together perform an erase and a marking of the new rightmost symbol on the stack.

Formally, suppose $¢w\$$ is in $¢\Sigma^*\$$. Either

(1) there exists an infinite sequence of ID's

$$(q_0, u_0, \gamma_0) \vdash_A \cdots \vdash_A (q_i, u_i, \gamma_i) \vdash_A \cdots,$$

with $u_0 = ¢w\$$ and $\gamma_0 = Z_0\rceil$; or

(2) there exists a finite sequence of ID's

$$(q_0, u_0, \gamma_0) \vdash_A \cdots \vdash_A (q_k, u_k, \gamma_k),$$

with $u_0 = ¢w\$$, $\gamma_0 = Z_0\rceil$, and $u_k = \epsilon$; or

(3) there exists a finite sequence of ID's

$$(q_0, u_0, \gamma_0) \vdash_A \cdots \vdash_A (q_k, u_k, \gamma_k),$$

with $u_0 = ¢w\$$, $\gamma_0 = Z_0\rceil$, $u_k \neq \epsilon$, and for no q, u, γ does $(q_k, u_k, \gamma_k) \vdash_A (q, u, \gamma)$.

For each i, if $\gamma_i = \alpha_i\rceil\alpha_i'Y_i$ for some Y_i in Γ, let $\tau_i = V_0\alpha_i\rceil\alpha_i'\bar{Y}_i$; and if $\gamma_i = \alpha_iY_i\rceil$ for some Y_i in Γ, let $\tau_i = V_0\alpha_i\bar{Y}_i\rceil$. If (1) holds, then

(4) $(q_0, u_0, V_0\rceil) \vdash_B (q_0, u_0, \tau_0) \vdash_B^* \cdots \vdash_B^* (q_i, u_i, \tau_i) \vdash_B^* \cdots$.

If (2) holds, then

(5) $(q_0, u_0, V_0\rceil) \vdash_B (q_0, u_0, \tau_0) \vdash_B^* \cdots \vdash_B^* (q_k, u_k, \tau_k)$.

Suppose that (3) holds. Then

(6) $(q_0, u_0, V_0\rceil) \vdash_B (q_0, u_0, \tau_0) \vdash_B^* \cdots \vdash_B^* (q_k, u_k, \tau_k)$.

Since $u_k \neq \epsilon$, $u_k = a_ku_k'$ for some a_k in $\Sigma \cup \{¢, \$\}$. Now either $\gamma_k = \alpha_k'W_k\rceil\alpha_kY_k$, W_k in Γ, and $\delta(q_k, a_k, W_k) = (d_k, q_{k+1}, e_k, y_k)$, with $W_k \neq y_k$; or $\gamma_k = \alpha_kY_k\rceil$ and $\delta(q_k, a_kY_k) = (d_k, q_{k+1}, 1, y_k)$. In the former case, $\delta_B(q_k, a_k, W_k) = (1, h, 0, W_k)$. In the latter, $\delta_B(q_k, a_k, \bar{Y}_k) = (1, h, 0, \bar{Y}_k)$. Thus

$$(q_k, u_k, \tau_k) \vdash_B (h, u_k', \tau_k) \vdash_B^* (h, \epsilon, \tau_k).$$

Hence $T(B) = T(A)$ and B is continuing.

We now introduce the second auxiliary concept used in proving the closure of D-languages under complementation.

Definition. A one-way sa $A = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$ is said to be *directed* if it is deterministic and the following holds for each w_1aw_2 in $¢\Sigma^*\$$, a in $\Sigma \cup \{¢, \$\}$: If $(q_0, w_1aw_2, Z_0\rceil) \vdash^* (p, aw_2, x\rceil x')$ and $(p, a, x\rceil) \vdash^{*s} (q, a, x\rceil)$, then (i) $p \neq q$ and (ii) $(p, a, x\rceil) \vdash^s (q, a, x\rceil)$.

Intuitively, A is directed if there is no computation from q_0 and Z_0 containing an ID $(p, aw_2, x\rceil x')$ with the following property: There is a sequence of at least two consecutive moves starting from $(p, aw_2, x\rceil)$ which (i) preserves the stack at each move, (ii) does not advance the input, and (iii) returns to the right end of the stack. Less formally, A is directed if whenever the device goes left on the stack, it cannot go right without advancing the input tape. The next result shows that every deterministic one-way sa is equivalent to a directed sa.

LEMMA 4.4. *For each deterministic one-way sa $A = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$, there exists a directed sa B such that $T(A) = T(B)$. Furthermore, B is continuing if A is continuing.*

PROOF. For each p, q in K, and a in $\Sigma \cup \{¢, \$\}$, let

$$R_{pqa}(A) = \{\gamma \text{ in } \Gamma^* \mid (p, a, \gamma\uparrow) \vdash_A^{*S} (q, a, \gamma\uparrow)\}.$$

By Lemma 4.1, each set $R_{pqa}(A)$ is regular. By Lemma 4.2, there is a deterministic one-way sa $A' = (K, \Sigma, ¢, \$, \Gamma', \delta', q_0, Z_0', F)$, a mapping ψ_{pqa} of Γ' into $\{0, 1\}$ for each (p, q, a) in $K \times K \times (\Sigma \cup \{¢, \$\})$, and a homomorphism μ of $(\Gamma')^*$ into Γ^* satisfying the conditions of the lemma. By (2) and (3) of Lemma 4.2,

$$\mu(R_{pqa}(A')) = R_{pqa}(A)$$

for each (p, q, a) in $K \times K \times (\Sigma \cup \{¢, \$\})$.

Let $B = (K \cup \{h\}, \Sigma, ¢, \$, \Gamma', \bar{\delta}, q_0, Z_0', F)$ be the deterministic one-way sa in which h is a new symbol and $\bar{\delta}$ is defined as follows (for each (q, a, Z) in $K \times (\Sigma \cup \{¢, \$\}) \times \Gamma'$):

(i) $\bar{\delta}(h, a, Z) = (1, h, 0, Z)$. (Thus h is a "dead" state.)

(ii) If $\delta'(q, a, Z) = (d, q', e, y)$ and (α) $e = 1$, or (β) $d = 1$, or (γ) $y \neq Z$; then let $\bar{\delta}(q, a, Z) = (d, q', e, y)$.[26]

(iii) Suppose $\delta'(q, a, Z) = (0, q', e, Z)$ and $e \leq 0$.

(a) If $\psi_{qpa}(Z) = \psi_{ppa}(Z) = 1$ for some p in K, then let $\bar{\delta}(q, a, Z) = (1, h, 0, Z)$.[27]

(b) If there exists p in K such that $\psi_{qpa}(Z) = 1$ and $\psi_{pp'a}(Z) = 0$ for each p' in K, then let $\bar{\delta}(q, a, Z) = (0, p, 0, Z)$.[28]

(c) If neither (a) nor (b) holds, so that $\psi_{qpa}(Z) = 0$ for all p in K, then let $\bar{\delta}(q, a, Z) = (0, q', e, Z)$.[29]

We first show that $T(A) = T(B)$. Let w be an arbitrary word in $¢\Sigma^*\$$ and

$$(q_0, u_0, \gamma_0) \vdash_{A'} (q_1, u_1, \gamma_1) \vdash_{A'} \cdots \vdash_{A'} (q_i, u_i, \gamma_i) \vdash_{A'} \cdots$$

a (possibly infinite) sequence of ID's of A', with $u_0 = w$ and $\gamma_0 = Z_0'\uparrow$. To show that $T(A') = T(B)$, it suffices to prove that for each $i \geq 0$:

(1) Either (α) w is in both $T(A')$ and in $T(B)$, or (β) w is neither in $T(A')$ nor $T(B)$, or (γ) there exists $j \geq i$ such that $(q_0, u_0, \gamma_0) \vdash_B^* (q_j, u_j, \gamma_j)$.

Suppose $T(A) \neq T(B)$. Let i be the smallest integer for which (1) is false for some w. (Since (1γ) is true for $i = 0$, $i \geq 1$.) Then

$$(q_0, w_1aw_2, Z_0'\uparrow) \vdash_{A'}^* (q, aw_2, xZ\uparrow x')$$

and

$$(q_0, w_1aw_2, Z_0'\uparrow) \vdash_B^* (q, aw_2, xZ\uparrow x'),$$

where $q = q_{i-1}$, $w = w_1aw_2$, a in $\Sigma \cup \{¢, \$\}$, $aw_2 = u_{i-1}$, and $\gamma_{i-1} = xZ\uparrow x'$, Z in Γ'.

Suppose $\delta'(q, a, Z) = (d, q', e, y)$, with $e = 1$, $d = 1$, or $y \neq Z$. By (ii), $\bar{\delta}(q, a, Z) = (d, q', e, y)$. Then $(q, aw_2, xZ\uparrow x') \vdash_B (q_i, u_i, \gamma_i)$. This contradicts the choice of i.

Suppose $\delta'(q, a, Z) = (0, q', e, Z)$, with $e \leq 0$. Three cases arise.

[26] Any instruction in which A' moves right on the input or on the stack, or alters the stack cannot prevent A' from being directed. Hence it remains unchanged in B.

[27] In this case A' has some xZ on the stack and $(q, a, xZ\uparrow) \vdash_{A'}^{*S} (p, a, xZ\uparrow)$ and $(p, a, xZ\uparrow) \vdash_{A'}^{*S} (p, a, xZ\uparrow)$. Therefore A' is in a nontrivial loop. In B we break the loop and go to the dead state.

[28] In A', we have $(q, a, xZ\uparrow) \vdash_{A'}^{*S} (p, a, xZ\uparrow)$ while $(p, a, xZ\uparrow) \vdash_{A'}^{*S} (p', a, xZ\uparrow)$ is false for each p'. In B, we send the device directy from state q to state p.

[29] Since $(q, a, xZ\uparrow) \vdash_{A'}^{*S} (p, a, xZ\uparrow)$ is false for each p in K, we allow B to mimic A'

(a) Suppose $(q, a, xZ\uparrow) \vdash_{A'}^{*S} (p, a, xZ\uparrow) \vdash_{A'}^{*S} (p, a, xZ\uparrow)$ for some p. Then $(q, aw_2, xZ\uparrow x') \vdash_{A'}^{*} (p, aw_2, xZ\uparrow x') \vdash_{A'}^{*} (p, aw_2, xZ\uparrow x')$ and $w = w_1 a w_2$ is not in $T(A')$. By (iii-a), $\bar{\delta}(q, a, Z) = (1, h, 0, Z)$. Thus $(q, a, xZ\uparrow x') \vdash_B (h, \epsilon, xZ\uparrow x')$. Then $(q_0, w, Z_0'\uparrow) \vdash_B^{*} (h, \epsilon, xZ\uparrow x')$, so that w is not in $T(B)$, a contradiction.

(b) Suppose $(q, a, xZ\uparrow) \vdash_{A'}^{*S} (p, a, xZ\uparrow)$ and, for all p' in K, $(p, a, xZ\uparrow) \vdash_{A'}^{*S} (p', a, xZ\uparrow)$ is false. By (iii-b), $\bar{\delta}(q, a, Z) = (0, p, 0, Z)$. Then

$$(q_0, w, Z_0'\uparrow) \vdash_{A'}^{*} (q, aw_2, xZ\uparrow x') \vdash_{A'}^{*S} (p, aw_2, xZ\uparrow x')$$

$$= (q_j, u_j, \gamma_j) \quad \text{for some} \quad j \geq i$$

and

$$(q_0, w, Z_0'\uparrow) \vdash_B^{*} (q, aw_2, xZ\uparrow x') \vdash_B (p, aw_2, xZ\uparrow x') = (q_j, u_j, \gamma_j).$$

This contradicts the choice of i.

(c) Suppose $(q, a, xZ\uparrow) \vdash_{A'}^{*S} (p, a, xZ\uparrow)$ is false for each p in K. By (iii-c), $\bar{\delta}(q, a, Z) = \delta'(q, a, Z)$. Then $(q, aw_2, xZ\uparrow x') \vdash_{A'} (q_i, u_i, \gamma_i)$ and $(q, aw_2, xZ\uparrow x') \vdash_B (q_i, u_i, \gamma_i)$. This again yields a contradiction of i. Thus (1) is true for each i, so that $T(A') = T(B)$.

Now suppose that B is not directed. Then there exist a smallest integer $k \geq 0$ and an integer r satisfying the following:

(2) $r - k > 1$.

(3) There exist $u_0 = w_1 a w_2$ in $\cent\Sigma^*\$$, $u_k = aw_2$, a in $\Sigma \cup \{\cent, \$\}$, such that

$$(q_0, u_0, \gamma_0) \vdash_B \cdots \vdash_B (q_k, u_k, \gamma_k)$$

and

$$(q_k, a, \nu_k) \vdash_B^{S} \cdots \vdash_B^{S} (q_r, a, \nu_r),$$

with $\gamma_0 = Z_0'\uparrow$, $\gamma_k = xZ\uparrow x'$, Z in Γ', and $\nu_k = \nu_r = xZ\uparrow$.

Suppose that $\bar{\delta}(q_k, a, Z) = (d, q_{k+1}, e, y)$ and either $e = 1$, $d = 1$, or $y \neq Z$. Then $(q_k, a, \nu_k) \vdash_B^{S} (q_{k+1}, a, \nu_{k+1})$ is false. Therefore $\bar{\delta}(q_k, a, Z)$ is of type (iii).

Suppose one of the q_i is h. Then $q_r = h$ and $(q_{r-1}, a, \nu_{r-1}) \vdash_B (q_r, \epsilon, \nu_r)$, a contradiction. Thus no q_i is h. Thus $\bar{\delta}(q_k, a, Z)$ is of type (iii-b) or (iii-c).

At this point we note the following easily proved fact:

(4) Let w be in $\cent\Sigma^*\$$. If $(q_0, w, Z_0'\uparrow) \vdash_B^{*} (q, aw_2, y_1\uparrow y_2)$, a in Σ, $(q, a, y_1\uparrow) \vdash_B^{*S} (q', a, \gamma)$, and $q' \neq h$; then $(q_0, w, Z_0'\uparrow) \vdash_{A'}^{*} (q, aw_2, y_1\uparrow y_2)$ and $(q, a, y_1\uparrow) \vdash_{A'}^{*S} (q', a, \gamma)$.

Now suppose that $\bar{\delta}(q_k, a, Z)$ is of type (iii-b). Then $(q_k, a, xZ\uparrow) \vdash_B^{S} (q_{k+1}, a, xZ\uparrow)$ and $(q_{k+1}, a, xZ\uparrow) \vdash_B^{*S} (q_r, a, xZ)$. By (4), $(q_0, w_1 a w_2, Z_0'\uparrow) \vdash_{A'}^{*} (q_k, aw_2, xZ\uparrow x')$, $(q_k, a, xZ\uparrow) \vdash_{A'}^{*S} (q_{k+1}, a, xZ\uparrow)$, and $(q_{k+1}, a, xZ\uparrow) \vdash_{A'}^{*S} (q_r, a, xZ\uparrow)$. This contradicts $\bar{\delta}(q_k, a, Z)$ being of type (iii-b).

Finally, suppose that $\bar{\delta}(q_k, a, Z_k)$ is of type (iii-c). Since $(q_k, a, xZ\uparrow) \vdash_B^{*S} (q_r, a, xZ\uparrow)$, it follows from (4) that $(q_0, w_1 a w_2, Z_0'\uparrow) \vdash_{A'}^{*} (q_k, aw_2, xZ\uparrow x')$ and $(q_k, a, xZ\uparrow) \vdash_{A'}^{*S} (q_r, a, xZ\uparrow)$. This contradicts $\bar{\delta}(q_k, a, Z)$ being of type (iii-c). Thus the integer k cannot exist and B is directed.

If A is continuing, then A', as constructed in Lemma 4.2, is continuing. It is then easily seen that B is continuing.

We now present the third of the auxiliary concepts needed to prove the closure of D-languages under complementation.

Definition. A one-way sa $A = (K, \Sigma, \cent, \$, \Gamma, \delta, q_0, Z_0, F)$ is called *loop-free* if it is deterministic and, for each w in $¢\Sigma^*\$$, there exists y_1, y_2 in Γ^* and q in K such that

$$(q_0, w, Z_0\uparrow) \vdash^* (q, \epsilon, y_1\uparrow y_2).$$

Thus A is loop-free if it reads each word in $¢\Sigma^*\$$, that is, if it is continuing and does not operate forever on a given input tape.

LEMMA 4.5. *For every one-way sa A, there exists a loop-free one-way sa B such that $T(A) = T(B)$.*

PROOF. Let $A = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$. By Lemmas 4.3 and 4.4, it may be assumed that A is directed and continuing. Let $B = (K \cup \{h\}, \Sigma, ¢, \$, \Gamma, \delta_B, q_0, Z_0, F)$, where h is a new symbol and δ_B is defined as follows (for each (q, a, Z) in $K \times (\Sigma \cup \{¢, \$\}) \times \Gamma$):

(i) $\delta_B(h, a, Z) = (1, h, 0, Z)$.

(ii) If $\delta(q, a, Z) = (d, q', e, y)$ and (a) $d = 1$, or (b) $y = Z$, or (c) $y = \epsilon$; let $\delta_B(q, a, Z) = \delta(q, a, Z)$.

(iii) Suppose $\delta(q, a, Z) = (0, q', 0, y)$, $y \neq Z$, and $y \neq \epsilon$.

(a) If $(q, a, Z\uparrow) \vdash^* (q', a, \uparrow)$ for some q', let $\delta_B(q, a, Z) = \delta(q, a, Z)$.

(b) If $(q, a, Z\uparrow) \vdash^* (q', \epsilon, y_1\uparrow y_2)$ for some q', y_1, and y_2; let $\delta_B(q, a, Z) = \delta(q, a, Z)$.

(c) If $(q, a, Z\uparrow) \vdash^* (q', a, y_1\uparrow y_2)$ for some q', y_1, and $y_2 \neq \epsilon$; let $\delta_B(q, a, Z) = \delta(q, a, Z)$.

(d) In all other cases, let $\delta_B(q, a, Z) = (1, h, 0, Z)$.

Let $¢w\$$ be in $¢\Sigma^*\$$. Since A is continuing, either:

(1) there exists a finite sequence of ID's

$$(q_0, u_0, \gamma_0) \vdash_A \cdots \vdash_A (q_k, u_k, \gamma_k),$$

with $u_0 = ¢w\$$, $\gamma_0 = Z_0\uparrow$, and $u_k = \epsilon$; or

(2) there exists an infinite sequence of ID's

$$(q_0, u_0, \gamma_0) \vdash_A \cdots \vdash_A (q_i, u_i, \gamma_i) \vdash_A \cdots,$$

with $u_0 = ¢w\$$ and $\gamma_0 = Z_0\uparrow$.

For each i, $i \neq k$ in (1), let $\gamma_i = \alpha_i Y_i\uparrow\alpha_i'$ and $u_i = a_i u_i'$, with Y_i in Γ and a_i in $\Sigma \cup \{¢, \$\}$.

Suppose (1) holds. Consider i, $i \neq k$. If $\delta(q_i, a_i, Y_i)$ satisfies (ii), then $\delta_B(q_i, a_i, Y_i) = \delta(q_i, a_i, Y_i)$. Suppose $\delta(q_i, a_i, Y_i)$ satisfies (iii). Since $u_k = \epsilon$, there exists a smallest integer $f(i)$ such that $u_{f(i)} = u_i'$. Consider α_j, $i \leq j < f(i)$. If there exists some j such that $\alpha_j Y_j = \alpha_i$, then either $(q_i, a_i, Y_i\uparrow) \vdash_A^* (q', a_i, \uparrow)$ for some q' or $(q_i, a_i, Y_i\uparrow) \vdash_A^* (q', a_i, \uparrow y)$ for some q' and $y \neq \epsilon$. Suppose there is no j such that $\alpha_j Y_j = \alpha_i$. Then $(q_i, a_i, Y_i\uparrow) \vdash_A^* (q_{f(i)}, \epsilon, y_1\uparrow y_2)$ for some y_1 and y_2. In any case, $\delta_B(q_i, a_i, Y_i) = \delta(q_i, a_i, Y_i)$. Thus

$$(q_0, u_0, \gamma_0) \vdash_B \cdots \vdash_B (q_k, u_k, \gamma_k).$$

To prove that B satisfies the conclusion of the lemma, it thus suffices to show that if (2) holds, then $(q_0, u_0, \gamma_0) \vdash_B^* (h, \epsilon, \gamma)$ for some γ. Since either $\delta_B(q, a, Y) = \delta(q, a, Y)$ or $\delta_B(q, a, Y) = (1, h, 0, Y)$, it suffices to show that for some t, $(q_t, u_t, \gamma_t) \vdash_B (q_{t+1}, u_{t+1}, \gamma_{t+1})$ is false. Assume (2) holds and there is no i such that $\delta(q_i, a_i, Y_i)$ satisfies (iii-d). Since $u_0, u_1, \cdots$ is an infinite sequence, there exists a smallest integer s such that $u_i = u_s$ for all $i \geq s$. Hence there is no $i \geq s$ such that $\delta(q_i, a_s, Y_i)$ satisfies (iii-b).

Suppose there is some $i \geq s$ such that $\delta(q_i, a_s, Y_i)$ satisfies (iii-c). Then there exists $r \geq s$ such that $\delta(q_r, a_s, Y_r) = (0, q_{r+1}, -1, Y_r)$. Thus

(3) $(q_r, u_s, \alpha_r Y_r | \alpha_r') \vdash_A (q_{r+1}, u_s, \alpha_r | Y_r \alpha_r')$.

Assume there exists an integer, thus a smallest integer, $l > r$ such that $\gamma_r = \gamma_l$. Then $(q_r, u_s, \alpha_r Y_r \uparrow) \vdash_A^{*S} (q_l, u_s, \alpha_r Y_r \uparrow)$. Since A is directed, $(q_r, u_s, \alpha_r Y_r \uparrow) \vdash_A^{S} (q_l, u_s, \alpha_r Y_r \uparrow)$, contradicting (3). Thus there is no integer $i > r$ such that $\gamma_i = \gamma_r$. Hence $\alpha_i Y_i$ is an initial subword of α_r for each $i > r$. Since there are only a finite number of initial subwords of α_r, there exists a largest initial subword α of α_r with the following property:

(4) There exists an infinite sequence $i(1), \cdots, i(j), \cdots, \; r < i(1) < i(2) < \cdots$ such that $\alpha = \alpha_{i(j)} Y_{i(j)}$, for each j.

Then for some m and n, $m < n$, $q_{i(m)} = q_{i(n)}$. Hence $(q_{i(m)}, u_s, \alpha_{i(m)} Y_{i(m)} \uparrow) \vdash_A^* (q_{i(n)}, u_s, \alpha_{i(m)} Y_{i(m)} \uparrow)$. From the fact that α is the largest initial subword of α_r satisfying (4),

$$(q_{i(m)}, u_s, \alpha_{i(m)} Y_{i(m)} \uparrow) \vdash_A^{*S} (q_{i(n)}, u_s, \alpha_{i(m)} Y_{i(m)} \uparrow).$$

This contradicts A being directed since $q_{i(m)} = q_{i(n)}$. Hence there is no $i \geq s$ such that $\delta(q_i, a_s, Y_i \uparrow)$ satisfies (iii-c).

In view of the above, $\delta(q_i, a_s, Y_i)$ satisfies (ii-b), or (ii-c), or (iii-a) for each $i \geq s$. Now $\delta(q_i, a_s, Y_i)$ cannot satisfy (ii-b) for every $i \geq s$. For otherwise, A is merely scanning the stack γ_s and a contradiction arises by the argument in the preceding paragraph. Let $s(1)$ be the smallest integer greater than or equal to s such that $\delta(q_{s(1)}, a_s, Y_{s(1)})$ satisfies (ii-c) or (iii-a). Either alternative results in some $\gamma_{s(2)}$, $s(2) > s(1)$, for which $|\gamma_{s(2)}| < |\gamma_{s(1)}|$. Repeating the procedure, we get $s(4), s(6), \cdots$, such that $s(4) < s(6) < \cdots$ and $|\gamma_{s(4)}| > |\gamma_{s(6)}| > \cdots$. This is a contradiction since $|\gamma_s|$ is finite. We are thus forced to conclude that there exists some i such that $\delta(q_1, a_i, Y_i)$ satisfies (iii-d), thereby proving the lemma.

Remark. Given (q, a, Z), we can construct, although not done here, one-way sa C_1, C_2, and C_3 (depending on (q, a, Z)) such that (1) a is in $T(C_1)$ if and only if $\delta(q, a, Z)$ satisfies (iii-a); (2) a is in $T(C_2)$ if and only if $\delta(q, a, Z)$ satisfies (iii-b); and (3) a is in $T(C_3)$ if and only if $\delta(q, a, Z)$ satisfies (iii-c). Since languages are recursive sets [5], the construction of B in Lemma 4.5 is effective.

We are finally ready for the complementation result.

THEOREM 4.1. *The complement of a D-language is a D-language.*

PROOF. Let $L \subseteq \Sigma^*$ be a D-language. By Lemma 4.5, $L = T(A)$ for some loop-free one-way sa $A = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$. Let $B = (K, \Sigma, ¢, \$, \Gamma, \delta, q_0, Z_0, K - F)$. Since A is loop-free and the device is deterministic, obviously $T(B) = \Sigma^* - T(A)$.

COROLLARY. *For each D-language L and regular set R, $R \cup L$ and $R - L$ are D-languages.*

PROOF. Since $L \cup R = \Sigma^* - [(\Sigma^* - L) \cap (\Sigma^* - R)]$ and $R - L = R \cap (\Sigma^* - L)$, the result follows from Corollary 2 to Theorem 2.1 and from Theorem 4.1.

Another important operation which preserves D-languages is the inverse gsm mapping.

THEOREM 4.2. *Let L be a D-language and $S = (K_S, \Sigma \cup \{h\}, \Delta, \delta_S, \lambda, p_0)$ a gsm with h not in Σ. Then $L' = \{w \text{ in } \Sigma^* \mid S(wh) \text{ in } L\}$ is a D-language.*

PROOF. Let $L = T(A)$, where $A = (K, \Delta, ¢, \$, \Gamma, \delta, q_0, Z_0, F)$. Let

$$n = \max \{|\lambda(p, a)| \mid (p, a) \quad \text{in} \quad K_S \times (\Sigma \cup \{h\})\}$$

and let $\Delta_n = \bigcup_{i=0}^{n} \Delta^i$. Let 0 be a symbol not in K_S and r a symbol not in $K \cup (K \times (K_S \cup \{0\}) \times \Delta_n)$. Let $B = (K_B, \Sigma \cup \{h\}, ¢, \$, \Gamma, \delta_B, q_0, Z_0, F_B)$, where $K_B = K \cup \{r\} \cup (K \times (K_S \cup \{0\}) \times \Delta_n)$, $F_B = F \times \{0\} \times \{\epsilon\}$, and δ_B is defined as follows for (Z, p, q, b, x) in $\Gamma \times K_S \times K \times \Delta \times \Delta_{n-1}$:

(i) If $\delta(q, ¢, Z) = (d, q', e, y)$, then
  (a) $\delta_B(q, ¢, Z) = (d, q', e, y)$ if $d = 0$.
  (b) $\delta_B(q, ¢, Z) = (d, (q', p_0, \epsilon), e, y)$ if $d = 1$.

(ii) For a in Σ, let
  (a) $\delta_B((q, p, \epsilon), a, Z) = (1, (q, \delta_S(p, a), \lambda(p, a)), 0, Z)$.

(iii) If a is in $\Sigma \cup \{\$\}$ and $\delta(q, b, Z) = (d, q', e, y)$, then
  (a) $\delta_B((q, p, bx), a, Z) = (0, (q', p, bx), e, y)$ if $d = 0$.
  (b) $\delta_B((q, p, bx), a, Z) = (0, (q', p, x), e, y)$ if $d = 1$.

(iv) $\delta_B((q, p, \epsilon), \$, Z) = (0, (q, 0, \lambda(p, h)), 0, Z)$.

(v) For $\delta(q, b, Z) = (d, q', e, y)$,
  (a) $\delta_B((q, 0, bx), \$, Z) = (0, (q', 0, bx), e, y)$ if $d = 0$.
  (b) $\delta_B((q, 0, bx), \$, Z) = (0, (q', 0, x), e, y)$ if $d = 1$.

(vi) For $\delta(q, \$, Z) = (d, q', e, y)$,
  $\delta_B((q, 0, \epsilon), \$, Z) = (d, (q', 0, \epsilon), e, y)$.

(vii) For all other s in K_B, a in $\Sigma \cup \{h, ¢, \$\}$, and Z in Γ,
  $\delta_B(s, a, Z) = (1, r, 0, Z)$.

Clearly B is deterministic. Intuitively, B imitates A on ¢ by (i), simulates A on $\lambda(p, a)$ by (ii) and (iii), simulates A on $\lambda(p, h)$ by (iv) and (v), and imitates A on \$ by (vi). Formally, the construction does the following. By (i),

$$(q_0, ¢, Z_0\uparrow) \vdash_B^* ((q, p_0, \epsilon), \epsilon, y_1\uparrow y_2)$$

if and only if

$$(q_0, ¢, Z_0\uparrow) \vdash_A^* (q, \epsilon, y_1\uparrow y_2).$$

By (ii),

$$((q, p, \epsilon), a, y_1\uparrow y_2) \vdash_B ((q, \delta_S(p, a), \lambda(p, a)), \epsilon, y_1\uparrow y_2).$$

By (iii),

$$((q, p, u), a, y_1\uparrow y_2) \vdash_B^* ((q', p, \epsilon), a, y_1'\uparrow y_2')$$

if and only if

$$(q, u, y_1\uparrow y_2) \vdash_A^* (q', \epsilon, y_1\uparrow y_2).$$

By (iv) and (v),

$$\begin{aligned}((q, p, \epsilon), \$, y_1\uparrow y_2) &\vdash_B ((q, 0, \lambda(p, h)), \$, y_1\uparrow y_2)\\ &\vdash_B^* ((q', 0, \epsilon), \$, y_1'\uparrow y_2')\\ &\vdash_B^* ((q'', 0, \epsilon), \epsilon, y_1''\uparrow y_2'')\end{aligned}$$

if and only if

$$(q, \lambda(p, h)\$, y_1\uparrow y_2) \vdash_A^* (q', \$, y_1'\uparrow y_2') \vdash_A^* (q'', \epsilon, y_1''\uparrow y_2'').$$

Then for $a_1 \cdots a_k$, $k \geq 0$, each a_i in Σ, we have

$$\begin{aligned}(q_0, ¢a_1 \cdots a_k\$, Z_0\uparrow) &\vdash_B^* ((q_1, p_0, \epsilon), a_1 \cdots a_k\$, \gamma_1)\\ &\vdash_B ((q_1, p_1, w_1), a_2 \cdots a_k\$, \gamma_1)\\ &\vdash_B^* ((q_2, p_2, w_2), a_3 \cdots a_k\$, \gamma_2)\\ &\cdots\\ &\vdash_B^* ((q_k, p_k, w_k), \$, \gamma_k)\\ &\vdash_B ((q_{k+1}, 0, w_{k+1}), \$, \gamma_{k+1})\\ &\vdash_B^* ((q_{k+2}, 0, \epsilon), \$, \gamma_{k+2})\\ &\vdash_B^* ((q_{k+3}, 0, \epsilon), \epsilon, \gamma_{k+3})\end{aligned}$$

if and only if $\delta_S(p_i, a_{i+1}) = p_{i+1}$ and $\lambda(p_i, a_{i+1}) = w_{i+1}$ for each i, $0 \leq i < k$, $\lambda(p_k, h) = w_{k+1}$ (so that $S(a_1 \cdots a_k h) = w_1 \cdots w_{k+1}$), and

$$(q_0, ¢w_1 \cdots w_{k+1}\$, Z_0\uparrow) \vdash_A^* (q_1, w_1 \cdots w_{k+1}\$, \gamma_1),$$

$$(q_i, w_i \cdots w_{k+1}\$, \gamma_i) \vdash_A^* (q_{i+1}, w_{i+1} \cdots w_{k+1}\$, \gamma_{i+1}) \quad \text{for} \quad 1 \leq i \leq k+1,$$

and

$$(q_{k+2}, \$, \gamma_{k+2}) \vdash_A^* (q_{k+3}, \epsilon, \gamma_{k+3}).$$

Hence

$$(q_0, ¢a_1 \cdots a_k\$, Z_0\uparrow) \vdash_B^* ((q, 0, \epsilon), \epsilon, u\uparrow v)$$

if and only if

$$(q_0, ¢S(a_1 \cdots a_k h)\$, Z_0\uparrow) \vdash_A^* (q, \epsilon, u\uparrow v).$$

Thus w is in $T(B)$ if and only if $S(wh)$ is in $T(A)$. Therefore

$$T(B) = \{w \text{ in } \Sigma^* \mid S(wh) \text{ in } L\}.$$

Corollary 1. *For each D-language L and each gsm S,*

$$S^{-1}(L) = \{w \mid S(w) \text{ in } L\}$$

is a D-language.

Proof. Let $S = (K, \Sigma, \Delta, \delta, \lambda, p_0)$. Let h be a symbol not in K. Let S_1 be the gsm $(K, \Sigma \cup \{h\}, \Delta, \delta_1, \lambda_1, p_0)$, where $\delta_1(p, h) = p$ and $\lambda_1(p, h) = \epsilon$ for each p in K, and $\delta_1 = \delta$ and $\lambda_1 = \lambda$ otherwise. Then

$$\begin{aligned}S^{-1}(L) &= \{w \text{ in } \Sigma^* \mid S(w) \text{ in } L\}\\ &= \{w \text{ in } \Sigma^* \mid S_1(wh) \text{ in } L\}\end{aligned}$$

is a D-language by Theorem 4.2.

Corollary 2. *Let w be a word in Σ^*. If L is a D-language, then $\{x \mid wx \text{ in } L\}$ is a D-language.*

PROOF. Let $L_1 = L - \{w\}$. By Corollary 2 of Theorem 2.1, L_1 is a D-language. Let $S = (\{p_0, p_1\}, \Sigma, \Sigma, \delta, \lambda, p_0)$ be the gsm where $\delta(p_0, b) = \delta(p_1, b) = p_1$, $\lambda(p_0, b) = wb$, and $\lambda(p_1, b) = b$ for each b in Σ. By Corollary 1 above, $S^{-1}(L_1)$ is a D-language. Then $B_w(L) = S^{-1}(L_1)$ if L does not contain w, and $B_w(L) = S^{-1}(L_1) \cup \{\epsilon\}$ if L does contain w, where $B_w(L) = \{x \mid wx \text{ in } L\}$. Thus $B_w(L)$ is a D-language.

COROLLARY 3. *Let w be a word in Σ^*. If L is a D-language, then $\{x \mid xw \text{ in } L\}$ is also a D-language.*

PROOF. Let h be a symbol not in Σ. Let S be the gsm $(\{p_0\}, \Sigma \cup \{h\}, \Sigma, \delta, \lambda, p_0)$, where $\delta(p_0, b) = \delta(p_0, h) = p_0$, $\lambda(p_0, b) = b$, and $\lambda(p_0, h) = w$ for each b in Σ. Then $\{x \mid xw \text{ in } L\} = \{x \text{ in } \Sigma^* \mid S(xh) \text{ in } L\}$ is a D-language by Theorem 4.2.

A number of operations which do not preserve D-languages are now presented.

THEOREM 4.3. *The family of D-languages is not closed under* (i) *union,* (ii) *product,* (iii) $*$, *and* (iv) *homomorphism.*[30]

PROOF. Let L_1 and L_2 be two D-languages such that $L_1 \cap L_2$ is not a D-language. By Theorem 3.2, L_1 and L_2 exist. Let Σ_0 be the alphabet over which L_1 and L_2 are defined. Let g be a new symbol. Let $\bar{L}_1 = \Sigma_0^* - L_1$ and $\bar{L}_2 = \Sigma_0^* - L_2$.

As to (i), suppose $\bar{L}_1 \cup \bar{L}_2$ is a D-language. Then $L_1 \cap L_2 = \Sigma_0^* - (\bar{L}_1 \cup \bar{L}_2)$ is a D-language, a contradiction.

Consider (ii). Clearly $\bar{L}_1 \cup g\bar{L}_2$ is a D-language. Also, $\{g, g^2\}$ is a D-language. Suppose D-languages are closed under product. Then

$$\begin{aligned} L' &= \{g, g^2\}(\bar{L}_1 \cup g\bar{L}_2) \\ &= g^2(\bar{L}_1 \cup \bar{L}_2) \cup g\bar{L}_1 \cup g^3\bar{L}_2 \end{aligned}$$

is a D-language. By the corollary to Theorem 2.3, $L' \cap g^2\Sigma_0^* = g^2(\bar{L}_1 \cup \bar{L}_2)$ is a D-language. By Corollary 2 of Theorem 4.2, $\bar{L}_1 \cup \bar{L}_2$ is a D-language, a contradiction.

Consider (iii). Suppose D-languages are closed under $*$. Then $L'^* \cap g^2\Sigma_0\Sigma_0^* = g^2(\bar{L}_1 \cup \bar{L}_2)$ is a D-language. By Corollary 2 of Theorem 4.2, $\bar{L}_1 \cup \bar{L}_2$ is a D-language, a contradiction.

Consider (iv). As noted above, $\bar{L}_1 \cup g\bar{L}_2$ is a D-language. Let h be the homomorphism defined by $h(a) = a$ for a in Σ_0 and $h(g) = \epsilon$. Then $h(\bar{L}_1 \cup g\bar{L}_2) = \bar{L}_1 \cup \bar{L}_2$ is a D-language, a contradiction.

5. *Decision Problems*

We now consider the decidability of various questions. We use the fact that all constructions given so far can be made effective. We also use the fact that a language is recursive [5], i.e., it is recursively solvable to determine if an arbitrary word is in an arbitrary language.

Turning to solvability results, we have

THEOREM 5.1. *It is recursively solvable to determine whether $T(A)$ is empty for an arbitrary one-way sa A.*

PROOF. Let A be a one-way sa over Σ. Let h be the homomorphism defined by $h(a) = \epsilon$ for each a in Σ. By the corollary to Theorem 2.6, $h(T(A))$ is a language. Now $T(A) = \phi$ if and only if ϵ is not in $h(T(A))$. Since $h(T(A))$ is a language, it is recursive. Thus it is decidable if ϵ is in $h(T(A))$.

[30] (iv) implies nonclosure under gsm mappings.

The next solvability result concerns D-languages.

THEOREM 5.2. *It is recursively solvable to determine for an arbitrary D-language L and a regular set R whether $L = R$.*

PROOF. Since L is a D-language and R is regular, $\Sigma^* - L$ and $(\Sigma^* - L) \cap R$ are D-languages. Thus $L' = [L \cap (\Sigma^* - R)] \cup [(\Sigma^* - L) \cap R]$ is a language. Now $L = R$ if and only if $L' = \phi$, which is solvable by Theorem 5.1.

Turning to unsolvability results, we have

THEOREM 5.3. *It is recursively unsolvable to determine whether an arbitrary language is (a) regular, (b) context free, (c) a D-language.*

PROOF. (a) is known in that it is recursively unsolvable to determine whether an arbitrary context-free language is regular [1].

Consider (b). Let $\Sigma_1 \cap \Sigma_2 = \phi$ and let $M_2 \subseteq \Sigma_2^*$ be a language which is not context free, say $\{a^i b^i c^i \mid i \geq 1\}$. For each context-free language $M \subseteq \Sigma_1^*$, let $L(M)$ be the language $M\Sigma_2^* \cup \Sigma_1^* M_2$. Since it is unsolvable to determine if an arbitrary context-free language is Σ_1^* [1], it suffices to show that

(1) $L(M)$ is context free if and only if $M = \Sigma_1^*$.

Thus suppose that $M = \Sigma_1^*$. Then $L(M) = \Sigma_1^*\Sigma_2^*$ is regular, hence context free. Suppose $M \neq \Sigma_1^*$ and $L(M)$ is context free. Let w be some word in $\Sigma_1^* - M$. Then $L(M) \cap w\Sigma_2^* = wM_2$ is context free. Therefore M_2 is context free, a contradiction. Thus (1) is justified.

To prove (c), let M_3 be a language which is not a D-language. By the corollary to Theorem 3.2, M_3 exists. For each context-free language $M \subseteq \Sigma_1^*$, let $H(M)$ be the language $M\Sigma_2^* \cup \Sigma_1^* M_3$. It suffices to show that

(2) $H(M)$ is a D-language if and only if $M = \Sigma_1^*$.

Suppose $M = \Sigma_1^*$. Then $H(M) = \Sigma_1^*\Sigma_2^*$ is regular, thus a D-language. Suppose $M \neq \Sigma_1^*$ and $H(M)$ is a D-language. Let w be in $\Sigma_1^* - M$. Then $H(M) \cap w\Sigma_2^* = wM_3$ is a D-language. By Corollary 2 of Theorem 4.2, M_3 is a D-language, a contradiction. Thus (2) is justified.

We close with the following open problem:

($*$) Is it recursively solvable to determine for an arbitrary one-way sa A whether $T(A)$ is finite?

Given an arbitrary language $L \subseteq \Sigma^*$ and a fixed element a in Σ, let h be the homomorphism defined by $h(b) = a$ for each b in Σ. Then L is finite if and only if $h(L)$ is finite. Let $M(L) = \text{Init}(h(L))$. By the corollary to Theorem 2.7, $M(L)$ is a language. Clearly L is finite if and only if $M(L)$ is finite. $M(L)$ has the following interesting properties: (i) It is regular; (ii) it is finite if and only if it does not coincide with a^*, i.e., if and only if $a^* - M(L) = \phi$. Thus ($*$) can be reduced to the following problem:

($**$) Is it recursively solvable to determine whether $T(A)$ is finite for an arbitrary one-way sa A, over a one-letter alphabet, with the property that $\text{Init}(T(A)) = T(A)$?

ACKNOWLEDGMENT. The authors wish to thank Professor Donald Knuth for a critical reading of the manuscript.

REFERENCES

1. BAR-HILLEL, Y., PERLES, M., AND SHAMIR, E. On formal properties of simple phrase structure grammars. *Z. Phonetik, Sprachwissen. Kommunikationsforsch.* *14* (1961), 143–172; in

Y. Bar-Hillel, *Language and Information*, Addison-Wesley, Reading, Mass., 1965, pp. 116–150.

2. Chomsky, N. On certain formal properties of grammars. *Inform. and Contr. 2* (1959), 137–167.
3. Davis, M. *Computability and Unsolvability*. McGraw-Hill, New York, 1958.
4. Ginsburg, S., and Greibach, S. Deterministic context free languages. *Inform. and Contr. 9* (1966), 620–648.
5. Ginsburg, S., Greibach, S., and Harrison, M. Stack automata and compiling. *J. ACM 14* (1967), 172–201.
6. Rabin, M., and Scott, D. Finite automata and their decision problems. *IBM J. Res. Develop. 3* (1959), 114–125.

RECEIVED MAY, 1966; REVISED AUGUST, 1966