Multi-Tape and Multi-Head Pushdown Automata*

Michael A. Harrison and Oscar H. Ibarra

University of California, Berkeley, California 94720

This paper considers multi-tape and multi-head extensions of various models of pushdown automata. One-way and two-way deterministic and nondeterministic multi-tape and multi-head pushdown automata are introduced and studied. The closure, characterization, and decision properties of the sets definable by these automata are investigated and the relationship between these sets and some well known families of languages is established.

INTRODUCTION

The class of context free languages is well known [2] to be coextensive with the class of languages accepted by pushdown automata. Other families of languages which are richer than the context free languages have been introduced and studied such as the context sensitive languages ([14], [15]). The families of languages between context free and context sensitive are of current interest also ([8], [9], [12], [14], [15]). The present study resulted from an investigation into pushdown automata which have several reading heads on the input. (This corresponds to the "look ahead" features in machines). It turns out that such a model is closely connected with pushdown automata operating on n-tapes. The present paper is taken from [13] and gives the basic theory of multi-head and multi-tape pushdown automata.

It is hoped that such a theory will shed additional light on the special case n = 1 as has happened for n-tape finite automata through the work of Elgot and Mezei [5] and Rosenberg [20].

The present paper is divided into 6 sections and an appendix. Section 1 contains the basic models and a fundamental theorem which relates n-head and n-tape pushdown automata. Section 2 relates n-tape push-

* Research sponsored by the Air Force Office of Scientific Research, Office of Aerospace Research, United States Air Force, Grant No. AF-AFOSR-639-65 and by the National Science Foundation under Grant GP-6945.
down automata with context free languages over the direct product of free monoids. Some important projection theorems are given.

Section 3 contains a proof that for an alphabet of one letter, the sets defined by $n$-tape pushdown automata coincide with the sets accepted by $n$-tape finite automata. In Section 4, it is shown that the sets accepted by $n$-head (deterministic) pushdown automata are accepted by (deterministic) linear bounded automata. Section 5 contains numerous closure properties of both families (the appendix contains the proof of the closure of the sets definable by $n$-tape deterministic pushdown automata under complementation) while Section 6 briefly mentions decision problems.

SECTION 1. BASIC PROPERTIES OF THE MODELS

In this section the models are introduced and a number of basic properties are established. An important theorem is proven which relates the families of devices.

Intuitively, an $n$-tape pushdown automaton ($n$-TPDA, for short) is a device with an $n$-tuple of finite input tapes (each tape is provided with endmarkers and an independent reading head), a finite number of internal states, and a "last-in-first-out" pushdown storage tape. In addition, an $n$-TPDA has an initial state, an initial pushdown symbol, and a set of final states. Initially, the $n$-TPDA is set to its initial state, with the initial pushdown symbol written on its pushdown storage tape, and each reading head positioned on the left endmarker of its corresponding input tape. In general, the $n$-TPDA is in a state, has a rightmost symbol on the pushdown storage tape and a reading head is scanning an input symbol on its tape channel. In a nondeterministic manner, the device simultaneously goes to another state, moves the reading head right or remains stationary on its tape channel, and rewrites the topmost symbol of the pushdown tape by some word. (If this word is empty, the top symbol is "erased.") The process is continued until one reading head leaves its tape, at which time the $n$-TPDA halts. More precisely, we have the following definition.

**Definition.** An $n$-tape pushdown automaton ($n$-TPDA, for short) is a 10-tuple $A = \langle S, \Sigma, \epsilon, \$,$ \Gamma,$ $M,$ $s_0,$ $\gamma_0,$ $F \rangle,$ where

1. $S$ is a finite nonempty set (of states),
2. $\Sigma$ is an alphabet (of inputs),
3. $\epsilon$ and $\$ are two elements not in $\Sigma$ (the left and right endmarkers, for each of the $n$ tapes).
MULTI-TAPE AND MULTI-HEAD PUSHDOWN AUTOMATA

(4) \( \Gamma \) is a finite nonempty set (of pushdown symbols),
(5) \( M \) is a mapping from \( S \times (\Sigma \cup \{ \epsilon, \$ \}) \times \Gamma \) into the finite subsets of \( \{0, 1\} \times S \times \Gamma^* \),
(6) \( \delta \) is a mapping from \( S \) into \( N_n = \{1, \cdots, n\} \) and is called the tape selector function. We indicate this by writing \( \delta : S \rightarrow N_n \).
(7) \( s_0 \in S \) (the start state),
(8) \( \gamma_0 \in \Gamma \) (the initial pushdown symbol),
(9) \( F \subseteq S \) (the set of final states).

The input to the \( n \)-TPDA \( A \) is an \( n \)-tuple \((x_1, \cdots, x_n)\), where each \( x_i \in \Sigma^* \). The function of the endmarkers \( \epsilon \) and \$ is to let a reading head know when it is at the beginning or end of its corresponding input channel. The symbolism \((d, s', w) \in M(s, \sigma, \gamma)\) means the following. If the \( n \)-TPDA \( A \) is in a state \( s \), and some reading head is scanning a symbol \( \sigma \) on its input channel (determined by tape selector function \( \delta \)), and \( \gamma \) is the rightmost symbol on the pushdown store, then this reading head moves right on its input channel if \( d = 1 \), and does not move if \( d = 0 \). \( A \) goes to state \( s' \), and \( w \) is written in place of \( \gamma \) on the pushdown store. We write \((d, s', w) \in M(s, \sigma, \gamma)\) to indicate that \( A \) has many "choices," that is, \( A \) is a "nondeterministic" automaton.

We now formalize these concepts.

**Definition.** Let \( A = (S, \Sigma, \epsilon, $, \Gamma, \delta, s_0, \gamma_0, F) \) be an \( n \)-TPDA, and let \( P = \{p_i | 1 \leq i \leq n\} \), \( P \cap (\Sigma \cup \{ \epsilon, \$ \}) = \emptyset \). An instantaneous description (abbreviated ID) of \( A \) is any element \(^1\) of

\[
S \times (\Sigma \cup \{ \epsilon, \$ \} \cup P)^* \times \Gamma^*.
\]

\(^1\) Let \( X, Y \) be set of words. \( XY = \{xy | x \in X, y \in Y\} \) where \( xy \) is the concatenation of \( x \) and \( y \). Let \( X^0 = \{\epsilon\} \), where \( \epsilon \) is the empty word. For \( i \geq 0 \), let \( X^{i+1} = X^i X \) and \( X^* = \bigcup_{i \geq 0} X^i \). \( \emptyset \) denotes the empty set.

\(^2\) An \( n \)-tuple of tapes over \( \Sigma \) is an \( n \)-tuple \((x_1, \cdots, x_n)\) where each \( x_i \) is a tape over \( \Sigma \). We make no distinction between a \( 1 \)-tuple of tape and a tape. Let \((x_1, \cdots, x_n)\) and \((y_1, \cdots, y_n)\) be \( n \)-tuples of tapes. The concatenation or product of \((x_1, \cdots, x_n)\) and \((y_1, \cdots, y_n)\) is the \( n \)-tuple \((x_1, \cdots, x_n)(y_1, \cdots, y_n) = (x_1y_1, \cdots, x_ny_n)\). \( XY \) and \( X^* \) are defined in the same manner as before when \( X \) and \( Y \) are sets of \( n \)-tuples of words except that now concatenation is component-wise.

\(^3\) The cross-product of \( A \) and \( C \) is the set of \((n + m)\)-tuples of tapes \( A \times C = \{(x_1, \cdots, x_n, y_1, \cdots, y_m) | (x_1, \cdots, x_n) \in A \text{ and } (y_1, \cdots, y_m) \in C\} \). \( |A|^{(l \geq 1)} \) is defined inductively as follows: \( |A|^1 = A \) and \( |A|^{i+1} = A \times |A|^i \). We note that \( \Sigma^* = \Sigma^* \times \cdots \times \Sigma^* \) (\( n \) times) is the set of all \( n \)-tuples of tapes over \( \Sigma \). We write \( \Sigma^* \) for \( |\Sigma| \). Let \( \Delta_n = \{(x_1, \cdots, x) | x \in \Sigma^*\} \) be the diagonal relation on \( \Sigma^* \). For any \( X \subseteq \Sigma^* \) and \( 1 < i < n \), define \( P_{(i)}(X) = \{x_i | (x_1, \cdots, x_n) \in X \text{ for some } x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n \in \Sigma^*\} \).
The ID \((s, x_1p_1y_1, \ldots, x_np_ny_n, w)\), denotes the fact that \(A\) is in state \(s\), with \(x_1 \in \Sigma^{*}\$ (1 \leq i \leq n)\) the inputs (with \(p_i\) indicating the position of the \(i\)th reading head), and \(w \in \Gamma^{*}\) on the pushdown.

**Definition.** Given an \(n\)-TPDA \(A = (S, \Sigma, \delta, \$)\), \(M, \delta, s_0, \gamma_0, F)\), let \(\vdash A\) or \(\vdash\) when \(A\) is understood, be the relation between ID’s defined as follows: For each \(k \geq 2, 1 \leq j \leq k, 1 \leq i \leq n\) let \((s, x_1p_1y_1, \ldots, s_1 \cdots p_i \sigma_j \cdots s_k, \ldots, x_np_ny_n, w)\) \(\vdash (s, x_1p_1y_1, \ldots, s_1 \cdots p_i \sigma_j \cdots s_k, w')\) if the following conditions are satisfied:

1. \(\delta(s) = i\),
2. \(\sigma_1, \ldots, \sigma_k \in \Sigma \cup \{\$, \\$\}, \sigma_{k+1} = \Lambda\),
3. \((d, s', w') \in M(s, \sigma_j, \gamma)\) for some \((s, \sigma_j, \gamma) \in S \times (\Sigma \cup \{\$, \\$\}) \times \Gamma\).

The convention \(\sigma_{k+1} = \Lambda\) allows the \(i\)th reading head to leave the right end of its input channel.

The notation for describing a sequence of movements of \(A\) is now presented.

**Definition.** Let \(A = (S, \Sigma, \delta, \$)\), \(M, \delta, s_0, \gamma_0, F)\) be an \(n\)-TPDA. Define \(\vdash A^*\) or \(\vdash^*\) when \(A\) is understood as follows: For ID's \(a\) and \(b\) of \(A\), write \(a \vdash^* b\) if there exist \(r > 0, \) ID's \(a_0, \ldots, a_r\) such that \(a_0 = a, a_r \vdash a_{r+1}\) for \(0 \leq i \leq r\).

We now define acceptance of an \(n\)-tuple of tapes.

**Definition:** An \(n\)-tuple of tapes \((z_1, \ldots, z_n) \in [\Sigma^{*}]^n\) is accepted by an \(n\)-TPDA \(A = (S, \Sigma, \delta, \$)\), \(M, \delta, s_0, \gamma_0, F)\) if \((s_0, p_1z_1\$, \ldots, p_nz_n\$, \gamma_0) \vdash^* (s, x_1p_1y_1, \ldots, \$z_1\$p_1, \ldots, x_np_ny_n, w)\) for some \(1 \leq i \leq n, s \in F, \) and \(w \in \Gamma^{*}\). The set of \(n\)-tuples of tapes accepted by \(A\) is denoted by \(T(A)\).

Some of the results that we shall derive hold for a more general class of \(n\)-TPDA, thus we introduce the following definition.

**Definition.** An \(n\)-tape two-way pushdown automaton \((n\)-TTWPDA, for short\) is a 10-tuple \(A = (S, \Sigma, \delta, \$)\), \(M, \delta, s_0, \gamma_0, F)\), where \(S, \Sigma, \delta, \$)\), \(\Gamma, \delta, s_0, \gamma_0, F)\) are as in an \(n\)-TPDA and \(M\) is a mapping from \(S \times (\Sigma \cup \{\$, \\$\}) \times \Gamma\) into the finite subsets of \([-1, 0, 1] \times S \times \Gamma^{*}\).

Here we allow the reading heads of an \(n\)-TPDA to move both ways on their corresponding tape channels. We assume without loss of generality that the reading heads are prevented from going off the left end of their inputs.
The notions of ID's, computation, and acceptance of an n-tuple of tapes are similar to those given for an n-TPDA.

We shall also consider the following special classes of n-TPDA and n-TTWDPA.

**Definition:** An n-TPDA (n-TTWPDA) $A = \langle S, \Sigma, \delta, \gamma_0, F, M, \delta, \delta_0, \gamma \rangle$ is said to be **deterministic** if $|M(s, \sigma, \gamma)| \leq 1$ for each $(s, \sigma, \gamma) \in S \times (\Sigma \cup \{\delta, \gamma\}) \times \Gamma$. We write n-TDPDA (n-TTWDPDA) as an abbreviation for a deterministic n-TPDA (n-TTWPDA). In the deterministic case, we may write $M(s, \sigma, \gamma) = (d, s', w)$ instead of $M(s, \sigma, \gamma) = \{(d, s', w)\}$.

**Definition.** A subset $L \subseteq [\Sigma^n]^n$ is n-TPDA (n-TPDA) definable if there exists some n-TPDA (n-TPDA) n-TPDPA (n-TPDPA) $A$ such that $T(A) = L$.

**Remark.** It can easily be shown that the class of sets definable by n-TPDA (n-TPDA) n-TPDPA (n-TPDPA) is unchanged if we define the tape selector function to have as its domain either $\Gamma$ or $S \times \Gamma$ instead of $S$ and its range $2^n$ instead of $N^n$. We choose $S$ and $N^n$, however, for convenience. Further details may be found in [13]. One may also show that the left endmarker can be removed without reducing the computing capability of an n-TDPA (n-TPDA), and that this is not true for right endmarkers. Further details and documentation of these claims are in [13].

The following special case of an n-TPDA is important and has been studied before ([5], [16], [20]).

**Definition.** An $n$-tape finite automaton (n-TFA, for short) is an 8-tuple $A = \langle S, \Sigma, \delta, \gamma, F \rangle$, where $S, \Sigma, \delta, \gamma, F$ have the same significance as in an n-TPDA, and $M$ is a mapping from $S \times (\Sigma \cup \{\delta, \gamma\})$ into the subsets of $\{0, 1\} \times S$. Thus, an n-TFA is an n-TPDA without a pushdown tape.

We omit formal definitions of ID, $\vdash$, $\vdash^*$ and acceptance for n-TFA's. These definitions are essentially the same as for n-TPDA's if one ignores the pushdown.

We now consider another generalization of the model of a pushdown automaton which will be shown to be closely related to n-TPDA and is our principal object of study.

An $n$-head pushdown automaton (n-HPDA, for short) has n reading
heads on one input tape (provided with endmarkers). Initially, these heads are positioned on the left endmarker and the device set to its initial state and initial pushdown symbol. At each time unit, a reading head operates on a symbol on the input as in the n-TPDA. (It is assumed that the reading heads are idealized in the sense that they may pass over one another freely.) The process is continued until one reading head leaves the input tape, at which time the n-HPDA halts. We now formalize this intuitive description.

**Definition.** An n-head pushdown automaton (n-HPDA, for short) is a 10-tuple $A = \langle S, \Sigma, \delta, \$, $\Gamma, M, \nu, s_0, \gamma_0, F \rangle$, where $S, \Sigma, \delta, \$, $\Gamma, M, s_0, \gamma_0, F$ have the same significance as in an n-TPDA, and $\nu$ is a mapping from $S$ into the set $N_n$ (the head selector function).\(^5\)

**Remark.** $\nu$ could be defined as $\nu: S \rightarrow 2^N_n$, or $\nu: \Gamma \rightarrow 2^N_n$, or $\nu: S \times \Gamma \rightarrow 2^N_n$. However, as in the case of an n-TPDA, $\nu$ could be reduced to a function from $S$ into $N_n$. Furthermore, the left endmarker can be dispensed with.

We now introduce symbolism enabling us to discuss the computation of an n-HPDA.

**Definition.** Let $A = \langle S, \Sigma, \delta, \$, $\Gamma, M, \nu, s_0, \gamma_0, F \rangle$ be an n-HPDA. Let $P = \{p_i | 1 \leq i \leq n\}$, $P \cap (\Sigma \cup \{\delta, \$\}) = \emptyset$. An instantaneous description (abbreviated ID) of $A$ is any element of $S \times (\Sigma \cup \{\delta, \$\} \cup P)^* \times \Gamma^*$. The ID $(s, x_0 p_i x_2 p_i x_2 \cdots p_i x_n, w)$ with $p_i \in P (1 \leq j \leq n)$, denotes the fact that $A$ is in state $s$, with $x_0 x_1 \cdots x_n \in \Sigma*$ the input, $p_i$ being the position of $i$th reading head, and $w \in \Gamma^*$ on the pushdown.

**Definition.** Given an n-HPDA $A = \langle S, \Sigma, \delta, \$, $\Gamma, M, \nu, s_0, \gamma_0, F \rangle$, let $\vdash_A$ or $\vdash$ when $A$ is understood, be the relation between ID's defined as follows: For each $k \geq 2$, $1 \leq r \leq k$, $1 \leq j \leq n$, let $(s, x_0 p_i x_1 \cdots x_{j-1} p_i x_j \cdots p_i x_n, w)$ $\vdash (s', y_0 p_i y_1 \cdots y_{j-1} p_i y_j \cdots p_i y_n, w w')$ if the following conditions are satisfied:

1. $v(s) = i_j, 1 \leq i_j \leq n$
2. $x_0 x_1 \cdots x_n = y_0 y_1 \cdots y_n = \sigma_1 \cdots \sigma_k \in \Sigma^*$$\$$
x_0 x_1 \cdots x_{j-1} = \sigma_1 \cdots \sigma_{r-1} \quad \text{(if} \ r = 1, \ \sigma_{r-1} = \Lambda)$$\$$
x_2 x_{j+1} \cdots x_n = \sigma_r \cdots \sigma_k$$\$$
y_0 y_1 \cdots y_{j+1} = \sigma_{r+d} \cdots \sigma_k$$\$$
\sigma_{k+1} = \Lambda, w \in \Gamma^*, \quad \text{and} \quad \gamma \in \Gamma$

\(^5\) Recall that $N_n = \{1, \cdots, n\}$. 
\( (3) \) \((d, s', w') \in M(s, \sigma_r, \gamma)\) for some \((s, \sigma_r, \gamma) \in S \times (\Sigma \cup \{\epsilon, \$\}) \times \Gamma\)

The convention \(a_{k+1} = \Lambda\) allows a reading head to leave the input.

**Definition.** Let \(A = \langle S, \Sigma, \phi, \$, M, \nu, s_0, \gamma_0, F \rangle\) be an \(n\)-HPDA. Define \(\vdash_A \ast\) or \(\vdash \ast\) when \(A\) is understood as follows: For \(\alpha\)'s \(\alpha_0 \in B\) of \(A\), write \(\alpha \vdash \ast \beta\) if there exist \(r \geq 0\), \(\alpha_0, \ldots, \alpha_r\) such that \(\alpha_0 = \alpha, \alpha_r = \beta\) and \(\alpha_i \vdash \alpha_{i+1}\) for \(0 \leq i < r\).

A tape \(x \in \Sigma^*\) is accepted by \(A\) if \((e_0, p_1p_2 \cdots p_n \epsilon \$, \gamma_0) \vdash \ast (s, x_0p_{i_1}x_1 \cdots x_{n-1}p_{i_n}, \$)\) for some \(x_0x_1 \cdots x_{n-1} = \epsilon \$, p_{i_n} \in P, s \in F, \$ \in \Gamma^*\). The set of tapes accepted by \(A\) is denoted by \(T(A)\).

**Notation.** As in the case of an \(n\)-TPDA, we may define in an obvious way, the following classes:

1. \(n\)-head deterministic pushdown automaton (\(n\)-HDPDA)
2. \(n\)-head two-way pushdown automaton (\(n\)-HTWPDA)
3. \(n\)-head two-way deterministic pushdown automaton (\(n\)-HTWDPDA)

**Definition.** A subset \(L \subseteq \Sigma^*\) is \(n\)-HPDA (\(n\)-HDPDA) [\(n\)-HTWPDA] \{\(n\)-HTWDPDA\} definable if there exists some \(n\)-HPDA (\(n\)-HDPDA) [\(n\)-HTWPDA] [\(n\)-HTWDPDA] \(A\) such that \(L = T(A)\).

**Remark.** A 1-TPDA or a 1-HPDA is a pushdown automaton (abbreviated PDA)\([2], [3], [6]\). These devices accept precisely the context free languages (CFL, for short) \([2], [3], [6]\). Similarly, a 1-TDPDA or a 1-HDPDA is a deterministic pushdown automaton (= the class of PDA which accept deterministic context free languages, abbreviated det CFL) \([7], [10]\).

A 1-TTPWDA or a 1-HTWPDA is the two-way pushdown automaton studied in \([9]\). Similarly, a two-way deterministic pushdown automaton \([9]\) is an \(n\)-TTWDPDA or an \(n\)-HTWDPDA with \(n = 1\).

An \(n\)-TPDA (\(n\)-HPDA) reduces to an \(n\)-tape finite automaton (\(n\)-head finite automaton) \([5], [16], [20]\) \([18], [20]\) when the pushdown tape is not present.

We now relate the sets definable by the \(n\)-tape models with those definable by the \(n\)-head devices.

**Theorem 1.1.** A set \(L_1 \subseteq \Sigma^*\) is definable by an \(n\)-HPDA (\(n\)-HDPDA) [\(n\)-HTWPDA] [\(n\)-HTWDPDA] if and only if there exists an \(n\)-TPDA (\(n\)-TDPDA) [\(n\)-TTWPDPA] [\(n\)-TTWDPDA] definable set \(L_2 \subseteq [\Sigma^*]^n\) such that for each \(1 \leq i \leq n\), \(L_1 = P_{(i)}(L_2 \cap \Delta_n)\).
Proof. If $A = \langle S, \Sigma, \xi, \$, $\$\$, \Gamma, M, v, s_0, \gamma_0, F \rangle$ is an n-HPDA (n-HDPDA) [n-HTWPDA] [n-HTWDPDA] defining $L_1$, construct an n-TPDA (n-TDPDA) [n-TTWPDA] [n-TTWDPA] $B = \langle S, \Sigma, \xi, \$, $\$\$, \Gamma, M, \delta, s_0, \gamma_0, F \rangle$ where $\delta$ is defined as follows: For each $s \in S, \delta(s) = v(s)$. The computation of $A$ on a single tape will be simulated by $B$ on $n$ tapes. To assure that the projection on all coordinates of the defined set of $n$-tuples is the desired set, we need only intersect the defined set of $n$-tuples with $\Delta_n$. Then $L_1 = T(A) = P_{\{0\}}(T(B) \cap \Delta_n)$ for each $1 \leq i \leq n$.

Similarly, given an n-TPDA (n-TDPDA) [n-TTWDPA] [n-ITWPDA] $B = \langle S, \Sigma, \xi, \$, $\$\$, \Gamma, M, \delta, s_0, \gamma_0, F \rangle$ defining $L_2$, we construct an n-HPDA (n-HDPDA) [n-HTWPDA] [n-HTWDPDA] $A = \langle S, \Sigma, \xi, \$, $\$\$, \Gamma, M, v, s_0, \gamma_0, F \rangle$ where we define $v$ by: For each $s \in S, v(s) = \delta(s)$. Then $T(A) = P_{\{i\}}(T(B) \cap \Delta_n)$ for each $1 \leq i \leq n$.

The proofs of the following theorems are generalizations of the one given for two-way pushdown automata [9] and are omitted. The full proofs can be found in [13].

**Theorem 1.2.** Let $A = \langle S, \Sigma, \xi, \$, $\$\$, \Gamma, M, \delta, s_0, \gamma_0, F \rangle$ be an n-TPDA, and $(z_1, \ldots, z_n) \in [\Sigma^*]^n$. It is decidable whether $(z_1, \ldots, z_n) \in T(A)$.

**Theorem 1.3.** Let $A = \langle S, \Sigma, \xi, \$, $\$\$, \Gamma, M, v, s_0, \gamma_0, F \rangle$ be an n-HPDA, and $z \in \Sigma^*$. It is decidable whether $z \in T(A)$.

### Section 2. n-Context Free Languages and n-TPDA

It is well known that the class of PDA definable sets is precisely the class of context free languages (see, for example, [9], [9]). In this section we introduce the notion of an n-context free language and prove an n-tape analogue of this result. Using this result, a necessary condition for a set to be n-TPDA definable is then proved. For completeness, we include the following definitions.

**Definition.** A context free grammar, CFG, is a 4-tuple $G = \langle V, P, v, \Sigma \rangle$ where

1. $V$ is a finite nonempty set;
2. $P$ is a finite set of ordered pairs, $(\xi, x) \in (V - \Sigma) \times V^*$ (productions);
3. $v \in (V - \Sigma)$ is the initial symbol;
4. $\Sigma \subseteq V$ is the alphabet of terminal symbols.

Elements of $(V - \Sigma)$ are called variables (or nonterminals). We shall write $\xi \rightarrow x$ instead of $(\xi, x) \in P$. 
DEFINITION. Let \( G = (V, P, v, \Sigma) \) be a CFG. For \( x, y \in V^* \), write \( x \Rightarrow y \) if there exist \( x_1, x_2, w \in V^* \), \( \xi \in (V - \Sigma) \) such that \( x = x_1\xi x_2 \), \( y = x_2wx_1 \), and \( \xi \rightarrow w \) is in \( P \). For \( x, y \in V^* \), write \( x \Rightarrow^* y \) if there exist \( r > 0 \), \( w_0, \ldots, w_r \) such that \( w_0 = x \), \( w_r = y \), and \( w_i \Rightarrow w_{i+1} \) for \( 0 \leq i < r \).

DEFINITION. A CFG \( G = (V, P, v, \Sigma) \) is called a right linear CFG (abbreviated RLCFG) if all the rules of \( P \) are of the form: \( \xi \rightarrow w \), or \( \xi \rightarrow w' \), where \( \xi, \xi' \in (V - \Sigma) \) and \( w \in \Sigma^* \).

DEFINITION. If \( G = (V, P, v, \Sigma) \) is a CFG [RLCFG], then the subset \( \{ \Sigma^* \} \) is called a context free language [CFL] if and only if there exists a CFG [RLCFG] \( G \) such that \( L = L(G) \).

Notation. Let \( \Sigma \) be an alphabet and \( n \) be a positive integer. For each \( \sigma \in \Sigma \) and \( 1 \leq i \leq n \), let \( [\Lambda, \ldots, \sigma, \ldots, \Lambda] \) (with \( n - 1 \) occurrences of \( \Lambda \) and \( \sigma \) occurring in the \( i \)th position) be an abstract symbol. Let \( \Sigma_n \) be the set of all such abstract symbols.

DEFINITION. The mapping \( \tau_n \) is a homomorphism from \( \Sigma_n^* \) into \( [\Sigma^*]^n \) defined as follows:

1. \( \tau_n(\Lambda) = (\Lambda, \ldots, \Lambda) \) (with \( n \) occurrences of \( \Lambda \)).
2. For each \( [\Lambda, \ldots, \sigma, \ldots, \Lambda] \) in \( \Sigma_n \), let \( \tau_n([\Lambda, \ldots, \sigma, \ldots, \Lambda]) = (\Lambda, \ldots, \sigma, \ldots, \Lambda) \) (with \( \sigma \) occurring in the same position in both \( [\Lambda, \ldots, \sigma, \ldots, \Lambda] \) and \( (\Lambda, \ldots, \sigma, \ldots, \Lambda) \)).
3. For each \( a_1, \ldots, a_m \) in \( \Sigma_n(m \geq 1) \), let \( \tau_n(a_1 \cdots a_m) = \tau_n(a_1) \cdots \tau_n(a_m) \). Note that the product \( \tau_n(a_1) \cdots \tau_n(a_m) \) yields an element of \( [\Sigma^*]^n \). Hence, \( \tau_n(a_1) \cdots \tau_n(a_m) = (z_1, \ldots, z_n) \) for some \( (z_1, \ldots, z_n) \in [\Sigma^*]^n \). Clearly, \( m = \sum_{i=1}^n lg(z_i) \). Conversely, if \( a_1 \cdots a_m \in \tau_n^{-1}((z_1, \ldots, z_n)) \) then \( m = \sum_{i=1}^n lg(z_i) \).

Example. If \( \Sigma = \{0, 1\} \) and \( n = 2 \), then

\[
\tau_n([0, 1][\Lambda, 1][1, \Lambda]) = (01, 1)
\]

\[
\tau_n^{-1}((01, 1)) = \{[0, 1][\Lambda, 1][1, \Lambda], [0, \Lambda][1, \Lambda][\Lambda, 1], [\Lambda, 1][0, \Lambda][1, \Lambda]\}.
\]

6 A mapping \( \varphi \) from \( \Sigma^* \) into \( \Delta^* \) is a homomorphism if \( \varphi(xy) = \varphi(x)\varphi(y) \) for each \( x, y \in \Sigma^* \).

7 The length of \( x \in \Sigma^* \), denoted by \( lg(x) \), is the number of occurrences of letters of \( \Sigma \) in \( x \). For \( (x_1, \ldots, x_n) \in [\Sigma^*]^n \), \( lg(x) = \sum_{i=1}^n lg(x_i) \).
**Definition.** A subset \( L \subseteq [\Sigma^*]^n \) is called an \( n \)-context free language (\( n \)-CFL) [right linear \( n \)-context free language (\( n \)-RLCFL)] if and only if there exists a CFG [RLCF G] \( G = \langle V, P, v, \Sigma_n \rangle \) such that \( L = \tau_n(L(G)) \).

The proof of the following theorem is essentially a generalization of the one given for 1-TPDA ([3], [6]) and is omitted. The generalized proof is given in [13].

**Theorem 2.1.** A set \( L \subseteq [\Sigma^*]^n \) is an \( n \)-CFL if and only if it is \( n \)-TPDA definable.

The following definitions are borrowed from [20].

**Definition.** Let \( L \subseteq [\Sigma^*]^n \), and \( \emptyset \neq C \subseteq N_n \). Define the existential quantification of \( L \) on the set of coordinates \( C \) (denoted by \( E_c(L) \)) recursively as follows:

\[
E_0(L) = \{ (x_1, \ldots, x_n) \mid \exists x_i \in C \}.
\]

For each \( i \in C \), \( |C| \geq 2 \),

\[
E_i(L) = E_{C(i)}(E_{C-i(i)}(L)).
\]

**Definition.** Let \( L \subseteq [\Sigma^*]^n \), and \( \emptyset \neq C \subseteq N_n \). Define the projection of \( L \) on the set of coordinates \( C \) (denoted by \( P_c(L) \)) as follows:

\[
P_c(L) = E_{\tilde{C}}(L), \text{ where } \tilde{C} = N_n - C.
\]

**Example.** If \( L \subseteq [\Sigma^*]^3 \), then \( E_{(1,1)}(L) = P_{(2,3,0)}(L) = \{ (x_2, x_3, x_3) \mid \exists x_1, x_2 \text{ such that } (x_1, x_2, x_3) \in L \} \).

**Proposition.** Let \( n \geq 2 \), and \( L \subseteq [\Sigma^*]^n \). Let \( L' \subseteq [\Sigma^*]^n \) be obtained from \( L \) by permutation of coordinates. If \( L \) is \( n \)-TPDA (\( n \)-TDPDA) [\( n \)-TTWPDPA] \( n \)-TTWPDPA) definable, then so is \( L' \).

**Proof.** One need only permute the values of the tape selector function, \( \delta \).

We now prove a necessary condition for a set to be \( n \)-TPDA definable.

**Theorem 2.2.** Let \( L \subseteq [\Sigma^*]^n \) \((n \geq 2) \) be definable by an \( n \)-TPDA. Then for each \( 1 \leq i \leq n \), the set \( E_{(i)}(L) \subseteq [\Sigma^*]^{n-1} \) is \((n-1) \)-TPDA definable.

**Proof.** By the preceding proposition it suffices to prove the case when \( i = 1 \). Let \( L \) be an \( n \)-TPDA definable set \((n \geq 2) \). By Theorem 2.1, \( L \) is an \( n \)-CFL. Let \( G = \langle V, P, v, \Sigma_n \rangle \) be a CFG such that \( L = \tau_n(L(G)) \). Define a homomorphism \( \varphi \) from \( \Sigma_n^* \) into \( \Sigma_{n-1}^* \) as follows:

1. \( \varphi(\Lambda) = \Lambda \)
2. For each \( \sigma \in \Sigma \), let \( \varphi([\sigma, \Lambda, \ldots, \Lambda]) = \Lambda \).
3. For each \( \alpha = [\Lambda, \ldots, \sigma, \ldots, \Lambda] \in \Sigma_n \), if \( \sigma \) occurs in the \( k \)th coordinate, \( k \neq 1 \), then let \( \varphi(\alpha) = \beta \), where in \( \beta = [\Lambda, \ldots, \sigma, \ldots, \Lambda] \in \Sigma_{n-1} \), \( \sigma \) occurs in \((k-1)\)th coordinate.
(4) For each $\alpha_1, \alpha_2, \ldots, \alpha_r \in \Sigma_n^*$, let $\varphi(\alpha_1 \alpha_2 \ldots \alpha_r) = \varphi(\alpha_1) \varphi(\alpha_2) \ldots \varphi(\alpha_r)$.

It follows by the closure of context free languages under homomorphisms [1] that $\varphi(L(G))$ is a CFL. So let $G' = (V', P', \nu', \Sigma_{n-1})$ be a CFG such that $\varphi(L(G)) = L(G')$. Clearly, $\tau_{n-1}(L(G')) = E_{11}(L)$, completing the proof.

We immediately get the following result by induction. The simple argument is omitted.

**Theorem 2.3.** Let $L \subseteq [\Sigma^*)^n (n \geq 2)$ be definable by an $n$-TPDA ($n$-TPDA), $C \subseteq N, |C| = m, 1 \leq m < n$. Then $P_c(L)$ and $E_c(L)$ are $m$-TPDA and $(n - m)$-TPDA definable sets, respectively. Moreover if $C = \{i\}$, then $P_{i1}(L) = E_{n-1}(L)$ is a context free language.

We now show that Theorem 2.3 cannot be made stronger in the following sense.

**Theorem 2.4.** For all $n \geq 1$, there is an $(n - 1)$-TPDA definable set $L \subseteq \Sigma^*$ such that for some $1 \leq i \leq n$, $E_{i1}(L)$ is not an $n$-TPDA definable set.

**Proof.** It suffices to prove the case when $n = 1$. Let $L = \{(xx^r, xx^r) \mid x \in \Sigma^*, c \in \Sigma\}$. Clearly, $L$ is definable by a 2-TPDA. However, $E_{11}(L) = P_{i1}(L) = \{xx^r \mid x \in \Sigma^*\}$ is not definable by a deterministic pushdown automaton [7].

It is natural to wonder whether Theorem 2.2 can be strengthened to an “if and only if” condition. We shall show that this is not possible.

**Theorem 2.5.** For all $n \geq 2$, there is a set $L \subseteq [\Sigma^*)^n$ such that for each $i(1 \leq i \leq n)$, $P_{i1}(L)$ is a regular set, and such that $L$ is not $n$-TPDA definable.

**Proof.** It suffices to prove the case when $n = 2$. Let $\phi \neq L \subseteq \Sigma^*$ be any recursively enumerable set which is not recursive (see [4]). Define the set $L' = (L \times \{1\}) \cup ((\Sigma^* - L) \times \{0\})$. Then $P_{i1}(L') = \Sigma^*$ and $P_{i2}(L') = \{0, 1\}$ are regular sets. If $L'$ were 2-TPDA definable, then it would be recursive (by Theorem 1.2). This in turn would imply that $L$ is recursive. Since $L$ was chosen to be nonrecursive, we conclude that $L'$ is not 2-TPDA definable.

---

We are indebted to Professor S. Cook for this proof which shortens a previous proof.
The previous result implies the following theorem. The complete argument involves some extraneous concepts which are not included here. The full proof can be found in [13].

**Theorem 2.6.** For each \( n \geq 1 \), there is an \((n + 1) - \text{TTWPDA}\) \((n + 1) - \text{TTWDPA}\) definable set \( L \subseteq \Sigma^{n+1} \) such that for some \( 1 \leq i \leq n + 1 \), \( E_{(i)}(L) \) is not \( n\)-TTWPDA \((n\)-TTWDPA\) definable.

**SECTION 3. n-TPDA DEFINABLE SETS WITH \( |\Sigma| = 1 \)**

In this section, we shall show that the class of \( n\)-TPDA definable sets coincides with the class of \( n\)-TFA definable sets when the alphabet \( \Sigma \) consists of a single letter. This generalizes a well known result that the CFL's with \( |\Sigma| = 1 \) are regular [6].

**Notation.** Let \( M \) denote the set of nonnegative integers. For each integer \( n \geq 1 \), let \( M^n = M \times \cdots \times M \) \((n\)-times\). We regard \( M^n \) as a subset of the vector space \( \mathbb{Q}^n \) of all \( n\)-tuples of rational numbers over the rational numbers. Thus for elements \((l_1, \ldots, l_n), (k_1, \ldots, k_n) \) in \( M^n \), \( a(l_1, \ldots, l_n) = (al_1, \ldots, al_n) \) and \((l_1, \ldots, l_n) \pm (k_1, \ldots, k_n) = (l_1 \pm k_1, \ldots, l_n \pm k_n) \).

**Definition.** A subset \( L \subseteq M^n \) is called a linear set if there exist elements \( c, p_1, \ldots, p_k \) in \( M^n \) such that \( L = \{c + \sum_{i=1}^{k} k_ip_i \mid k_i \in M\} \). \( c \) is called a constant and \( p_i \) is called a period.

**Notation.** If \( L \) is a linear set with constant, \( c \), and periods \( p_1, \ldots, p_k \), we write \( L = L(c; p_1, \ldots, p_k) \).

**Definition.** A subset of \( M^n \) is called a semilinear set if it is a finite union of linear sets.

**Notation.** Let \( \Sigma = \{\sigma_i \mid 1 \leq i \leq n\} \). The mapping \( \psi \) is the function from \( \Sigma^* \) into \( M^n \) defined by \( \psi(z) = (\#_{\sigma_1}(z), \ldots, \#_{\sigma_n}(z)) \), where \( \#_{\sigma_i}(z) \) is the number of occurrences of \( \sigma_i \) in \( z \). Thus \( \psi(\Lambda) = (0, \ldots, 0) \) and \( \psi(z_1 \cdots z_r) = \sum_{i=1}^{r} \psi(z_i) \), each \( z_i \in \Sigma^* \).

The following theorem which relates context free languages with semilinear sets is due to Parikh [17].

**Theorem 3.1.** If \( L \) is a context free language, then \( \psi(L) \) is a semilinear set.

We will need the following result proved in [20].

**Theorem 3.2.** A subset \( L \subseteq [\Sigma^n]^n \) is \( n\)-TFA definable if and only if it is \( n\)-RLCFL.
We shall show that if $|\Sigma| = 1$, then a subset $L \subseteq [\Sigma^*]^n$ is $n$-TPDA definable if and only if $L$ is $n$-TFA definable; or equivalently, $L$ is $n$-CFL if and only if $L$ is $n$-RLCFL. We do this by a sequence of lemmas.

**Lemma 3.1.** Let $\Sigma = \{0\}$ and $G = \langle V, P, \nu, \Sigma_n \rangle$ be a CFG. For each $z \in \Sigma_n^*$, let $\#_i(z) = \#_{\Delta, \ldots, 0, \ldots, \Delta}(z)$ if $0$ is in the $i$th position of $[\Delta, \ldots, 0, \ldots, \Delta]$. Then $\tau_n(L(G)) = \{(0 \#_1(z), \ldots, 0 \#_n(z)) \mid z \in L(G)\}$.

**Proof.** This is easily verified using the fact that $[\Sigma^*]^n = [[\{0\}^*]^n$ is commutative.

**Corollary 3.1.** Let $\Sigma = \{0\}$ and $G = \langle V, P, \nu, \Sigma_n \rangle$ be a CFG. Then there exists a semilinear set $L \subseteq M^n$ such that $\tau_n(L(G)) = \{(0^l_1, \ldots, 0^l_n) \mid (l_1, \ldots, l_n) \in L\}$.

**Proof.** Follows from Lemma 3.1 and Theorem 3.1.

**Lemma 3.2.** For each linear set $L(c; p_1, \ldots, p_k)$, there exists a RLCFG $G = \langle V, P, \nu, \Sigma_n \rangle$ such that $\tau_n(L(G)) = \{(0^l_1, \ldots, 0^l_n) \mid (l_1, \ldots, l_n) \in L\}$.

**Proof.** The lemma is trivial if $k = 0$ so assume $k > 0$. Let $c = (c_1, \ldots, c_n)$, $p_i = (p_{i1}, \ldots, p_{in})$ for each $1 \leq i \leq k$. Let $G = \langle V, P, \nu, \Sigma_n \rangle$, where $V = \Sigma = \{\nu\} \cup \{v_i \mid 1 \leq i \leq k\}$, $v_i, \ldots, v_n$ are distinct new symbols. $P$ is defined as follows:

1. $v \rightarrow a_1 \cdots a_m v_1$ for each $a_1 \cdots a_m \in \tau_1^{-1}(0^{c_1}, \ldots, 0^{c_n})$.
2. For each $1 \leq i \leq k$, let $v_i \rightarrow a_1 \cdots a_m v_i$ for each $a_1 \cdots a_m \in \tau_1^{-1}(0^{p_{i1}}, \ldots, 0^{p_{in}})$.
3. $v_i \rightarrow v_{i+1}$ for $1 \leq i \leq k - 1$.
4. $v_k \rightarrow \Delta$.

Clearly, $G$ is a RLCFG. It is a straightforward matter to verify that $\tau_n(L(G)) = \{(0^l_1, \ldots, 0^l_n) \mid (l_1, \ldots, l_n) \in L\}$.

**Corollary 3.2.** For each semilinear set, $L$, there exists a RLCFG, $G$, such that $\tau_n(L(G)) = \{(0^l_1, \ldots, 0^l_n) \mid (l_1, \ldots, l_n) \in L\}$.

**Proof.** By definition, $L = \bigcup_{j=1}^r L_j$, each $L_j$ is a linear set. If $r = 1$, we are done. So assume that $r \geq 2$. By Lemma 3.1, for each $1 \leq j \leq r$, there exists a RLCFG $G_j = \langle V_j, P_j, \nu, \Sigma_n \rangle$ such that $\tau_n(L(G_j)) = \{(0^l_1, \ldots, 0^l_n) \mid (l_1, \ldots, l_n) \in L_j\}$. Assume without loss of generality that $(V_j - \Sigma_n) \cap (V_k - \Sigma_n) = \emptyset$ for each $j \neq k$. Let $\nu$ be a new symbol. Define a RLCFG $G = \langle V, P, \nu, \Sigma_n \rangle$, where $V = \{\nu\} \cup \bigcup_{j=1}^r V_j$. $P$ is
defined as follows:

\[ P = (\bigcup_{j=1}^{r} P_j) \cup \{v \rightarrow v_j | 1 \leq j \leq r\}. \]

It is clear that

\[ \tau_n(L(G)) = \bigcup_{i=1}^{r} \tau_n(L(G_j)) = \{(0^i_1, \ldots, 0^n_1) | (l_1, \ldots, l_n) \in L\}. \]

**Lemma 3.3.** Let \( \Sigma = \{0\} \) and \( G = \langle \mathcal{V}, P, v, \Sigma_n \rangle \) be a CFG. Then there exists a RLCFG \( \tilde{G} = \langle \mathcal{V}, \tilde{P}, \tilde{v}, \Sigma_n \rangle \) such that \( \tau_n(L(G)) = \tau_n(L(\tilde{G})) \).

**Proof.** By Corollary 3.1, there exists a semilinear set \( L \subseteq M^n \) such that \( \tau_n(L(G)) = \{(0^1_1, \ldots, 0^n_1) | (l_1, \ldots, l_n) \in L\} \). Then by Corollary 3.2, there exists a RLCFG \( \tilde{G} \) such that

\[ \tau_n(L(\tilde{G})) = \{(0^1_1, \ldots, 0^n_1) | (l_1, \ldots, l_n) \in L\} = \tau_n(L(G)). \]

From Theorems 2.1 and 3.2, and Lemma 3.3 we have the next result.

**Theorem 3.3.** Let \( \Sigma = \{0\} \). A subset \( L \subseteq [\Sigma^*]^n \) is \( n \)-TPDA definable if and only if it is an \( n \)-TFA definable set.

We close this section with a lemma which will be needed in the next section.

**Lemma 3.4.** Let \( n \geq 2 \) and \( 1 \leq j < k \leq n \). If \( L \subseteq [\{0\}^*]^n \) is an \( n \)-TFA definable set, then so is the set \( L_{jk} = L \cap \{(x_1, \ldots, x_n) | x_i \in \{0\}^*, x_j = x_k\} \).

**Proof.** Let \( A = \langle S, \{0\}, \delta, \$, s_0, F \rangle \) be an \( n \)-TFA defining \( L \). We shall construct an \( n \)-TPDA \( B \) such that \( T(B) = L_{jk} \). The result will then follow by Theorem 3.3. Let \( \gamma_0, \gamma_1, \gamma_2, q_1, q_2, f, \) and \( g \) be distinct new symbols. Construct \( B = \langle S_B, \{0\}, \delta, \$, \Gamma_B, M_B, \delta_B, s_0, \gamma_0, \{f\} \rangle \), where

\[ S_B = S \cup \{q_1, q_2, f, g\} \]

\[ \Gamma_B = \{\gamma_0, \gamma_1, \gamma_2\} \]

\( \delta_B \) is defined as follows:

1. For each \( s \in S \), let \( \delta_B(s) = \delta(s) \).
2. \( \delta_B(q_1) = j, \delta_B(q_2) = k \).
3. \( \delta_B(f) = \delta_B(g) = k \).

\( M_B \) is defined by cases. \([s, s'] \in S, \sigma \in (\{0\} \cup \{\$, \$\})\].

1. If \( (d, s') \in M(s, \sigma), \delta(s) \notin \{f, g\}, \) and \( d \neq 1 \), or \( \sigma \neq \$, or \( s' \notin F \), then \( (d, s', \gamma_i) \in M_B(s, \sigma, \gamma_i) \) for each \( 0 \leq i \leq 2 \).
2. If \( (1, s') \in M(s, \$), \delta(s) \notin \{f, g\}, \) and \( s' \in F \), then \( (0, q_1, \gamma_i) \in M_B(s, \$, \gamma_i) \) for each \( 0 \leq i \leq 2 \).
3. If \( (d, s') \in M(s, \sigma), \delta(s) \notin \{j, k\} \) and either \( d \neq 1 \) or
\[ \sigma \in (\{0\} \cup \{\$\}), \text{ then } (d, s', \gamma_i) \in M_B(s, \sigma, \gamma_i) \text{ for each } 0 \leq i \leq 2. \]

(4) If \((1, s') \in M(s, 0) \text{ and } \delta(s) = j, \text{ then}\)
   - (a) \((1, s', \gamma_1) \in M_B(s, 0, \gamma_1)\)
   - (b) \((1, s', \Lambda) \in M_B(s, 0, \gamma_2)\)
   - (c) \((1, s', \gamma_0) \in M_B(s, 0, \gamma_0)\).

(5) If \((1, s') \in M(s, 0) \text{ and } \delta(s) = k, \text{ then}\)
   - (a) \((1, s', \gamma_2) \in M_B(s, 0, \gamma_2)\)
   - (b) \((1, s', \Lambda) \in M_B(s, 0, \gamma_1)\)
   - (c) \((1, s', \gamma_0) \in M_B(s, 0, \gamma_0)\).

(6) If \((1, s') \in M(s, \$), \delta(s) \in \{j, k\}, \text{ and } s' \not\in F, \text{ then}\)
   - (a) \((1, s', \gamma_1) \in M_B(s, \$, \gamma_1)\)
   - (b) \((1, s', \Lambda) \in M_B(s, \$, \gamma_2)\)
   - (c) \((1, s', \gamma_0) \in M_B(s, \$, \gamma_0)\).

(7) If \((1, s') \in M(s, \$), \delta(s) \in \{j, k\}, \text{ and } s' \in F, \text{ then}\)
   - (a) \((0, q, \gamma_1) \in M_B(s, \$, \gamma_1)\)
   - (b) \((0, q, \Lambda) \in M_B(s, \$, \gamma_2)\)
   - (c) \((0, q, \gamma_0) \in M_B(s, \$, \gamma_0)\).

(8) For each \(0 \leq i \leq 1, \text{ let}\)
   - (a) \(M_B(q_1, 0, \gamma_i) = \{(1, q_1, \gamma_1)\}\)
   - (b) \(M_B(q_1, \$, \gamma_i) = \{(0, q_2, \gamma_i)\}\)
   - (c) \(M_B(q_1, 0, \gamma_2) = \{(1, q_1, \Lambda)\}\)
   - (d) \(M_B(q_2, 0, \gamma_1) = \{(1, q_2, \Lambda)\}\)
   - (e) \(M_B(q_2, \$, \gamma_0) = \{(1, q, \gamma_0)\}\).

(9) For each \((s'', \sigma'', \gamma'') \in S_B \times (\{0\} \cup \{\$, \$\}) \times \Gamma_B \text{ such that}\)
   - \(M(s'', \sigma'', \gamma'') \text{ is not defined in rules (1) through (8), let}\)
   - \(M_B(s'', \sigma'', \gamma'') = \{(1, g, \gamma'')\}\).

We sketch briefly the operation of \(B\). Given an \(n\)-tuple \((x_1, \ldots, x_n) \in \{(0)^n\}, B \text{ simulates the action of } A \text{ on } (x_1, \ldots, x_n), \text{ and at the same time compares } \log(x_j) \text{ with } \log(x_k) \text{ via the pushdown tape. The set of rules of } M_B \text{ may be divided as follows: Rules (1) and (2) are used to simulate } A \text{ on tapes other than } x_j \text{ and } x_k . \text{ Rules (2) through (8) are used to simulate } A \text{ on tapes } x_j \text{ and } x_k . \text{ This set of rules allows the comparison of } \log(x_j) \text{ and } \log(x_k). A \text{ rightmost symbol, } \gamma_1, \text{ (respectively, } \gamma_2) \text{ on the pushdown store implies that the length of the prefix}^9 \text{ of } x_j, (\text{respectively, } x_k) \text{ so far scanned by the } j^{th} (\text{respectively, } k^{th}) \text{ reading head of } B \text{ is greater than the length of the prefix of } x_k, (\text{respectively, } x_j) \text{ so far scanned by the } k^{th} (\text{respectively, } j^{th}) \text{ reading head. A rightmost symbol, } \gamma_0, \text{ on the pushdown store implies equality of the lengths of the prefixes of } x_j \text{ and } x_k \text{ so far scanned by the } j^{th} \text{ and } k^{th} \text{ reading heads.}

The \(n\)-tuple \((x_1, \ldots, x_n) \) is accepted by \(B\), in state \(f\), if and only if

---

^9 If \(z = xy\), then \(x(y)\) is a prefix (suffix) of \(z\). In particular, \(\Lambda\) is a prefix and suffix of \(z\).
 SECTION 4. n-HPDA AND LINEAR BOUNDED AUTOMATA

In this section, we shall prove that the class of n-HPDA (n-HDPDA) definable sets is properly contained in the class of sets definable by linear bounded automata (deterministic linear bounded automata).

DEFINITION. A linear bounded automaton (abbreviated, LBA) is a 7-tuple $A = \langle S, \Sigma, \delta, \$, $M, s_0, F \rangle$, where

(1) $S$ and $\Sigma$ are finite nonempty sets (of states and inputs, respectively), with $S \cap \Sigma = \varnothing$.

(2) $\delta, \$ \notin (S \cup \Sigma)$.

(3) $M$ is a mapping from $S \times (\Sigma \cup \{\delta, \$\})$ into the subsets of $\{-1, 0, 1\} \times S \times (\Sigma \cup \{\delta, \$\})$ satisfying the following requirements:

(a) For each $(s, \sigma) \in (S \times \{\delta, \$\})$, if $(d, s', \sigma') \in M(s, \sigma)$, then $\sigma' = \sigma$.

(b) For each $(s, \sigma) \in (S \times \Sigma)$, if $(d, s', \sigma) \in M(s, \sigma)$, then $\sigma' \notin \{\delta, \$\}$.

DEFINITION. A configuration of the LBA is any element of $(\Sigma \cup \{\delta, \$\})*S(\Sigma \cup \{\delta, \$\})*$. A configuration $\sigma_1 \cdots \sigma_{i-1} s \sigma_i \cdots \sigma_m$, each $\sigma_i \in (\Sigma \cup \{\delta, \$\})$, $s \in S$ is to be interpreted as the LBA reading the $i$th symbol of $\sigma_1 \cdots \sigma_m$ in state $s$.

We now describe a relation $\vdash^*$ on configurations.

DEFINITION. Let $A = \langle S, \Sigma, \delta, \$, $M, s_0, F \rangle$ be an LBA. Define the relation $\vdash$ on configurations as follows: Let $x, y \in (\Sigma \cup \{\delta, \$\})*$; $s, s', \sigma \in S$. Then

(1) $x \sigma'' \delta y \vdash x s' \delta' y$ if $(-1, s', \sigma') \in M(s, \sigma)$;

(2) $x \sigma y \vdash x s' \sigma' y$ if $(0, s', \sigma') \in M(s, \sigma)$;

(3) $x \sigma y \vdash x s' \sigma' y$ if $(1, s', \sigma') \in M(s, \sigma)$.

For configurations $\alpha$ and $\beta$ of $A$, we write $\alpha \vdash^* \beta$ if there exist $r > 0$ and configurations $\alpha_0, \cdots, \alpha_r$ such that $\alpha_0 = \alpha$, $\alpha_r = \beta$, and $\alpha_i \vdash \alpha_{i+1}$ for $0 \leq i < r$.

We now define acceptance for an LBA.

DEFINITION. A tape $x \in \Sigma^*$ is accepted by an LBA $A = \langle S, \Sigma, \delta, \$, $M, s_0, F \rangle$ if there exist $s \in F$ and $y \in (\Sigma \cup \{\delta, \$\})*$ such that $s_0 \delta^i x \$ \vdash^* y$s. The set of all tapes accepted by $A$ is denoted by $T(A)$. 
DEFINITION. An LBA for which \(|M(s, \sigma)| \leq 1\) for every \((s, \sigma) \in S \times (\Sigma \cup \{\emptyset, \$\})\) is called a deterministic linear bounded automaton (DLBA, for short).

We now show that given any \(n\)-HPDA \((n\text{-HDPDA})\) definable set \(L\), there is an LBA \((\text{DLBA})\), \(B\), such that \(T(B) = L\).

**Theorem 4.1.** Given any \(n\)-HDPDA \((n\text{-HDPDA})\), \(A\), one can effectively construct an LBA \((\text{DLBA})\), \(B\), such that \(T(B) = T(A)\).

**Proof.** We may assume without loss of generality that \(A = (S, \Sigma, \$, \$, M, \gamma, s_0, \$, F)\) is an \(n\)-HPDA \((n\text{-HDPDA})\) without left endmarker. Furthermore, by Corollary A.2 of the Appendix, we may assume that \(A\) has the following property: For each \(z \in \Sigma^*\), \(lg(z) = m \geq 1\), \((s_0, p_1p_2 \cdots p_nz \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$, \$
time during the computation, channel $k$ may contain at most one 1. (If, at some time, channel $k$ contains all 0's, the $k$th reading head of $A$ must be scanning $\$). Channel $n + 2$ will contain a copy of the pushdown tape of $A$ which has been "compressed." We pack $K$ squares of $A$'s pushdown tape into one square of the present tape. The contents of channel $n + 1$ will be left-justified and the rest of channel $n + 1$ will be $\beta$'s.

For each $\sigma \in \Sigma$,

1. $M_B(h_0, \phi) = \{(+1, h_0, \phi)\}$.
2. $M_B(h_0, \sigma) = \{(+1, h_1, (1, \cdots, 1, \sigma, \gamma_0))\}$.
3. $M_B(h_1, \sigma) = \{(+1, h_1, (0, \cdots, 0, \sigma, \beta))\}$.
4. $M_B(h_1, \$) = \{(-1, h_2, \$)\}$.
5. $M_B(h_2, (0, \cdots, 0, \sigma, \beta)) = \{(-1, h_2, (0, \cdots, 0, \sigma, \beta))\}$.
6. $M_B(h_1, (1, \cdots, \sigma, \gamma_0)) = \{(-1, h_2, (1, \cdots, 1, \sigma, \gamma_0))\}$.
7. $M_B(h_2, \phi) = \{(0, [s_0, k], \phi)\}$, where $k = \nu(s_0)$.

Phase 2. After initialization, $B$ searches for the rightmost $\Gamma$-symbol in channel $n + 2$ under the assumption that channel $n + 2$ is left-justified. It will be seen in later construction that this is the case. Starting in state $[s, k]$ with its reading head on $\phi$, $B$ ends in state $[s, k, \gamma, -1]$ ($\gamma$ being the rightmost $\Gamma$-symbol in channel $n + 2$) with its reading head on $\phi$. The $\gamma$ on channel $n + 2$ is subsequently "erased."

For each $s \in S$, $k = \nu(s)$, $\gamma \in \Gamma$, $(i_1, \cdots, i_n, \sigma, w) \in \Sigma_B$,

8. $M_B([s, k], \phi) = \{(+1, [s, k], \phi)\}$.
9. $M_B([s, k], (i_1, \cdots, i_n, \sigma, w))$

\[
= \begin{cases} 
\{(+1, [s, k], (i_1, \cdots, i_n, \sigma, w))\} & \text{if } w \neq \beta \text{ and } \lg(w) = K.
\{\{(0, [s, k, \gamma, 0], (i_1, \cdots, i_n, \sigma, w'))\} & \text{if } w \neq \beta,
\text{where } w' = \beta \text{ if } w = \gamma \text{ or } w' \in \Gamma^* \text{ if } w = w' \gamma,
\{\{(-1, [s, k, 0], (i_1, \cdots, i_n, \sigma, w))\} & \text{if } w = \beta.
\end{cases}
\]

10. $M_B([s, k, 0], (i_1, \cdots, i_n, \sigma, w)) = \{(0, [s, k, \gamma, 0], (i_1, \cdots, i_n, \sigma, w'))\} \text{ if } w \neq \beta, \lg(w) = K, \text{ where } w' = \beta \text{ if } w = \gamma \text{ or } w' \in \Gamma^* \text{ if } w = w' \gamma.$
(11) \(M_B([s, k, \gamma, 0], (i_1, \ldots, i_n, \sigma, w)) = \{(-1, [s, k, \gamma, 0], (i_1, \ldots, i_n, \sigma, w))\}\).

(12) \(M_B([s, k, \gamma, 0], \notin) = \{(0, [s, k, \gamma, -1], \notin)\}\).

Phase 3. Starting in state \([s, k, \gamma, -1]\) with its reading head on \(\notin\), \(B\) searches for a 1 in channel \(k\). (If channel \(k\) does not contain a 1, \(A\) must be scanning \(\notin\)). This phase ends with \(B\) in state \([s, k, \gamma, \sigma]\) or \([s, k, \gamma, \notin]\), depending on whether or not channel \(k\) contains a 1.

For each \(s \in S, k = v(s), \gamma \in \Gamma, (i_1, \ldots, i_n, \sigma, w) \in \Sigma_B\),

(13) \(M_B([s, k, \gamma, -1], \notin) = \{(+1, [s, k, \gamma, -1], \notin)\}\).

(14) \(M_B([s, k, \gamma, -1], (i_1, \ldots, i_n, \sigma, w))
\begin{align*}
&= \begin{cases} 
(0, [s, k, \gamma, \sigma], (i_1, \ldots, i_n, \sigma, w)) & \text{if } \gamma = 1, \\
(-1, [s, k, \gamma, 0], (i_1, \ldots, i_n, \sigma, w)) & \text{if } \gamma = 0.
\end{cases}
\end{align*}

(15) \(M_B([s, k, \gamma, -1], \notin) = \{(0, [s, k, \gamma, \notin], \notin)\}\).

Phase 4. Starting in state \([s, k, \gamma, \sigma]\) (or \([s, k, \gamma, \notin]\)), \(B\) replaces the 1 on channel \(k\) with a 0 (or does not alter \(\notin\)), moves \(d\) units on the input and enters state \([s', k', w', k]\) where \((d, s', \sigma, w') \in M(s, c, v)\) and \(v(s') = k\).

(16) For each \(1 \leq k \leq n, (s, \sigma, \gamma) \in S \times \Sigma \times \Gamma,\) if \((d, s', w') \in M(s, c, v)\) and \(v(s') = k',\) then \((d, s', w') \in M(s, c, v)\) and \(v(s') = k',\) then \((d, s', w') \in M(s, c, v)\) and \(v(s') = k'.\)

(17) For each \(1 \leq k \leq n, (s, \gamma) \in S \times \Gamma,\) if \((d, s', w') \in M(s, c, v)\) and \(v(s') = k',\) then \((d, s', w') \in M(s, c, v)\) and \(v(s') = k'.\)

Note. If in (17), \(d = 1, B\) will leave the tape as \(A\) would. Thus, the final step of a computation is an application of rule (17).

Phase 5. Starting in state \([s', k', w', k]\), \(B\) puts a 1 on channel \(k\) if its head is not on \(\notin\) (otherwise, it does not alter \(\notin\)), enters state \([s', k', w', -2]\) and moves its reading head toward \(\notin\).

For each \(s' \in S, 1 \leq k \leq n, k' = v(s'), w' \in \Gamma^* (0 \leq lg(w') \leq K),\)

\((i_1, \ldots, i_n, \sigma, w) \in \Sigma_B\).
\begin{align*}
(18) \quad & M_B([s', k', w', k], (i_1, \ldots, i_{k-1}, 0, i_{k+1}, \ldots, i_n, \sigma, w)) = \\
& \{(-1, [s', k', w', -2], (i_1, \ldots, i_{k-1}, 1, i_{k+1}, \ldots, i_n, \sigma, w))\}.
\end{align*}

\begin{align*}
(19) \quad & M_B([s', k', w', k], \Sigma) = \{(-1, [s', k', w', -2], \Sigma)\}.
\end{align*}

\begin{align*}
(20) \quad & M_B([s', k', w', -2], (i_1, \ldots, i_n, \sigma, w)) = \{(-1, [s', k', w', -2], (i_1, \ldots, i_n, \sigma, w))\}.
\end{align*}

\textbf{Phase 6.} Starting in state $[s', k', w', -2]$ with its reading head on $\phi$, $B$ searches for the rightmost symbol of channel $n + 2$ and writes $w'$ (left-justified). $B$ then enters state $[s', k', -1]$ and moves left towards $\phi$ ending in state $[s', k']$ with its reading head on $\phi$. Phase 2 is then repeated.

For each $s' \in S$, $k' = v(s')$, $w' \in \Gamma^* (0 \leq lg(w') \leq K)$, $(i_1, \ldots, i_n, \sigma, w) \in \Sigma_B$,

\begin{align*}
(21) \quad & M_B([s', k', w', -2], \phi) = \{(+1, [s', k', w', -3], \phi)\}.
\end{align*}

\begin{align*}
(22) \quad & \text{If } w' = A, \text{ then } M_B([s', k', w', -3], (i_1, \ldots, i_n, \sigma, w)) = \\
& \{(+1, [s', k', w', -3], (i_1, \ldots, i_n, \sigma, w))\}.
\end{align*}

\begin{align*}
(23) \quad & \text{If } w' \neq A, \text{ then } M_B([s', k', w', -3], (i_1, \ldots, i_n, \sigma, w)) = \\
& \begin{cases}
(+1, [s', k', w', -3], (i_1, \ldots, i_n, \sigma, w)) \text{ if } w \neq \beta \\
(0, [s', k', -1], (i_1, \ldots, i_n, \sigma, ww')) \text{ if } w \neq \beta, 1 \leq lg(w') \leq K \text{ and } w' = w'' \\
(0, [s', k', -1], (i_1, \ldots, i_n, \sigma, w)) \text{ if } w = \beta.
\end{cases}
\end{align*}

\begin{align*}
(24) \quad & M_B([s', k', -1], (i_1, \ldots, i_n, \sigma, w)) = \{(-1, [s', k', -1], (i_1, \ldots, i_n, \sigma, w))\}.
\end{align*}

\begin{align*}
(25) \quad & M_B([s', k', -1], \phi) = \{(0, [s', k'], \phi)\}.
\end{align*}

In all other cases, $M_B$ is empty.

Clearly, $B$ is deterministic if $A$ is. A straightforward induction argu-
MULTI-TAPE AND MULTI-HEAD PUSHDOWN AUTOMATA

ment will show that:

\[ T(B) = \begin{cases} T(A) \setminus \{A\} & \text{if } A \in T(A) \\ T(A) & \text{if } A \notin T(A) \end{cases} \]

**Corollary 4.1.** If \( L \) is an \( n \)-HDPDA definable set, then \( L \) and \( \overline{L} \) are both DLBA definable.

**Proof.** This follows from Theorem 4.1 and the fact that DLBA definable sets are closed under complementation [14].

We now show that an \( n \)-HPDA definable set over a 1-letter alphabet is regular. More precisely we have

**Theorem 4.2.** A subset \( L_1 \subseteq \{0\}^* \) is \( n \)-HPDA definable if and only if it is regular.

**Proof.** It suffices to show necessity. The result is known for the case when \( n = 1 \) [7]. So, suppose \( n \geq 2 \). Then by Theorem 1.1, there exists an \( n \)-TPDA definable set \( L_2 \) such that for each \( 1 \leq i \leq n \), \( L_i = P_{\{i\}}(L_2 \cap \Delta_n) \). By Theorem 3.3, \( L_2 \) is also \( n \)-TFA definable, and by induction using Lemma 3.4, \( L_2 \cap \Delta_n \) is \( n \)-TFA definable. The conclusion now follows from Theorems 2.3 and 3.3 and the result for the case when \( n = 1 \).

**Theorem 4.3.** For each \( n \geq 1 \), the class of \( n \)-HPDA (\( n \)-HDPDA) definable sets is properly contained in the class of LBA (DLBA) definable sets.

**Proof.** Follows from Theorem 4.1 and 4.2 and the well known fact that a DLBA can accept nonregular sets over a 1-letter alphabet.

**Section 5. Closure Properties**

The closure properties of a family of automata have proven to be quite important. Unfortunately, recent techniques [12] for simplifying the study of closure operations do not extend to \( n \)-tape and \( n \)-head automata. Rather than give full proofs of constructions which are well known, we shall state a number of the results without proof. In all cases, details can be found in [13].

First we investigate the closure properties of these sets with respect to boolean operations.

**Theorem 5.1.** For each \( n \geq 1 \), the family of \( n \)-TPDA (\( n \)-TTWPDA) definable sets is closed under union.

This result follows by standard techniques and the argument is omitted.
Next we show that the $n$-TPDA definable sets are not closed under either intersection or complementation.

**Theorem 5.2.** For each $n \geq 1$, the family of $n$-TPDA definable sets is not closed under intersection and complementation.

**Proof.** For $n = 1$, the result is known [1]. We now show the result for $n = 2$ which generalizes in an obvious way. Let $L_1 = \{(x, x) | x \in \{0, 1\}^*\}$ and $L_2 = \{(0^i1^m0^n, 0^i1^m0^n) | i, j, m \geq 0\}$. It is clear that $L_1$ and $L_2$ are both 2-TDPDA definable. Then $L_1 \cap L_2 = \{(0^m1^m0^n, 0^m1^m0^n) | m \geq 1\}$. Assuming $L_1 \cap L_2$ is a 2-TPDA definable set, then by Theorem 2.3, $P_{11}(L_1 \cap L_2) = P_{22}(L_1 \cap L_2) = \{(0^m1^m0^n) | m \geq 1\}$ would be a context-free language. But this set is not a CFL [1], proving that $L_1 \cap L_2$ is not 2-TPDA definable. Nonclosure under complementation now follows from Theorem 5.1 and De Morgan's law.

**Theorem 5.3.** For each $n \geq 1$, the family of $n$-TTWPDA (or $n$-TTWDPDA) definable sets is closed under intersection.

Again the methods are standard and so formal details are omitted. Reference [14] contains full proofs.

For the case of $n$-TDPDA, we have the following theorem, the proof of which is given in the appendix.

**Theorem 5.4.** For each $n \geq 1$, the family of $n$-TDPDA definable sets is closed under complementation.

The following result now follows easily.

**Theorem 5.5.** For each $n \geq 1$, the family of $n$-TDPDA definable sets is not closed under union and intersection.

**Proof.** Using sets $L_1$ and $L_2$ in the proof of Theorem 5.2, we get nonclosure under intersection. Nonclosure under union follows from Theorem 5.4 and De Morgan's law.

We now investigate the closure properties of the $n$-tape models under the operations of concatenation, closure, and transposition.

**Theorem 5.6.** For each $n \geq 1$, the family of $n$-TPDA definable sets is closed under the operations of concatenation, closure, and transposition.

**Proof.** The result follows from Theorem 2.1 and the standard constructions of context free language theory.

---

10 Define the transposition operator by $A^T = A$, $(a_1 \cdots a_k)^T = a_k \cdots a_1$ for $k \geq 1$, $a_i \in \Sigma$. For $(x_1, \cdots, x_n) \in [\Sigma^*]^n$, define $(x_1, \cdots, x_n)^T = (x_1^T, \cdots, x_n^T)$. For $X \subseteq [\Sigma^*]^n$, define $X^T = \{x^T | x \in X\}$. 
As one would expect, we have the following result whose straight-forward proof is omitted.

**Theorem 5.7.** For each \( n \geq 1 \), the family of \( n \)-TTWPDA (\( n \)-TTWDPDA) definable sets is closed under transposition.

Next we prove a technical result which will simplify subsequent proofs.

**Theorem 5.8.** Let \( L \subseteq \{0,1\}^* \) be such that \( L = L_1 \times \cdots \times L_r \), \( L_i \subseteq \{0,1\}^{n_i} \) \((1 \leq i \leq r)\), \( n = \sum_{i=1}^{r} n_i \). Then \( L \) is \( n \)-TPDA (\( n \)-TDPDA) [\( n \)-TTWPDA] (\( n \)-TTWDPDA) definable if and only if for each \( 1 \leq i \leq r \), \( L_i \) is \( n_i \)-TPDA (\( n_i \)-TDPDA) (\( n_i \)-TTWPDA) (\( n_i \)-TTWDPDA) definable.

**Proof.** Sufficiency is clear. To prove necessity, we may assume that \( r = 2 \); the generalization for arbitrary \( r \) can be done by induction. It suffices to show that if \( L = L_1 \times L_2 \) is \( n \)-TPDA (\( n \)-TDPDA) [\( n \)-TTWPDA] (\( n \)-TTWDPDA), then \( L_2 \) is \( n_2 \)-TPDA (\( n_2 \)-TDPDA) (\( n_2 \)-TTWPDA) (\( n_2 \)-TTWDPDA) definable.

Let \( A = \langle S, \Sigma, \delta, \gamma, s_0, \delta_0, F \rangle \) be an \( n \)-TPDA (\( n \)-TDPDA) [\( n \)-TTWPDA] (\( n \)-TTWDPDA) defining \( L \). We may assume without loss of generality that \( A \) has only one final state, that is, \( F = \{ f \} \) and that \( A \) leaves tape \( k \) on acceptance and that \( n_1 < k \leq n \). We shall construct an \( n_2 \)-TPDA (\( n_2 \)-TDPDA) [\( n_2 \)-TTWPDA] (\( n_2 \)-TTWDPDA) \( B \) defining \( L_2 \).

Since \( L = L_1 \times L_2 \), then for each \( (z_1, \ldots, z_n) \in L_1 \), \( \{(z_1, \ldots, z_1)\} \times L_2 \subseteq L \). Let \( (z_1, \ldots, z_n) \in L_1 \) be fixed.

\((\varphi z_1\$, \ldots, \varphi z_n\$)\) will be encoded in the states of \( B \). \( B \) will operate as \( A \) except that when \( A \) does a computation on any one of the tape channels 1, \( \cdots \), \( n_1 \), \( B \) would simulate this computation via the states. The formal construction is now given.

Let \( (z_1, \ldots, z_n) \in L_1 \) such that \( \sum_{i=1}^{n_1} \lg(z_i) \) is minimal. Let \( p_1, \ldots, p_{n_1} \) be distinct new symbols. Construct

\[ B = \langle S_B, \Sigma, \delta, \gamma, s_0, \gamma_0, F_B \rangle, \]

where

\[ T = \{(x_1p_1y_1, \ldots, x_{n_1}p_{n_1}y_{n_1}) \mid s \in S, x_iy_i \]

\[ = \varphi z_i\$ \quad \text{for} \quad 1 \leq i \leq n_1, \quad S_B = S \times T, \quad \text{and} \quad F_B = \{ f \} \times T. \]

\[ s_0 = (s_0, p_1\varphi z_1\$, \ldots, p_{n_1}\varphi z_{n_1}\$) \]

\( \delta_B \) is defined as follows:

1. For each \( (s, x_1p_1y_1, \ldots, x_{n_1}p_{n_1}y_{n_1}) \in S_B \), if \( n_1 < \delta(s) \leq n \), then \( \delta_B((s, x_1p_1y_1, \ldots, x_{n_1}p_{n_1}y_{n_1})) = \delta(s) - n_1 \)

\(^{11}\) See [18], Corollary 2.1 for further documentation.
(2) For each \((s, x_1p_1y_1, \ldots, x_n,p_n y_n) \in S_B\), if \(1 \leq \delta(s) \leq n_1\),
then \(\delta_B((s, x_1p_1y_1, \ldots, x_n,p_n y_n)) = n_2\)

\(M_B\) is defined by cases.

(1) For each \((s, x_1p_1y_1, \ldots, x_n,p_n y_n) \in S_B\), \((\sigma, \gamma) \in (\Sigma \cup \{\epsilon, \$\}) \times \Gamma\), if \((d, s', w) \in M(s, \sigma, \gamma)\) and \(n_1 < \delta(s) \leq n\), then
\((d, (s', x_1p_1y_1, \ldots, x_n,p_n y_n), w) \in M_B((s, x_1p_1y_1, \ldots, x_n,p_n y_n), \sigma, \gamma)\)

(2) For each \((s, x_1p_1y_1, \ldots, \sigma_1 \cdots p_{\sigma_j} \cdots \sigma_m, \ldots, x_n,p_n y_n) \in S_B\), \((\sigma_j, \gamma) \in (\Sigma \cup \{\epsilon, \$\}) \times \Gamma\), if \((d, s', w) \in M(s, \sigma_j, \gamma)\)
and \(\delta(s) = i, 1 \leq i \leq n_1\), then \((0, (s', x_1p_1y_1, \ldots, \sigma_1 \cdots p_{\sigma_j+1} \cdots \sigma_m, \ldots, x_n,p_n y_n), w) \in M_B((s, x_1p_1y_1, \ldots, \sigma_1 \cdots p_{\sigma_j} \cdots \sigma_m, \ldots, x_n,p_n y_n), \sigma, \gamma)\)

It is easily verified that \(B\) satisfies the requirement of the theorem.

**Definition.** A set \(L \subseteq [\Sigma^*]^n\) is an \(n\)-regular product set \((n-RP\ set)\) if \(L\) can be written as \(L = L_1 \times \cdots \times L_n\), each \(L_i\) a regular subset of \(\Sigma^*\).

We now show that the important "\(L \cap R\)" theorem of context free language theory generalizes to the present case.

**Theorem 5.9.** Let \(L_1\) be a \(m\)-TPDA \((m\)-TDPDA\) \(\{m\)-TTWPDA\} definable set and let \(L_2\) be an \(m\)-RP set. Then \(L_1 \cup L_2\) and \(L_1 \cap L_2\) are both \(m\)-TPDA \((m\)-TDPDA\) \(\{m\)-TTWPDA\} definable sets.

**Proof.** (a) Let \(A = (S, \Sigma, \epsilon, \$, \Gamma, M, \delta, s_0, \gamma_0, F)\) be an \(m\)-TPDA \((m\)-TDPDA\) such that \(T(A) = L_1\), and \(L_2 = R_1 \times \cdots \times R_m\), each \(R_i\) a regular set. We may assume that for each \(1 \leq i \leq m, R_i\) is accepted by a deterministic finite automaton with endmarkers. Let \(A_i = (S_i, \Sigma, \epsilon, \$, M_i, s_{0_i}, F_i)\) be finite automata such that \(T(A_i) = R_i\) \((1 \leq i \leq m)\). We shall construct an \(m\)-TPDA \((m\)-TDPDA\) \(B\) such that \(T(B) = L_1 \cap L_2\).

Let \(B = (S_B, \Sigma, \epsilon, \$, \Gamma, M_B, \delta_B, s_{0_B}, \gamma_0, \{F_B\})\) be an \((m\)-TPDA \((m\)-TDPDA) definable as follows:

\[S_B = S \times S_1 \times S_2 \times \cdots \times S_m\]
\[s_{0_B} = (s_0, s_{0_1}, s_{0_2}, \ldots, s_{0_m})\]
\[F_B = F \times F_1 \times F_2 \times \cdots \times F_m\]

\(M_B\) is defined by cases.
MULTI-TAPE AND MULTI-HEAD PUSHDOWN AUTOMATA

For each \((s, \sigma, \gamma) \in S \times (\Sigma \cup \{\$\}) \times \Gamma, (s_1, s_2, \ldots, s_m) \in S_1 \times S_2 \times \cdots \times S_m\),

1. If \((1, s', w) \in M(s, \sigma, \gamma), \delta(s) = i\) and \(1 \leq i \leq m\), then
   \((1, (s', s_1, \ldots, M_i(s_i, \sigma), \ldots, s_m), w) \in M_n((s, s_1, \ldots, s_i, \ldots, s_m), \sigma, \gamma)\)

2. If \((d, s', w) \in M(s, \sigma, \gamma)\) and \(d = 0\) then \((d, (s', s_1, \ldots, s_m), w) \in M_n((s, s_1, \ldots, s_m), \sigma, \gamma)\).

\(\delta_B\) is defined by: For each

\[ (s, s_1, \ldots, s_m) \in S_B, \quad \delta_B((s, s_1, \ldots, s_m)) = \delta(s). \]

It is easily verified that \(L_1 \cap L_2\).

If \(F_B = F \times S_1 \times \cdots \times S_m \cup S \times F_1 \times \cdots \times F_m\), then clearly, \(T(B) = L_1 \cup L_2\).

(b) Let \(L_1\) be an \(m\)-TTWPDA (\(m\)-TTWDPDA) definable set. Then \(L_1 \cap L_2\) is \(m\)-TTWPDA (\(m\)-TTWDPDA) definable. This follows from the closure of \(m\)-TTWPDA (\(m\)-TTWDPDA) definable sets under intersection (Theorem 5.3) and the fact that every \(m\)-RP set is \(m\)-TTWDPDA definable. Since \(m\)-TTWPDA definable sets are also closed under union (Theorem 5.1), it follows that \(L_1 \cup L_2\) is also \(m\)-TTWDPDA definable.

(c) Now suppose that \(L_1\) is \(m\)-TTWDPDA definable, and let \(A = (S, \Sigma, \delta, \gamma_0, F)\) be an \(m\)-TTWDPDA defining \(L_1\). Let \(A_i (1 \leq i \leq m)\) be the deterministic finite automata of part (a). We shall construct an \(m\)-TTWDPDA, \(B\), defining \(L_1 \cup L_2\). Assume that \(S, S_1, \ldots, S_m, \Gamma\) are pairwise disjoint sets. Let \(h, \ldots, h_m\) be distinct new symbols. Construct \(B = (S_B, \Sigma, \epsilon, \$, \Gamma, M_B, \delta_B, s_0, \gamma_0, F \cup F_n)\), where \(S_B = S \cup \bigcup_{i=1}^m S_i \cup \{h_1, \ldots, h_m\}\).

\(\delta_B\) is defined as follows:

1. For each \(s \in S\), \(\delta_B(s) = \delta(s)\).
2. For each \(1 \leq i \leq m\) if \(s \in S_i\), then \(\delta_B(s) = i\).
3. For each \(1 \leq i \leq m\), let \(\delta_B(h_i) = i\).

\(M_B\) is defined by cases.

1. For each \(1 \leq i \leq m\), \(s \in S_i\), \(\sigma \in \Sigma \cup \{\$\}\), if \(M_i(s, \sigma) = s'\), then \(M_B(s, \sigma, \gamma_0) = (1, s', \gamma_0)\).
2. For each \(1 \leq i \leq m - 1\), \(s \in S_i\), if \(M_i(s, \$) = s'\) and \(s' \in F_i\), then \(M_B(s, \$, \gamma_0) = (0, s_0(i+1), \gamma_0)\).
3. For each \(s \in S_m\), if \(M_m(s, \$) = s'\) and \(s' \in F_m\) then \(M_B(s, \$, \gamma_0) = (1, s', \gamma_0)\).
4. For each \(1 \leq i \leq m\), \(s \in S_i\), if \(M_i(s, \$) = s'\) and \(s' \in F_i\), then \(M_B(s, \$, \gamma_0) = (0, h_1, \gamma_0)\).
For each \(1 \leq i \leq m\), \(\sigma \in (\Sigma \cup \{\$\})\), let \(M_B(h_i, \sigma, \gamma_0) = (-1, h_i, \gamma_0)\).

For each \(1 \leq i \leq m - 1\), let \(M_B(h_i, \emptyset, \gamma_0) = (0, h_{i+1}, \gamma_0)\).

Let \(M_B(h_m, \emptyset, \gamma_0) = (0, s_0, \gamma_0)\).

For each \((s, \sigma, \gamma) \in S \times (\Sigma \cup \{\$\}) \times \Gamma\), \(M_B(s, \sigma, \gamma) = M(s, \sigma, \gamma)\).

\(B\) simulates on the \(m\) input tapes the action of the \(m\) deterministic finite automata. If all tapes are in the desired regular set, \(B\) accepts the \(m\)-tuple of tapes, otherwise, \(B\) rewinds all its input tapes and simulates \(A\). Thus \(T(B) = L_1 \cup L_2\).

**Definition.** A set \(L \subseteq [\Sigma^*]^n\) is a semi \(n\)-RP set if it is a Boolean combination of \(n\)-RP sets.

We shall use the following result due to Rosenberg [20] to get some important corollaries to Theorem 5.9.

**Lemma 5.1** (Rosenberg [20]). Every semi \(n\)-RP set is a finite union of \(n\)-RP sets.

**Corollary 5.1.** If \(L_1\) is an \(n\)-TPDA (\(n\)-TTWPDA) definable set and \(L_2\) is a semi \(n\)-RP set, then both \(L_1 \cup L_2\) and \(L_1 \cap L_2\) are \(n\)-TPDA (\(n\)-TTWPDA) definable sets.

**Proof.** Follows from Theorem 5.9, Lemma 5.1, distributive laws, and the closure of \(n\)-TPDA (\(n\)-TTWPDA) definable sets under union (Theorem 5.1).

**Corollary 5.2.** If \(L_1\) is an \(n\)-TDPDA definable set and \(L_2\) is a semi \(n\)-RP set, then both \(L_1 \cup L_2\) and \(L_1 \cap L_2\) are \(n\)-TDPDA definable sets.

**Proof.** (1) That \(L_1 \cup L_2\) is \(n\)-TDPDA definable follows by induction from Theorem 5.9 and Lemma 5.1. (2) To show that \(L_1 \cap L_2\) is \(n\)-TDPDA definable, we have:

\[L_1 \cap L_2 = \overline{L_1} \cup \overline{L_2}\]. By Theorem 5.4, \(\overline{L_1}\) is \(n\)-TDPDA definable, and by definition, \(\overline{L_2}\) is a semi \(n\)-RP set. From (1), \(\overline{L_1} \cup \overline{L_2}\) is \(n\)-TDPDA definable, and again by Theorem 5.4, \(\overline{L_1} \cup \overline{L_2} = L_1 \cap L_2\) is \(n\)-TDPDA definable.

**Corollary 5.3.** If \(L_1\) is an \(n\)-TTWDPDA definable set and \(L_2\) is a semi \(n\)-RP set, then \(L_1 \cup L_2\) is \(n\)-TTWDPDA definable.

**Proof.** Induction from Theorem 5.9 and Lemma 5.1.
MULTI-TAPE AND MULTI-HEAD PUSHDOWN AUTOMATA

Next we show that the family of \( n \)-TDPDA definable sets is not closed under concatenation, \( * \), and transposition.

**Theorem 5.10.** For each \( n \geq 1 \), the family of \( n \)-TDPDA definable sets is not closed under the operations of concatenation, \( * \), and transposition.

**Proof.** Let \( L_1 = L_{11} \times L_{12} \times \cdots \times L_{1n} \) and \( L_2 = L_{21} \times L_{22} \times \cdots \times L_{2n} \) where each \( L_{ij} \subseteq \Sigma^* \) is a deterministic CFL. By Theorem 5.8, \( L_1 \) and \( L_2 \) are both \( n \)-TDPDA definable sets. Furthermore, we have

1. \( L_1 L_2 = L_{11} L_{21} \times \cdots \times L_{1n} L_{2n} \) is \( n \)-TDPDA definable if and only if each \( L_{ij}L_{kj} \) (\( 1 \leq j \leq n \)) is a dot CFL.
2. \( L_1^* = L_{11}^* \times \cdots \times L_{1n}^* \) is \( n \)-TDPDA definable if and only if each \( L_{ij}^* \) (\( 1 \leq j \leq n \)) is a dot CFL.
3. \( L_1^T = L_{11}^T \times \cdots \times L_{1n}^T \) is \( n \)-TDPDA definable if and only if each \( L_{ij}^T \) (\( 1 \leq j \leq n \)) is a dot CFL.

Non-closure of \( n \)-TDPDA definable sets under concatenation, closure, and transposition now follows from (1), (2), and (3), and the fact that the family of deterministic CFL's is not closed under any of these operations [7].

We now establish a number of basic closure results for sets definable by the \( n \)-head models. To this end we state a preliminary lemma. The proof is clear and is omitted.

**Lemma 5.2.** Let \( L_1, L_2 \subseteq \Sigma^* \) and \( L_3, L_4 \subseteq [\Sigma^*]^n \). If for each \( 1 \leq i \leq n \), \( L_1 = P_{[i]}(L_3 \cap \Delta_n) \) and \( L_2 = P_{[i]}(L_4 \cap \Delta_n) \), then for each \( 1 \leq i \leq n \), we have:

(a) \( L_1 = P_{[i]}(L_3 \cap \Delta_n) \)
(b) \( L_1^T = P_{[i]}(L_3^T \cap \Delta_n) \)
(c) \( L_1 \cap L_2 = P_{[i]}([L_3 \cap L_4] \cap \Delta_n) \)
(d) \( L_1 \cup L_2 = P_{[i]}([L_3 \cup L_4] \cap \Delta_n) \)

We now derive our closure results.

**Theorem 5.11.** The family of \( n \)-HPDA (\( n \)-HTWPDA) definable sets is closed under union.

**Proof.** Let \( L_1, L_2 \subseteq \Sigma^* \) be \( n \)-HPDA (\( n \)-HTWPDA) definable sets. By Theorem 1.1, there exist \( n \)-TPDA (\( n \)-TTWPDA) definable sets \( L_3 \) and \( L_4 \) such that for each \( 1 \leq i \leq n \), \( L_1 = P_{[i]}(L_3 \cap \Delta_n) \) and \( L_2 = P_{[i]}(L_4 \cap \Delta_n) \). By Lemma 5.2, \( L_1 \cup L_2 = P_{[i]}([L_3 \cup L_4] \cap \Delta_n) \). The conclusion now follows from the closure of \( n \)-TPDA (\( n \)-TTWPDA) definable sets under union, and Theorem 1.1.
Since \( n\text{-TTWPDA} \) (\( n\text{-TTWDPDA} \)) are closed under intersection (Theorem 5.3), we have, using Lemma 5.2 and Theorem 1.1, the following result.

**Theorem 5.12.** The family of \( n\text{-HTWPDA} \) (\( n\text{-HTWDPDA} \)) definable sets is closed under intersection.

**Theorem 5.13.** The family of \( n\text{-HDPDA} \) definable sets is closed under complementation.

*Proof.* Let \( L_1 \subseteq \Sigma^* \) be \( n\text{-HDPDA} \) definable. By Theorem 1.1 there exists an \( n\text{-TDPDA} \) definable set \( L_2 \subseteq [\Sigma^*]^n \) such that for each \( 1 \leq i \leq n \), \( L_i = P_{\{i\}}(L_2 \cap \Delta_n) \). By Lemma 5.2, \( L_1 = P_{\{i\}}(L_2 \cap \Delta_n) \). Since \( n\text{-TDPDA} \) definable sets are closed under complementation, we conclude by Theorem 1.1 that \( L_1 \) is \( n\text{-HDPDA} \) definable. 

**Corollary.** \( \bigcup_{n=1}^{\infty} \{n\text{-HDPDA definable sets}\} \) is a Boolean algebra.

*Proof.* By the preceding theorem, \( n\text{-HDPDA} \) definable sets are closed under complementation. It is clear that if \( L_1 \) and \( L_2 \) are \( n\text{-HDPDA} \) and \( m\text{-HDPDA} \) definable sets respectively, then \( L_1 \cup L_2 \) and \( L_1 \cap L_2 \) are both \( (n + m)\text{-HDPDA} \) definable sets. It follows that \( \bigcup_{n=1}^{\infty} \{n\text{-HDPDA definable sets}\} \) is a Boolean algebra.

We now consider operations of union and intersection with regular sets.

**Theorem 5.14.** Let \( R \) be a regular set. If \( L_1 \) is an \( n\text{-HPDA} \) (\( n\text{-HTWPDA} \)) \( \{n\text{-HTWDPDA} \} \) definable set, then so are \( L_1 \cap R \) and \( L_1 \cup R \).

*Proof.* By Theorem 1.1, there exists an \( n\text{-TPDA} \) (\( n\text{-TDPDA} \)) \( \{n\text{-TTWPDA} \} \) \( \{n\text{-TTWDPDA} \} \) definable set \( L_2 \) such that for each \( 1 \leq i \leq n \), \( L_i = P_{\{i\}}(L_2 \cap \Delta_n) \). Let \( L_3 = R \times [\Sigma^*]^{n-1} \). By Theorem 5.9 \( L_2 \cup L_3 \) and \( L_2 \cap L_3 \) are both \( n\text{-TPDA} \) (\( n\text{-TDPDA} \)) \( \{n\text{-TTWPDA} \} \) \( \{n\text{-TTWDPDA} \} \) definable sets. The result now follows from Lemma 5.2 and Theorem 1.1.

**Corollary.** Let \( R \) be a regular set. If \( L \) is an \( n\text{-HDPDA} \) definable set, then so are \( L - R \), and \( R - L \).

*Proof.* Now \( L - R = L \cap (\Sigma^* - R) \), and \( R - L = R \cap (\Sigma^* - L) \). The result then follows from Theorems 5.13, 5.14 and the closure of regular sets under complementation ([11], [19]).

We now obtain closure under transposition.
**THEOREM 5.15.** The family of $n$-HPDA ($n$-HTWPDA) [$n$-HTWDPDA] definable sets is closed under transposition.

*Proof.* Let $L_1 \subseteq \Sigma^*$ be $n$-HPDA ($n$-HTWPDA) [$n$-HTWDPDA] definable. Then there exists an $n$-TPDA ($n$-TTWPDA) [$n$-TTWDPDA] definable set $L_2 \subseteq [\Sigma^*]^n$ such that for each $1 \leq i \leq n$, $L_1 = P_{i\Omega}(L_2 \cap \Delta_n)$. By Lemma 5.2, $L_1^T = P_{\Omega}(L_2^T \cap \Delta_n)$. Since $n$-TPDA ($n$-TTWPDA) [$n$-TTWDPDA] definable sets are closed under transposition, the conclusion follows from Theorem 1.1.

In [1], it is shown that context-free languages are closed under homomorphism. We now show that this result does not hold for $n$-TPDA ($n$-HDPDA) [$n$-TTWPDA] [$n$-TTWDPDA] definable sets ($n \geq 2$).

**THEOREM 5.16.** For each $n \geq 2$, the family of $n$-HPDA ($n$-HDPDA) [$n$-HTWPDA] [$n$-HTWDPDA] definable sets is not closed under homomorphism.

*Proof.* It suffices to prove the case when $n = 2$. If $L_1$ and $L_2$ are deterministic context-free languages, then $L_1 \cap L_2$ is 2-HDPDA definable by the Corollary to Theorem 5.13. If the 2-HDPDA definable sets are closed under homomorphism, then any recursively enumerable set can be defined by a 2-HDPDA. This contradicts Theorem 1.3.

**Remark.** Theorem 5.16 is also true for 1-HDPDA definable sets [7], 1-HTWPDA, and 1-HTWDPDA definable sets [9].

Another important operation which preserves context-free languages is a generalized sequential machine (gsm) mapping and inverse gsm mapping [6]. It can be shown (see, [13]) that $n$-HPDA ($n$-HDPDA) definable sets are closed under inverse gsm mappings but not under gsm mappings.

It is well known that the class of context-free languages (sets definable by 1-HPDA or 1-TPDA) is not closed under intersection and complementation ([1], [6]). In trying to show the same nonclosure properties for the case of $n$-HPDA ($n \geq 2$), the authors arrived at a set which is $n$-HDPDA definable but which does not appear to be $(n - 1)$-HPDA definable. For the case $n = 1$, a proof that this set satisfies the requirement can be shown using the results in [1]. At present, a formal proof

---

12 We are using a theorem from [9] which states that any recursively enumerable set is the homomorphic image of the intersection of two deterministic context free languages.
for the case $n \geq 2$ is not available. We shall close this section with the following conjecture, based on which, several interesting results follow.

**Conjecture.** Let $|\Sigma| \geq 2$, and $c$ a symbol not in $\Sigma$. For each $n \geq 2$, define the following sets:

\[ L_2 = \{ xxz | x \in \Sigma^* \} \]
\[ L_{n+1} = \{ xyx | y \in L_n, x \in \Sigma^* \} \] (For example, $L_3 = \{ xxycycl | x, y \in \Sigma^* \}$.)

Then for each $n \geq 2$, $L_n$ is $n$-HDPDA definable but not $(n-1)$-HPDA definable.

The proof that $L_2$ is not a CFL follows from the results in [1]. It is also clear that each $L_n$ is $n$-HDPDA definable; in fact, one can construct an $n$-HDPDA defining $L_n$ without using its pushdown store. Because of the "last-in-first-out" character of the pushdown tape, it seems that for this particular set, the pushdown store can in no way be made part of any successful computation, except to possibly speed up the operation.

If our conjecture were true, then we would have the following:

**Consequence 1.** For each $n \geq 2$, the class of $n$-HPDA ($n$-HDPDA) definable sets is not closed under intersection and complementation (intersection and union).

**Proof.** Let $A_n = \{ xxycycl | y \in L_n, x, z \in \Sigma^* \}$
\[ A_2 = \{ xxycycl | y \in (\Sigma \cup \{c\})^*, x \in \Sigma^* \} \]

Clearly, $A_n$ and $A_2$ are $n$-HDPDA and 2-HDPDA definable sets.

However, $A_n \cap A_2 = L_{n+1}$.

One may also prove the following consequence if the conjecture were true.

**Consequence 2.** For each $n \geq 2$, the family of $n$-HDPDA definable sets is not closed under the operations of concatenation, closure, and transposition.

**SECTION 6. DECISION QUESTIONS**

In this section, we shall investigate decision questions associated with the classes of sets definable by our $n$-tape and $n$-head models. A number of undecidability results from the theory of context free languages ([1], [6]) carry over to the classes under consideration.

**Theorem 6.1.** For each $n \geq 1$, it is effectively decidable for an arbitrary $n$-TPDA ($n$-TDPDA) definable set $L$ whether it is empty, finite, or infinite.
Proof. The argument is a straightforward induction on \( n \) using Theorem 1.1.

We have another positive decision result.

**Theorem 6.2.** For each \( n \geq 1 \), it is recursively solvable to determine of an arbitrary \( n \)-TDPDA definable set and semi \( n \)-RP set \( R \) whether \( L = R \).

**Proof.** \( L = R \) if and only if
\[
L' = [L \cap ([\Sigma^*]^n - R)] \cup ([\Sigma^*]^n - L) \cap R = \emptyset.
\]
By Theorem 5.4, \([\Sigma^*]^n - L\) is \( n \)-TDPDA definable. Hence by Corollary 5.2, \( L \cap ([\Sigma^*]^n - R) \) and \(([\Sigma^*]^n - L) \cap R\) are \( n \)-TPDA definable. It follows from Theorem 5.1 that \( L' \) is \( n \)-TPDA definable. Then \( L = R \) if and only if \( L' = \emptyset \), which is decidable by Theorem 6.1.

**Corollary.** It is recursively solvable to determine for an arbitrary \( n \)-TDPDA definable set \( L(n \geq 1) \) whether \( L = [\Sigma^*]^n \).

**Proof.** \([\Sigma^*]^n\) is clearly a semi \( n \)-RP set.

We now turn to unsolvable problems.

**Theorem 6.3.** For arbitrary \( n \)-TPDA definable sets \( L_1 \) and \( L_2 \) and a semi \( n \)-RP set \( R(n \geq 1) \), it is recursively unsolvable to determine whether
- (a) \( L_1 \) is \( n \)-TDPDA definable
- (b) \( L_1 = R \)
- (c) \( L_1 = [\Sigma^*]^n \)
- (d) \( L_1 \cap L_2 \) is empty, finite, infinite, or a semi \( n \)-RP set
- (e) \( L_1 \subseteq L_2 \)
- (f) \( L_1 = L_2 \)

**Proof.** The argument is an induction on \( n \) using well known techniques and is omitted. The full proof is in [13].

By analogous and well known techniques, the following result can be established.

**Theorem 6.4.** For arbitrary \( n \)-TDPDA definable sets \( L_1 \) and \( L_2 \) \( (n \geq 1) \), it is recursively unsolvable to determine whether
- (a) \( L_1 \cup L_2 \) is \( n \)-TDPDA definable
- (b) \( L_1 \subseteq L_2 \)
- (c) \( L_1 L_2 \) is \( n \)-TDPDA definable
- (d) \( L_1^* \) is \( n \)-TDPDA definable.
Turning now to two way automata, and $n$ head automata we find that all the conventional problems are unsolvable. We summarize these results with the following theorems whose proofs are omitted. Again, see [13].

**Theorem 6.5.** For arbitrary $n$-TTWPDA ($n$-TTWDPDA) definable sets $L_1$ and $L_2$ ($n \geq 1$), it is recursively unsolvable to determine whether

(a) Each of the sets $L_1$, $L_1$, $L_1 \cap L_2$ is empty, finite, infinite, semi $n$-RP set, or $n$-TPDA definable.

(b) $L_1 = [\Sigma^n]^n$

(c) $L_1 \subseteq L_2$

(d) $L_1 \neq L_2$

**Theorem 6.6.** For each $n \geq 2$, it is recursively unsolvable to determine for arbitrary $n$-HDPDA ($n$-HPDA) [$n$-HTWDPDA] [$n$-HTWPDA] sets $L_1$ and $L_2$ whether

(a) $\Sigma^* - L_1$ is empty, finite, infinite, regular, or context-free.

(b) $L_1 \cap L_2$ is empty, finite, infinite, regular, or context-free.

(c) $L_1 = \Sigma^*$

(d) $L_1 = L_2$

(e) $L_1 \subseteq L_2$

**Theorem 6.7.** For each $n \geq 2$ and an arbitrary $n$-HDPDA ($n$-HPDA) [$n$-HTWDPDA] [$n$-HTWPDA] $L$, it is recursively unsolvable to determine whether $L$ is empty, finite, infinite, regular, or context-free.

**Appendix**

**Proof of Theorem 5.4**

We shall assume here that an $n$-TPDA ($n$-TDPDA) shall mean an $n$-TPDA ($n$-TDPDA) without left endmarkers. This does not create any real loss of generality.

**Definition.** An $n$-TPDA ($n$-TDPDA) $A = \langle S, \Sigma, \$, $\Gamma, M, \delta, s_0, \gamma_0, F \rangle$ is said to be **completely specified** if for each $(s, \sigma, \gamma) \in S \times (\Sigma \cup \{$$\}$) $\times \Gamma$, the following conditions hold:

1. $\delta(s) \neq \emptyset$
2. $M(s, \sigma, \gamma) \neq \emptyset$
3. $(d, s', w) \in M(s, \sigma, \gamma_0)$ implies that $w = \gamma_0w'$ for some $w' \in \Gamma^*$.

Thus a completely specified $n$-TPDA ($n$-TDPDA) $A$ has no "blocking" configuration, that is, it has always a possible move.

The proof of the following proposition is obvious and is omitted.
PROPOSITION. For each $n$-TPDA ($n$-TDPDA) $A$ there exists a completely specified $n$-TPDA ($n$-TDPDA) $B$ such that $T(B) = T(A)$.

Remark. Although a completely specified $n$-TPDA ($n$-TDPDA) $A$ has no blocking configuration, it may happen that for some $(z_1, \ldots, z_n) \in [\Sigma^*]^n$, no reading head of $A$ leaves $(z_1\$,$ \ldots, z_n\$)$. This is because we allow rules of the form $(0, s', w) \in M(s, \sigma, \gamma)$. We, however, include such rules in our definition of $n$-TPDA ($n$-TDPDA) to allow more flexibility and generality. In the case of $n$-TDPDA, a wider class of sets can be defined with such rules. Thus, there exist $n$-TDPDA definable sets accepted by no $n$-TDPDA without rules of the form $M(s, \sigma, \gamma) = (0, s', w)$.

An example of such a set is the following: Let $\Sigma = \{a, b, c\}$ and $n \geq 2$. Let $L = \{a^i b^j a^i | i, j \geq 1\} \times [\Sigma^*]^{n-1} \cup \{a^i b^j c^a a^i | i, j \geq 1\} \times [\Sigma^*]^{n-1}$. Clearly $L$ can be defined by an $n$-TDPDA. The proof that $L$ is not definable by any $n$-TDPDA without rules of the form $M(s, \sigma, \gamma) = (0, s', w)$ follows from Theorem 5.8 and the fact that $\{a^i b^j a^i | i, j \geq 1\} \cup \{a^i b^j c^a a^i\}$ is not definable by any $1$-TDPDA without such rules [7].

DEFINITION. Let $A = (S, \Sigma, \delta, s_0, \gamma_0, F)$ be a completely specified $n$-TPDA ($n$-TDPDA). We define the relation $\rightarrow_d^*$ between ID's as follows:

$(s, x_1p_1y_1, \ldots, x_np_ny_n, w) \rightarrow_d^* (s', x_1'p_1y_1', \ldots, x_n'p_ny_n', w')$ if $(s, x_1p_1y_1, \ldots, x_np_ny_n, w) \rightarrow_d^* (s', x_1'p_1y_1', \ldots, x_n'p_ny_n', w')$ and either (1) $w' = \Lambda$ or (2) if $w' \neq \Lambda$, $(s', x_1'p_1y_1', \ldots, x_i'p_iy_i', \ldots, x_n'p_ny_n', w')$ $\rightarrow_d^* (s'', x_1''p_1y_1'', \ldots, x_i''p_iy_i'', \ldots, x_n''p_ny_n'', w'')$ for some $1 \leq i \leq n$, $y_i' = \sigma z_i$.

DEFINITION. Let $A = (S, \Sigma, \delta, \gamma_0, F)$ be a completely specified $n$-TPDA ($n$-TDPDA). An $n$-tuple $(z_1, \ldots, z_n) \in [\Sigma^*]^n$ is $d$-accepted by $A$ if $(s_0, p_1z_1\$, \ldots, p_nz_n\$, \gamma_0) \rightarrow_d^* (s, x_1p_1y_1, \ldots, z_i\$, \ldots, x_np_ny_n, w)$ $\rightarrow_d^* (s', x_1p_1y_1, \ldots, z_i\$, \ldots, x_np_ny_n, w')$ for some $s \in S$, $s' \in F$, $w, w' \in \Gamma^*$ and $1 \leq i \leq n$. Let $T_d(A) = \{(z_1, \ldots, z_n) | (z_1, \ldots, z_n) \text{ is } d\text{-accepted by } A\}$.

Notation. For each completely specified $n$-TPDA ($n$-TDPDA) $A = (S, \Sigma, \delta, \gamma_0, F)$, let $A_c = (S, \Sigma, \delta, s_0, \gamma_0, S - F)$. Note that $A_c$ is completely specified.
DEFINITION. A completely specified \( n \)-TPDA (\( n \)-TDPDA) \( A \) is said to be loop-free if \( T_d(A) \cup T_d(A_\gamma) = [\Sigma^*]^n \). Thus a completely specified \( n \)-TPDA (\( n \)-TDPDA) \( A \) is loop-free if and only if for every \((z_1, \ldots, z_n) \in [\Sigma^*]^n\), \((s_0, p_1z_1s_1, \ldots, p_nz_ns_\gamma) \)--- \( d^* \) \((s, x_1p_1y_1, \ldots, x_np_ny_n, w) \) --- \((s', x_1p_1y_1, \ldots, x_np_ny_n, w') \) for some \( s, s' \in S \), \( w, w' \in \Gamma^* \) and \( 1 \leq i \leq n \).

We now prove the following lemma.

**Lemma A.1.** If \( A \) is a completely specified \( n \)-TPDA (\( n \)-TDPDA), then \( T(A) = T_d(B) = T(B) \) for some completely specified loop-free \( n \)-TPDA (\( n \)-TDPDA) \( B \).

**Proof.** Let \( A = (S, \Sigma, \delta, \Gamma, \delta_0, s_0, \gamma_0, F) \) be a completely specified \( n \)-TPDA (\( n \)-TDPDA). Construct \( B = (S_B, \Sigma, \delta_B, s_{0B}, \gamma_0, F_B) \), where

\[
S_B = \{ s_{0B}, q \} \cup \{ s_0 \} \times (\bigcup_{1 \leq i \leq n} [\Sigma \cup \{\}$] \} U S \times [\Sigma \cup \{\}$] \}
\]

where \( s_{0B} \) and \( q \) are new symbols.

\[ \Gamma_B = \Gamma \cup \mathcal{S}, \quad \Gamma \cap S = \emptyset. \]

\( \delta_B \) is defined as follows:

1. \( \delta_B(s_{0B}) = \delta_B(q) = 1 \)
2. For each \( \sigma_1, \ldots, \sigma_j(1 \leq j \leq n - 1) \) in \( \Sigma \cup \{\}$\), let \( \delta_B(s_0, \sigma_1, \ldots, \sigma_j) = j + 1 \).
3. For each \( \sigma_1, \ldots, \sigma_n \in \Sigma \cup \{\}$\), \( s \in S \), let \( \delta_B((s, \sigma_1, \ldots, \sigma_n)) = \delta(s) \)

\( M_B \) is defined by cases.

1. For each \( \sigma \in \Sigma \cup \{\}$\), let \( M_B(s_{0B}, \sigma, \gamma_0) = \{(0, (s_0, \sigma), \gamma_0)\} \)
2. For each \( \sigma_1, \ldots, \sigma_j(1 \leq j \leq n - 1) \) in \( \Sigma \cup \{\}$\), \( \sigma \in \Sigma \cup \{\}$\), let \( M_B((s_0, \sigma_1, \ldots, \sigma_j, \sigma, \gamma_0) = \{(0, (s_0, \sigma_1, \ldots, \sigma_j, \sigma, \gamma_0)\} \)
3. For each \((s, \sigma, \gamma) \in S \times (\Sigma \cup \{\}$\) \times \( \Gamma \) if \((1, s', w) \in M(s, \sigma, \gamma) \) and \( \delta(s) = i \), then \((1, (s, \sigma_1, \ldots, \sigma_n, ws')) \in M_B((s, \sigma_1, \ldots, \sigma_n, \sigma, \gamma)) \) for each \((s, \sigma_1, \ldots, \sigma_n) \in S_B \) with \( \sigma_i = \sigma \).
4. For each \((s, \sigma_1, \ldots, \sigma_i, \ldots, \sigma_n) \in S_B \), \( s, s' \in S \), \( \sigma \in \Sigma \cup \{\}$\) if \( \delta(s) = i \), then \((0, (s', \sigma_1, \ldots, \sigma, \ldots, \sigma_n, \gamma)) \) \( \in M_B((s, \sigma_1, \ldots, \sigma_i, \ldots, \sigma_n, \sigma, s')) \).
5. For each \((s, \sigma, \gamma) \in S \times (\Sigma \cup \{\}$\) \times \( \Gamma \) and \((s, \sigma_1, \ldots, \sigma_n) \in S_B \) with \( \sigma_i = \sigma \) if \((0, s', w) \in M(s, \sigma, \gamma) \), \( \delta(s) = i \),
and \((s, \ p_1\sigma_1, \ \ldots, \ p_n\sigma_n, \ \gamma) \xrightarrow{\text{d}} (s', \ p_1\sigma_1, \ \ldots, \ p_n\sigma_n, \ \w')\)
then \((0, \ (s', \ \sigma_1, \ \ldots, \ \sigma_n), \ \w') \in M_B((s, \ \sigma_1, \ \ldots, \ \sigma_n), \ \sigma, \ \gamma)\)

(6) For each \((s, \ \sigma, \ \gamma) \in S_B \times (\Sigma \cup \{\$\}) \times \Gamma\) such that \(M_B(s, \ \sigma, \ \gamma)\) is not defined in rules (1) through (5), let \(M_B(s, \ \sigma, \ \gamma) = \{(1, \ q, \ \gamma)\}\).

Clearly \(B\) is completely specified and is deterministic if \(A\) is.

We describe briefly how \(B\) operates. Given an \(n\)-tuple, \(B\) uses rules (1) and (2) to encode in the states the initial symbol of each of the tape channels. This initializes the computation of \(B\). At the end of the initialization, each reading head of \(B\) is still scanning the first symbol of its corresponding tape channel, and the state is \((s_0, \ \sigma_1, \ \ldots, \ \sigma_n)\).

The state of the form \((s, \ \sigma_1, \ \ldots, \ \sigma_n)\) keeps track of the present symbol being scanned by each reading head. We now compare the computation of an \(n\)-tuple by \(B\) with that of \(A\). (a) If a reading head of \(A\) moves right on a symbol in the \(i\)th channel, \(B\) simulates this action using rules (3) and (4). \(B\) uses rule (4) to replace the \(i\)th symbol in the state by the present symbol being scanned by the \(i\)th reading head. (b) If the reading heads of \(A\) go through a long but finite sequence of “non-moving” actions, \(B\) uses rule (5) to effect these actions in one move. (c) If the reading heads of \(A\) go through an infinite sequence of “non-moving” actions, \(B\) uses rule (6) to reject the \(n\)-tuple. By induction, one can show that \(T(A) = T_a(B) = T(B)\).

**Corollary A.1.** Let \(A = \langle S, \ \Sigma, \ \$, \ \Gamma, \ M, \ \delta, \ s_0, \ \gamma_0, \ F \rangle\) be a completely specified \(n\)-TPDA (\(n\)-TDPDA). Then there exists a completely specified loop-free \(n\)-TPDA (\(n\)-TDPDA) \(B = \langle S_B, \ \Sigma, \ \$, \ \Gamma_B, \ M_B, \ \delta_B, \ s_0B, \ \gamma_0B, \ F \rangle\) such that the following conditions hold:

1. \(T_a(B) = T(B) = T(A)\)
2. For each \((z_1, \ \ldots, \ z_n) \in [\Sigma^*]^n\), \((s_0B, \ p_1z_1\$, \ \ldots, \ p_nz_n\$, \ \gamma_0B) \xrightarrow{\text{c}} (s, \ x_1p_1y_1, \ \ldots, \ x_np_ny_n, \ w)\) for some \((s, \ w) \in S_B \times \Gamma_B^*\) implies that \(1 \leq lg(w) \leq h(3(\sum_{i=1}^{n} lg(z_i))) + 4n\) where \(h = \max\{lg(w) \mid (d, \ s', \ w) \in M_B(s'', \ \sigma, \ \gamma)\}\).
3. If in (2), \(z_1 = z_2 = \cdots = z_n = z\), and \(lg(z) = m \geq 1\) then \(1 \leq lg(w) < (5hn)m\).

**Proof.** Let \(B\) be the completely specified loop-free \(n\)-TPDA (\(n\)-TDPDA) constructed in Lemma A.1. In computing an \(n\)-tuple \((z_1\$, \ \ldots, \ z_n\$)\), \(B\) has \(n\) moves for the initialization (see rules (1) and (2) of Lemma A.1), at most \(\sum_{i=1}^{n} (lg(z_i) + 1)\) moves using a rule of the...
form \((1, s', w) \in M_B(s, \sigma, \gamma)\) (see rules (3) and (6) of Lemma A.1) and at most \(2 \sum_{i=1}^{\gamma n} (\lg(z_i) + 1)\) moves using a rule of the form \((0, s', w) \in M_B(s, \sigma, \gamma)\) (see rules (4) and (5) of Lemma A.1). It follows that 
\[\lg(w) \leq h[3(\sum_{i=1}^{\gamma n} \lg(z_i)) + 4n].\]
Since \(B\) is completely specified, \(\lg(w) \geq 1\).

If \(z_1 = z_2 = \cdots = z_n = z\), then \(1 \leq \lg(w) \leq h[3 n \lg(z) + 4 n] = hn[3 \lg(z) + 4].\) If \(\lg(z) = m \geq 1\), then \(3m + 4 < 5m\).
Hence \(1 \leq \lg(w) < (5hn)m\).

**Corollary A.2.** Let \(A = \langle S, \Sigma, \$, \Gamma, M, \nu, s_0, \gamma_0, F \rangle\) be any \(n\)-HPDA (n-HDPDA). Then there exists an \(n\)-HPDA (n-HDPDA) \(B = \langle S_B, \Sigma, \$, \Gamma_B, M_B, \nu_B, s_0, \gamma_0, F_B \rangle\) such that the following conditions hold:

1. \(T(B) = T(A)\)
2. For each \(z \in \Sigma^*\), \(\lg(z) = m \geq 1\), \((s_0, p_1 p_2 \cdots p_n z\$, \gamma_0) \vdash \star (s, x_0 p_1 x_1 \cdots p_n x_n, w)\) for some \(x_0 x_1 \cdots x_n = z\$, \((s, w) \in \Sigma_B \times \Gamma_B^*\) implies that \(1 \leq \lg(w) < Km\) where \(K = 5hn\) and \(h = \max \{\lg(w) \mid (d, s', w) \in M_B(s', \sigma, \gamma)\}\)

**Proof.** Follows from Corollary A.1, and Theorem 1.1

Before we can prove the main result, we need another lemma.

**Lemma A.2.** For each completely specified \(n\)-TDPDA, \(A\), there exists a completely specified \(n\)-TDPDA, \(B\), such that \(T_d(A) = T_d(B) = T(B)\).

**Proof.** Let \(A = \langle S, \Sigma, \$, \Gamma, M, \delta, s_0, \gamma_0, F \rangle\) be a completely specified \(n\)-TDPDA. Without loss of generality we may assume that \(s_0 \not\in F\). Construct \(B = \langle S_B, \Sigma, \$, \Gamma, M_B, \delta_B, s_0, \gamma_0, F_B \rangle\), where \(S_B = S \cup \{\bar{s} \mid s \in S\}\), \(F_B = \{\bar{s} \mid s \in F\}\).

\(M_B\) is defined by cases.

1. For each \((s, \sigma, \gamma) \in S \times (\Sigma \cup \{\$\}) \times \Gamma\), if \(M(s, \sigma, \gamma) = (1, q, w)\) for some \((q, w) \in S \times \Gamma^*\), then let (a) \(M_B(\bar{s}, \sigma, \gamma) = (0, s, \gamma)\) (b) \(M_B(s, \sigma, \gamma) = (1, q, w)\)
2. For each \((s, \sigma, \gamma) \in S \times (\Sigma \cup \{\$\}) \times \Gamma\) if \(M(s, \sigma, \gamma) = (0, q, w)\) for some \((q, w) \in S \times \Gamma^*\), then \(M_B(s, \sigma, \gamma) = M_B(\bar{s}, \sigma, \gamma) = (0, q, w)\).

\(\delta_B\) is defined as follows:

For each \(s \in S\),
\[\delta_B(s) = \delta_B(\bar{s}) = \delta(s)\].
Clearly $B$ is deterministic and completely specified, and $T_d(A) = T_d(B)$. Furthermore, for each $(z_1, \ldots, z_n) \in [\Sigma^*]^n$,

$$(s_0, p_1s_1, \ldots, p_nz_n, \gamma_0) \vdash^* (s, x_1p_1y_1, \ldots, z_1s_1, \ldots, x_ny_n, w)$$

for some $s \in F_B$, $w \in \Gamma^*$, $1 \leq i \leq n$ if and only if

$$(s_0, p_1s_1, \ldots, p_nz_n, \gamma_0) \vdash^* (q, x_1p_1y_1, \ldots, z_1s_1, \ldots, x_ny_n, w')$$

$$(s_0, p_1s_1, \ldots, p_nz_n, \gamma_0) \vdash^* (q, x_1p_1y_1, \ldots, z_1s_1, \ldots, x_ny_n, w')$$

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

We are now ready to prove Theorem 5.4.

**Theorem 5.4.** For each $n \geq 1$, the family of $n$-TDPDA definable sets is closed under complementation.

**Proof.** We may assume that $L = T(A)$ for some completely specified $n$-TDPDA $A$. By Lemma A.1, $T(A) = T_d(B)$ for some completely specified loop-free $n$-TDPDA $B$. $B$ being completely specified and loop-free, $T_d(B) \cup T_d(B_c) = [\Sigma^*]^n$. Moreover since $B$ is deterministic, $T_d(B) \cap T_d(B_c) = \varnothing$. Hence

$$[\Sigma^*]^n - L = [\Sigma^*]^n - T(A) = [\Sigma^*]^n - T_d(B) = T_d(B_c)$$

with $B_c$ deterministic, loop-free, and completely specified. By Lemma A.2, there is a completely specified $n$-TDPDA $D$ such that $T_d(B_c) = T_d(D) = T(D)$. Hence $[\Sigma^*]^n - L$ is $n$-TDPDA definable.

**Received:** April 11, 1968

**References**


