Reversal-Bounded Multicounter Machines and Their Decision Problems

OSCAR H. IBARRA

University of Minnesota, Minneapolis, Minnesota

ABSTRACT Decidable and undecidable properties of various classes of two-way multicounter machines (deterministic, nondeterministic, multitape, pushdown store augmented) with reversal-bounded input and/or counters are investigated In particular it is shown that the emptiness, infiniteness, disjointness, containment, universe, and equivalence problems are decidable for the class of deterministic two-way multicounter machines whose input and counters are reversal-bounded

KEY WORDS AND PHRASES: multicounter machines, reversal-bounded, Turing machines, pushdown machines, multitape machines, decision problems, decidable, unsolvable

CR CATEGORIES: 5 22, 5 23, 5 26, 5 27

1. *Introduction*

A desirable property of the class of finite automata that is not shared by other well-known classes of devices is the existence of decision procedures for all interesting questions concerning finite automata [23]. For example, the following problems which we shall refer to as F-problems are decidable: emptiness, infiniteness, disjointness, containment, universe, and equivalence problems.[1]

The proof of the decidability of these questions makes use of the fact that a finite automaton has only a finite number of internal states or configurations. When a read-write unbounded memory is attached to the automaton, almost all F-problems become unsolvable. If the unbounded memory is in the form of a semi-infinite tape which is operated sequentially we get a Turing machine (TM). For TMs the F-problems are undecidable [4, 16] If the tape is restricted to operate on a last-in-first-out basis, the device reduces to a pushdown machine, and the emptiness and infiniteness problems become solvable [2]. All other decision questions are unsolvable [2]. For example, the containment and disjointness problems for deterministic pushdown machines are undecidable [9], and the results carry over to the restricted cases when the pushdown store is reversal-bounded (i.e. finite-turn) or when the pushdown store has a single-letter alphabet (i.e. it is operated as a counter). The status of the equivalence problem is still open although it has been shown decidable for the restricted cases just mentioned [26, 27]. For nondeterministic counter machines, the universe problem is unsolvable even if the counter is restricted to make at most one reversal [1]. Perhaps the simplest known subclass of deterministic pushdown machines is the simple deterministic pushdown

This research was supported by the National Science Foundation under Grant DCR75-17090

Author's address. Department of Computer Science, University of Minnesota, Minneapolis, MN 55455

[1] Let C be a class of machines The emptiness, infiniteness, disjointness, containment, universe, and equivalence problems are the problems of deciding for arbitrary machines M_1 and M_2 in C whether $T(M_1) = \varnothing$, $T(M_1)$ is infinite, $T(M_1) \cap T(M_2) = \varnothing$, $T(M_1) \subseteq T(M_2)$, $T(M_1)$ is the set of all finite-length strings, and $T(M_1) = T(M_2)$, respectively

machines studied in [19]. Even for this subclass, the containment problem is undecidable [8]. It has, however, a decidable equivalence problem [19].

The main purpose of this paper is to exhibit a large class of machines for which the F-problems are decidable. A class that immediately comes to mind is the family of deterministic counter machines. However, as we have already pointed out, the disjointness and containment problems are unsolvable for these machines. Moreover, the unsolvability remains even if the machines are restricted to operate in real time. This led us to put a restriction on the counter. Initially we looked at deterministic counter machines that can only make a bounded (i.e. finite) number of reversals on the counter. We found that all the F-problems are decidable. We then generalized the model by allowing it to have two-way input and several counters [1, 7, 12], but restricted it to operate in such a way that in any accepting computation, the device makes a bounded number of reversals on the counters as well as on the input. Again, we are able to show that the F-problems are decidable for these generalized machines.

The paper has six sections in addition to this Introduction. Section 2 contains some definitions and a fundamental theorem showing that the Parikh maps [20] of languages accepted by reversal-bounded multicounter machines are semilinear.

Section 3 establishes the decidability of the F-problems for the class of deterministic two-way multicounter machines whose input and counters are reversal-bounded. It is shown that removal of the restriction on the input or counters leads to the unsolvability of the F-problems. For example, the class of deterministic machines with unrestricted two-way input and two reversal-bounded counters has unsolvable F-problems. The proof uses the undecidability of Hilbert's tenth problem [20]

Section 4 looks at some unsolvable problems concerning bounds on input reversals and counter reversals.

Section 5 investigates various decision questions concerning reversal-bounded multicounter machines augmented by an unrestricted pushdown store. It is shown that the emptiness and infiniteness problems are decidable for one-way such machines. In contrast the same problems are undecidable for one-way machines with one unrestricted counter and a pushdown store which can make at most one reversal.

Section 6 extends some of the results of the earlier sections to machines with multiple input tapes In particular it is shown that most decision problems involving nondeterministic multitape machines are decidable provided the tapes can only assume strings from bounded sets. On the other hand, it is proved that the universe problem is undecidable for nondeterministic one-way two-tape finite-state machines (without counters) one of whose tapes is unrestricted, even if the other is restricted to a unary alphabet.

Section 7 concludes with some open problems.

2. *Reversal-Bounded Multicounter Machines and Semilinear Sets*

In this section we shall prove that the Parikh map of the language accepted by a reversal-bounded multicounter machine is an effectively computable semilinear set. This result is important since it forms the basis for showing the solvability of some decision questions concerning multicounter machines.

A two-way k-counter machine is a device with a finite-state control, a two-way read-only head which operates on an input tape delimited by endmarkers, and k counters, each capable of storing any nonnegative integer At the start of the computation, the device is set to a specified initial state with the (input) head on the left endmarker and all counters set to 0. An atomic move consists of moving the input head -1, 0, $+1$ position to the right, incrementing each counter by -1, 0, $+1$, and changing the state of the finite-state control. The machine prevents the head from falling off the input tape and the counters from storing a negative count. The device is nondeterministic in that it may have several choices of next-moves on a given configuration. The input is accepted by the device if the device eventually lands in one of a designated set of accepting states. We assume that no move is possible when the machine is in an accepting state.

Formally, a two-way k-counter machine M is represented by an 8-tuple $M = \langle k, K, \Sigma, ¢, \$, \delta, q_0, F\rangle$, where K, Σ, $¢$, $\$$, q_0, F are the states, inputs, left and right endmarkers, initial state, and accepting states, respectively. δ is a mapping from $K \times (\Sigma \cup \{¢, \$\}) \times \{0, 1\}^k$ into $K \times \{-1, 0, +1\} \times \{-1, 0, +1\}^k$. A configuration of M on an input $¢x\$$, x in Σ^*,[2] is given by a $(k + 3)$-tuple $(q, ¢x\$, i, c_1, \ldots, c_k)$ denoting the fact that M is in state q with the input head reading the ith symbol of $¢x\$$, and $c_1, c_2, \ldots, c_k$ the counts (i.e. integers) stored in the k counters. We define a relation $\Rightarrow$ among configurations as follows: Write $(q, ¢x\$, i, c_1, \ldots, c_k) \Rightarrow (p, ¢x\$, i + d; c_1 + d_1, \ldots, c_k + d_k)$ if a is the ith symbol of $¢x\$$ and $\delta(q, a, \lambda(c_1), \ldots, \lambda(c_k))$ contains $(p, d, d_1, \ldots, d_k)$, where

$$\lambda(c_i) = \begin{cases} 0 & \text{if } c_i = 0, \\ 1 & \text{if } c_i \neq 0. \end{cases}$$

The reflexive-transitive closure of $\Rightarrow$ is written $\overset{*}{\Rightarrow}$. A string x in Σ^* is *accepted* by M if $(q_0, ¢x\$, 1, 0, \ldots, 0) \overset{*}{\Rightarrow} (q, ¢x\$, i, c_1, \ldots, c_k)$ for some q in F, $1 \le i \le |¢x\$|$,[3] and nonnegative integers $c_1, \ldots, c_k$. The set of strings (language) accepted by M is denoted by $T(M)$.

Without loss of generality, we assume that if $\delta(q, a, b_1, \ldots, b_k)$ contains $(p, d, d_1, \ldots, d_k)$ then $d \ge 0$ if $a = ¢$ and $d \le 0$ if $a = \$$. (This restriction prevents the input head from falling off the tape.) We also assume that $d_i \ge 0$ if $b_i = 0$. (This prevents the counters from storing a negative count.) M is deterministic if $|\delta(q, a, b_1, \ldots, b_k)| \le 1$[4] for all q in K, a in $(\Sigma \cup \{¢, \$\})$, and $b_1, \ldots, b_k$ in $\{0, 1\}$.

Let m and n be nonnegative integers. An (m, n)-reversal-bounded k-counter machine is a two-way k-counter machine which operates in such a way that in every accepting computation the input head reverses direction at most m times and the count in each counter alternately increases and decreases by at most n times. Thus, a $(0, 0)$-reversal-bounded k-counter machine is equivalent to a one-way machine which never decrements any counter. A $(0, 1)$-reversal-bounded k-counter machine is equivalent to a one-way machine which has the property that once a counter is decremented, it can never increase its count again. Note that since our bounds on reversals are only for inputs that are accepted, a two-way k-counter machine accepting a finite set is (m, n)-reversal-bounded for some m and n. In particular, if the set accepted is empty, then the machine is $(0, 0)$-reversal-bounded.

We denote the class of nondeterministic (m, n)-reversal-bounded k-counter machines by NFCM(k, m, n). Obviously, NFCM$(k, m, n) \subseteq$ NCFM(k', m', n') for all $k' \ge k$, $m' \ge m$, $n' \ge n$. The deterministic class is denoted by DFCM(k, m, n). When we have no known finite bound for the input reversal or counter reversal, we use $m = \infty$ or $n = \infty$. Thus, a (∞, ∞)-reversal-bounded k-counter machine is an unrestricted two-way k-counter machine; a two-way k-counter machine which operates in such a way that in every accepting computation each counter makes no more than n reversals is (∞, n)-reversal-bounded; etc. We use the notation NFCM(k, ∞, ∞), NFCM(k, ∞, n), etc., to denote the classes of such machines.

Let Σ be a finite set of symbols and $\alpha = \langle a_1, a_2, \ldots, a_r\rangle$ be the elements of Σ written in some order. For x in Σ^*, define the r-tuple of natural numbers $f_\alpha(x) = (\#a_1(x), \#a_2(x), \ldots, \#a_r(x))$ where $\#a_i(x)$ is the number of occurrences of symbol a_i in x. (Note that $f_\alpha(\epsilon) = (0, \ldots, 0)$.) For $L \subseteq \Sigma^*$, define $f_\alpha(L) = \{f_\alpha(x) | x \text{ in } L\}$. The mapping $f_\alpha(L)$ which takes the set L of strings into a set of r-tuples of natural numbers is called a *Parikh map* of L [22].

[2] If Σ is a finite nonempty set of symbols, Σ^* denotes the set of all finite-length strings of symbols in Σ including the null string, denoted by ϵ

[3] $|¢x\$|$ is the number of symbols in $¢x\$$.

[4] $|S|$ denotes the cardinality of set S

Let N denote the set of nonnegative integers and let N^r be the Cartesian product of N with itself r times. A subset Q of N^r is called a *linear set* if there exist $v_0, v_1, \dots, v_m$ in N^r such that $Q = \{v | v = v_0 + k_1v_1 + \cdots + k_mv_m$, each k_i in $N\}$. $v_0, v_1, \dots, v_m$ are called the *generators* of Q. Any finite union of linear sets is called a *semilinear set.* Clearly, the empty set is semilinear since it is the union of zero linear sets. It is well known [22] that if L is a context-free language (or equivalently, recognized by a nondeterministic one-way pushdown machine) then $f_\alpha(L)$ is a semilinear set.

A set $L \subseteq \Sigma^*$ is *bounded* if there exist $w_1, \dots, w_r$ in Σ^* such that $L \subseteq w_1^* \cdots w_r^*$.[5] Let $\alpha = \langle w_1, \dots, w_r\rangle$ We define $f_\alpha(L)$ by: $f_\alpha(L) = \{(i_1, \dots, i_r) | w_1^{i_1} \cdots w_r^{i_r}$ in $L\}$. In [10, 22] it is shown that if $L \subseteq w_1^* \cdots w_r^*$ is a bounded context-free language, then $f_\alpha(L)$ is a semilinear set. This result has been extended to bounded languages recognized by one-way multihead finite-state machines [25] and to bounded languages recognized by one-way multihead pushdown machines [17, 18].

We now prove the following basic result which states that languages accepted by machines in NFCM(k, 0, n) are semilinear.

THEOREM 2.1. *Let M be in NFCM(k, 0, n) and $T(M) \subseteq \{a_1, \dots, a_r\}^*$. Then $f_\alpha(T(M))$ is a semilinear set effectively computable from M ($\alpha = \langle a_1, \dots, a_r\rangle$).*

PROOF. It is sufficient to prove the theorem for $n = 1$. (If $n > 1$, we can easily construct an equivalent multicounter machine each of whose counters makes at most one reversal [1].) We may assume without loss of generality that if a string is accepted, then acceptance must occur with the input head on the right endmarker and all counters empty.

For each $1 \le i \le k$, let b_i and c_i be new symbols, and let $\Sigma = \{a_1, \dots, a_r, ¢, \$\} \cup \{b_i, c_i | 1 \le i \le k\}$. Let g be a homomorphism defined by: $g(b_i) = g(c_i) = \epsilon$ for $1 \le i \le k$, $g(a) = a$ for a in $\{a_1, \dots, a_r, ¢, \$\}$. Let L be the set of all strings y in Σ^* with the following properties:

(1) the first symbol of y is ¢;

(2) $g(y) = ¢x\$$ for some x in $\{a_1, \dots, a_r\}^*$;

(3) for each $1 \le i \le k$, any occurrence of c_i in y must appear to the right of all occurrences of b_i.

Clearly, L is a regular set. We shall construct a one-way finite automaton M' (without endmarkers) whose input alphabet is Σ. Since L is regular, we may assume that inputs to M' come from L Intuitively, an input y from L will be accepted by M' if y represents a possible accepting computation of M on $¢x\$ = g(y)$. Since M' has no counters, the action of M on the ith counter ($1 \le i \le k$), be it an increment or decrement, is represented in the string y by the occurrence of b_i or c_i, respectively. Thus, the number of occurrences of b_i in y represents the largest integer stored in counter i during the computation of M on $¢x\$$. If y represents a valid accepting computation of M on $¢x\$$, we must have: $g(y) = ¢x\$$ and for each $1 \le i \le k$, the number of occurrences of b_i in y must equal the number of occurrences of c_i. We describe the construction of M' briefly, omitting most of the details.

M' stores in its finite-state control a $(k + 2)$-tuple of the form $\langle q, \sigma, s_1, \dots, s_k\rangle$, where q is the current state of M, σ is the symbol currently under M's head, and s_i is the status of counter i (0 if empty and 1 otherwise). Initially, q is set to the initial state of M, σ is set to ¢, and s_i is set to 0, $i \le i \le k$. The tuple $\langle q, \sigma, s_1, \dots, s_k\rangle$ is updated as follows. M' determines the next move of M and executes the following steps:

(1) If M moves its input head to read a new symbol, then M' also moves its input head and checks that the new symbol under the head comes from $\{a_1, \dots, a_r, \$\}$. (If not, M' rejects y.) σ is set to the new symbol.

(2) Let $1 \le i \le k$, and suppose counter i is incremented (decremented) by 1. Then M' moves its input head and checks that the new symbol under the head is b_i (c_i) which

[5] If x and y are in Σ^*, define xy to be the string x followed by y Define x^i as follows: $x^0 = \epsilon$ and $x^{i+1} = x^ix$ for all $i \ge 0$ For convenience, we denote $\{w_1^{i_1} \cdots w_r^{i_r} | i_1, \quad , i_r \ge 0\}$ by $w_1^* \quad w_r^*$

corresponds to an increment (decrement) of 1. s_i is set to 1 if counter i is incremented by 1. If counter i is decremented by 1, s_i can be set to either 1 or 0. The choice is made nondeterministically since M' has no way of comparing the number of occurrences of b_i to that of c_i. Once s_i is set to 0, M' has guessed that M has emptied counter i. From this point on, M' must make sure that no other occurrences of b_i or c_i can appear in the remainder of the string that has yet to be processed. Step 2 is done for $i = 1, 2, \dots, k$.

(3) q is set to the next state of M, say p.

(4) If p is an accepting state, M' moves right and accepts the input (which should be exhausted by this time).

It should be clear that x is in $T(M)$ if and only if there is a y in $T(M')$ such that $g(y) = ¢x\$$ and for each $1 \le i \le k$, the number of occurrences of b_i in y is equal to the number of occurrences of c_i. Since M' is a one-way finite automaton, $f_\beta(T(M'))$ is a semilinear set effectively computable from M' [22], $\beta = \langle a_1, \dots, a_r, ¢, \$, b_1, c_1, \dots, b_k, c_k\rangle$. Let this semilinear set be Q_1. Now let

$$Q_2 = \{(l_1, \dots, l_r, 1, 1, i_1, j_1, \dots, i_k, j_k) | l_1, \dots, l_r \ge 0, i_1 = j_1 \ge 0, \dots, i_k = j_k \ge 0\}.$$

Clearly, Q_2 is a semilinear set. Then $Q_3 = Q_1 \cap Q_2$ is a semilinear set effectively computable from Q_1 and Q_2 [10]. Let Q_4 be the semilinear set obtained from Q_3 by deleting the last $2k + 2$ coordinates from the generators of the linear sets forming Q_3. Then $Q_4 = f_\alpha(T(M))$, $\alpha = \langle a_1, \dots, a_r\rangle$. □

We shall show that we can construct for every machine in NFCM(k, m, n) an equivalent machine in NFCM(k', 0, n') for some k' and n'. Thus, Theorem 2.1 generalizes to machines in the class NFCM(k, m, n). First, we give the following definition.

Definition. A machine in NFCM(k, m, n) is in *normal form* if it has the following properties: (a) The input head can only reverse direction at the endmarkers, (b) in every accepting computation the machine makes exactly m input head reversals, and (c) acceptance is only made at the endmarkers (¢ or \$ depending on whether m is odd or even).

The following technical lemma is useful.

LEMMA 2.1. *Let M_1 be in NFCM(k, m, n). We can effectively construct a machine M_2 in NFCM($k + 1$, m, $\max\{n, 2m - 1\}$) in normal form such that $T(M_1) = T(M_2)$.*

PROOF. First, we note that we can modify M_1 to a machine M_1' which makes exactly m input head reversals on inputs that are accepted. To do this, we need only incorporate in the states of M_1 a counter that counts the number of input head reversals made during the computation. If M_1 attempts to accept an input before making exactly m reversals, M_1' executes dummy input moves to bring the number of reversals to m and then accepts the input.

We now describe the construction of M_2 from M_1'. M_2 simulates M_1' until a reversal is necessary. If the reversal is made on an endmarker, the simulation continues. If M_1' makes a reversal on a symbol that is not an endmarker, M_2 first executes the following steps before it can continue the simulation: M_2 moves the input head in the direction in which it was moving before the reversal was called until the head reaches the endmarker. While doing this, M_2 uses a counter (used only for this purpose), say C, to record the position on the input from which the reversal is supposed to take place. When M_2 reaches the endmarker, it reverses direction and uses the counter C to restore the head to the position on the input from which a reversal is supposed to occur. M_2 then resumes the simulation of M_1'. M_2 can be constructed so that acceptance is only made at the endmarker (on ¢ or \$ depending on whether m is odd or even). Clearly if M_1' makes m reversals on the input, counter C makes at most $2m - 1$ reversals. It follows that M_2 is in NFCM($k + 1$, m, $\max\{n, 2m - 1\}$) in normal form. Note that M_2 is deterministic if M_1 is. □

We now describe the construction of a machine M capable of simulating, in parallel, the computation of two machines in NFCM(k, m, n).

Let $M_i = \langle k_i, K_i, \Sigma, \text{¢}, \$, \delta_i, q_{0i}, F_i\rangle$, $i = 1, 2$ be (m, n_i)-reversal-bounded k_i-counter machines in normal form. The *parallel machine* corresponding to M_1 and M_2 is the $(m, \max\{n_1, n_2\})$-reversal-bounded $(k_1 + k_2)$-counter machine $M = M_1 \otimes M_2 = \langle k_1 + k_2, K, \Sigma, \text{¢}, \$, \delta, (q_{01}, 0, q_{02}, 0), F\rangle$, where $K = \{(q_1, i, q_2, j) | q_1 \text{ in } K_1, q_2 \text{ in } K_2, i, j = -1, 0, +1\}$, F depends on the application, and δ is defined as follows:

A. Suppose $\delta_1(q_1, a, \bar{b}_1)$ contains $(p_1, d_1, \bar{c}_1)$ and $\delta_2(q_2, a, \bar{b}_2)$ contains $(p_2, d_2, \bar{c}_2)$.[6] Then

$$\delta((q_1, 0, q_2, 0), a, \bar{b}, \bar{b}_2) \text{ contains} \begin{cases} ((p_1, 0, p_2, 0), d_1, \bar{c}, \bar{c}_2) \text{ if } d_1 = d_2 \text{ (case 1)}; \\ ((p_1, d_1, p_2, 0), 0, \bar{c}_1, \bar{c}_2) \text{ if } d_1 \neq 0, d_2 = 0 \text{ (case 2)}; \\ ((p_1, 0, p_2, d_2), 0, \bar{c}_1, \bar{c}_2) \text{ if } d_1 = 0, d_2 \neq 0 \text{ (case 3)}. \end{cases}$$

B. Suppose $\delta_2(q_2, a, \bar{b}_2)$ contains $(p_2, d_2, \bar{c}_2)$. Then for all q_1 in K_1, $\bar{b}_1$ in $\{0, 1\}^{k_1}$, $d_1 \neq 0$,

$$\delta((q_1, d_1, q_2, 0), a, \bar{b}_1, \bar{b}_2) \text{ contains} \begin{cases} ((q_1, d_1, p_2, 0), 0, \bar{0}, \bar{c}_2) \text{ if } d_2 = 0 \text{ (case 1)}; \\ ((q_1, 0, p_2, 0), d_1, \bar{0}, \bar{c}_2) \text{ if } d_1 = d_2 \neq 0 \text{ (case 2)}. \end{cases}$$

C. Suppose $\delta_1(q_1, a, \bar{b}_1)$ contains $(p_1, d_1, \bar{c}_1)$. Then for all q_2 in K_2, $\bar{b}_2$ in $\{0, 1\}^{k_2}$, $d_2 \neq 0$,

$$\delta((q_1, 0, q_2, d_2), a, \bar{b}_1, \bar{b}_2) \text{ contains} \begin{cases} ((p_1, 0, q_2, d_2), 0, \bar{c}_1, \bar{0}) \text{ if } d_1 = 0 \text{ (case 1)}; \\ ((p_1, 0, q_2, 0), d_2, \bar{c}_1, \bar{0}) \text{ if } d_1 = d_2 \neq 0 \text{ (case 2)}. \end{cases}$$

$M = M_1 \otimes M_2$ has $k_1 + k_2$ counters. M simulates M_1 and M_2 in parallel using rule A, case 1, as long as the input heads of M_1 and M_2 are synchronously moving in the same direction. When the input head of M_2 falls behind that of M_1 (i.e. M_2 stays in place and does not reverse while M_1 goes on), rule A, case 2, applies. The input head of M is not moved but its finite-state control remembers that M_2's input head is lagging by recording state p_1 of M_1 and the direction of move, $d_1 \neq 0$ in the states. Rule B, case 1, is used to simulate the transitions of M_2 until its input head advances, catching up with M_1. When this happens, rule B, case 2, applies. Rule A, case 3, and rule C, cases 1 and 2, take care of the situation when M_1's input head lags that of M_2. It is clear that M is in normal form and is deterministic if M_1 and M_2 are. Note that M's counters are $\max\{n_1, n_2\}$-reversal-bounded. □

The construction above can be extended to work for any number of machines. Thus, if $M_1, \ldots, M_r$ are (m, n_i)-reversal-bounded k_i-counter machines in normal form, we can construct a $(m, \max\{n_1, \ldots, n_r\})$-reversal-bounded $(k_1 + \cdots + k_r)$-counter machine $M_1 \otimes \cdots \otimes M_r$ which simulates the computation of $M_1, \ldots, M_r$ in parallel.

Now suppose M is in NFCM(k, m, n). We shall construct a machine M^R in NFCM(k, 0, n) which has the following property: If M computes on ¢x\$ in a right-to-left scan (i.e. from \$ to ¢, without reversing its input head), then M^R will simulate the computation of M in reverse, i.e. from left to right. Incrementing (decrementing) a counter of M^R corresponds to decrementing (incrementing) the corresponding counter of M. The formal construction of M^R follows. For convenience, we assume $k = 1$. The generalization is straightforward. Suppose $M = \langle 1, K, \Sigma, \text{¢}, \$, \delta, q_0, F\rangle$. Let $M^R = \langle 1, K^R, \Sigma, \text{¢}, \$, \delta^R, \_, \_\rangle$, where $K^R = K \times (\Sigma \cup \{\text{¢}, \$\}) \times \{0, 1\}$. The initial state and accepting states are not important at this point. A state of the form (p, b, j) represents the situation in which M just entered a configuration resulting in state p, current input symbol b, and counter status j (0 for empty and 1 otherwise) We now define δ^R.

For q in K, a in $(\Sigma \cup \{\text{¢}, \$\})$, $d \leq 0$, i in $\{0, 1\}$, if $\delta(q, a, i)$ contains (p, d, t) and (*) $d = -1$ if $a = \$$, then

$$\delta^R((p, b, j), b, j) \text{ contains } ((q, a, l), -d, -t),$$

[6] $\bar{b}_1 = (b_{11}, \ \ , b_{1k_1})$, $\bar{c}_1 = (c_{11}, \ldots, c_{1k_1})$, $\bar{0} = (0, \ . \ , 0)$, etc

where

(1) $b = a$ if $d = 0$; b can be any symbol in $\Sigma \cup \{¢\}$ if $d = -1$;

(2) if $i = 1$ and $t = +1$ then $j = l = 1$; if $i = 1$ and $t = 0$ then $j = l = 1$; if $i = 1$ and $t = -1$ then $j = l = 1$ or $j = 0$ and $l = 1$; if $i = t = 0$ then $j = l = 0$; if $i = 0$ and $t = +1$ then $j = 1$ and $l = 0$

Let M start on the right endmarker of $¢x\$$ in state q and counter value c. Now suppose M moves left of \$ in one atomic move and scans the input from right to left eventually entering state p and counter value c'. (See Figure 1(a).) Then when M^R is started in state $(p, ¢, \lambda(c'))$ with counter value c' on the left endmarker of $¢x\$$, M^R will make a left-to-right scan eventually entering state $(q, \$, \lambda(c))$ and counter value c. (See Figure 1(b).) Moreover, M^R reaches \$ for the first time in state $(q, \$, \lambda(c))$, and on entering this state, M^R has no next move. (This follows from restriction (*) in the definition of δ^R). □

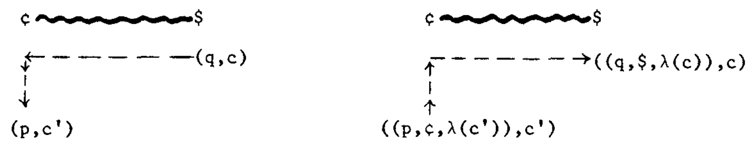
We can now prove the following result.

THEOREM 2.2. *Let M be in NFCM(k, m, n) in normal form. Then we can effectively construct a machine M' in NFCM$(k(m+1), 0, n+1)$ such that $T(M) = T(M')$.*

PROOF. To simplify discussion, we shall describe the construction when $k = 1$. The generalization for arbitrary k is straightforward. We shall assume that m is odd; the construction for m even is similar. We shall use the machine M^R in the construction of M' in NFCM$(m+1, 0, n+1)$ equivalent to $M = \langle 1, K, \Sigma, ¢, \$, \delta, q_0, F\rangle$.

For each q in K and j in $\{0, 1\}$, let $M^R_{(q,j)} = \langle 1, K^R, \Sigma, ¢, \$, \delta^R, (q, ¢, j), \_\rangle$ and $M_q = \langle 1, K, \Sigma, ¢, \$, \hat{\delta}, q, \_\rangle$, where $\hat{\delta}$ is δ with transitions of the form $(p, -1, t)$ deleted.

Let x be in $T(M)$ and consider the computation of M on $¢x\$$. Since M is in normal form, there is an accepting sequence of moves that causes M to make exactly m reversals on the input. (See Figure 2. Recall that m is odd.) M' will have $m + 1$ counters M' will simulate the computation of M on a single left to right scan of $¢x\$$. M' starts off by nondeterministically initializing the $m + 1$ counters with values $c_1 = 0$, $c_2, c_2, c_3, c_3, \ldots, c_{(m+1)/2}, c_{(m+1)/2}, c_{(m+3)/2}$ for some integers $c_2, c_3, \ldots, c_{(m+3)/2}$. (In Figure 2 M' could nondeterministically store $x_1 = 0, x_6, x_6, x_{10}, x_{10}, x_{13}$ in the 6 counters.) Next, M nondeterministically chooses $(m + 3)/2$ states: $q_1 = q_0, q_2, \ldots, q_{(m+3)/2}$ with $q_{(m+3)/2}$ in F. (In Figure 2 these states could be $q_1 = p_1 = q_0$, $q_2 = p_6$, $q_3 = p_{10}$, $q_4 = p_{13}$.) M' then simulates machines $M_{q_1}, M^R_{(q_2, \lambda(c_2))}, M_{q_2}, M^R_{(q_3, \lambda(c_3))}, M_{q_3}, \ldots, M^R_{(q_{(m+1)/2}, \lambda(c_{(m+1)/2}))}, M_{q_{(m+1)/2}}, M^R_{(q_{(m+3)/2}, \lambda(c_{(m+3)/2}))}$ in parallel on input $¢x\$$, i.e. M' simulates $M_{q_1} \otimes M^R_{(q_2, \lambda(c_2))} \otimes M_{q_2} \otimes \cdots \otimes M^R_{(q_{(m+3)/2}, \lambda(c_{(m+3)/2}))}$. Clearly, if M' made the right choices of $c_1, \ldots, c_{(m+3)/2}, q_1, \ldots, q_{(m+3)/2}$, there would be a sequence of moves that would bring the input heads of machines $M^R_{(q_2, \lambda(c_2))}, M^R_{(q_3, \lambda(c_3))}, \ldots, M^R_{(q_{(m+3)/2}, \lambda(c_{(m+3)/2}))}$ to \$. Moreover, these machines would have no next move after reaching \$ for the first time (This is because in the construction of M^R we did not consider transitions in which M does not move left on \$. See (*) in the definition of δ^R.) Let $((q'_2, \$, \lambda(c'_2)), c'_2)$, $((q'_3, \$, \lambda(c'_3)), c'_3), \ldots, ((q'_{(m+3)/2}, \$, \lambda(c'_{(m+3)/2})), c'_{(m+3)/2})$ be the state-counter configurations of these machines when they halt on \$. (In Figure 2 these are $((p_4, \$, \lambda(x_4)), x_4)$, $((p_9, \$, \lambda(x_9)), x_9)$, $((p_{11}, \$, \lambda(x_{11})), x_{11})$.) The simulation of machines $M_{q_1}, M_{q_2}, M_{q_3}, \ldots, M_{q_{(m+1)/2}}$ on \$ continues. Eventually, each of these machines will halt on \$. (Because $\hat{\delta}$ does not include transitions of the form $(p, -1, t)$.) Let $(q''_2, c''_2), (q''_3, c''_3), \ldots,$



(a) Computation of M (b) Computation of M^R

FIG 1 Computation of M and M^R Note that (p, c') need not be the first configuration entered by M on ¢

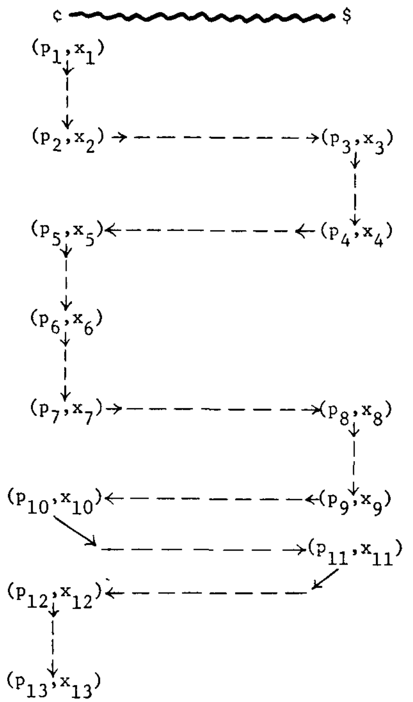


FIG 2 $m = 5$ (p_i, x_i) is a state-counter configuration $(p_1, x_1) = (q_0, 0)$, p_{13} is an accepting state

$(q''_{(m+3)/2}, c''_{(m+3)/2})$ be the state-counter configurations of these machines when they halt on \$. M' now checks that $(q''_i, c''_i) = (q'_i, c'_i)$ for each $i = 2, \ldots, (m + 3)/2$. If so, M' accepts the input; otherwise, M rejects the input. Now each of M_{q_i}, $M^R_{(q_i, \lambda(c_i))}$, is $(0, n)$-reversal-bounded. It follows that M' is $(0, n + 1)$-reversal-bounded. (The extra reversal on each counter is needed to check that $(q''_i, c''_i) = (q'_i, c'_i)$ for $i = 2, \ldots, (m + 3)/2$.) □

From Lemma 2.1 and Theorems 2.1 and 2.2 we have

THEOREM 2.3. *Let M be in NFCM(k, m, n) and $T(M) \subseteq \{a_1, \ldots, a_r\}^*$. Then $f_\alpha(T(M))$ is a semilinear set effectively computable from M ($\alpha = \langle a_1, \ldots, a_r\rangle$).*

3. *Decidable Properties of Reversal-Bounded Multicounter Machines*

This section investigates decision questions concerning the classes NFCM(k, m, n) and DFCM(k, m, n). We shall see that NFCM(k, m, n) has decidable emptiness, infiniteness, and disjointness problems. Moreover, in the case of DFCM(k, m, n) the containment and equivalence problems are also decidable. We shall demonstrate that these results are the best possible in that further generalization of the class DFCM(k, m, n) (e.g. dropping the bounded-reversal restriction on the input or counters) makes even the emptiness problem unsolvable.

First, we consider the operation of intersection of languages in NFCM(k, m, n). For this, we need the $M_1 \otimes M_2$ construction of Section 2.

LEMMA 3.1. *Let $M_i = \langle k_i, K_i, \Sigma, ¢, \$, \delta_i, q_{0i}, F_i\rangle$, $i = 1, 2$ be in NFCM(k_i, m, n_i) in normal form. We can effectively construct M_3 and M_4 in NFCM$(k_1 + k_2, m, \max\{n_1, n_2\})$ such that $T(M_3) = T(M_1) \cap T(M_2)$ and $T(M_4) = T(M_1) \cup T(M_2)$. Moreover, M_3 and M_4 are deterministic if M_1 and M_2 are.*

PROOF. Construct $M_1 \otimes M_2$, and define F by $F_1 \times \{0\} \times F_2 \times \{0\}$ for intersection and by $(F_1 \times \{0\} \times K_2 \times \{0\}) \cup (K_1 \times \{0\} \times F_2 \times \{0\})$ for union. □

From Theorem 2.3 and Lemma 3.1 and the fact that the emptiness and infiniteness problems are decidable for semilinear sets [10] we have our first decidable properties.

THEOREM 3.1. *The emptiness, infiniteness, and disjointness problems for the class NFCM(k, m, n) are decidable.*

The next result concerns complementation.

LEMMA 3.2. *Let M_1 be in DFCM(k, m, n). We can effectively construct a machine M_2 in DFCM(k, m, n) such that $T(M_2) = \overline{T(M_1)}$.*[7]

PROOF. Given $¢x\$$, M_2 simulates the computation of M_1 on $¢x\$$ and at the same time keeps track of the number of input head reversals and the number of reversals made by each counter. We consider several situations that may arise during the simulation. In each case, we describe the appropriate action of M_2.

(1) M_1 halts in an accepting state. In this case M_2 halts in a nonaccepting state

(2) M_1 halts in a nonaccepting state after making no more than m reversals on the input and no more than n reversals on any counter. In this case, M_2 halts and accepts the input.

(3) M_1 attempts to make more than m reversals on the input or more than n reversals on one of the counters. Since M_1 is (m, n)-reversal-bounded, the input could not possibly be in $T(M_1)$. Thus, in this case, M_2 halts and accepts the input.

(4) The only other situation not covered in (1)-(3) is the case when M_1 goes into an infinite computational loop without making more than m reversals on the input nor more than n reversals on any counter. For this to happen, M_1 must enter a configuration from which the input head is never again moved nor a counter decremented. M_2 is able to detect this situation by noting that if M_1 has neither moved its input head nor decremented a counter in $|K_1|$ (= number of states of M_1) atomic moves since the last time it has done either of these, then M_1 must be in an infinite loop. □

We can now state the main result of this section.

THEOREM 3.2. *The universe, containment, and equivalence problems for the class DFCM(k, m, n) are decidable.*

PROOF. It is sufficient to show that containment is decidable. Let M_1 and M_2 be in DFCM(k, m, n). Then $T(M_1) \subseteq T(M_2)$ if and only if $T(M_1) \cap \overline{T(M_2)} = \varnothing$. By Lemma 3.2, we can effectively find a machine M_3 in DFCM(k, m, n) such that $T(M_3) = \overline{T(M_2)}$. We may assume, by Lemma 2 1, that M_1 and M_3 are in normal form. By Lemma 3.1, we can construct a machine M_4 in DFCM($2k$, m, n) such that $T(M_4) = T(M_1) \cap T(M_3)$. The result now follows from Theorem 3.1. □

The universe problem is undecidable for the class of nondeterministic one-way one-counter machines which make at most one reversal on the counter [1]. Thus, Theorem 3.2 does not hold for NFCM(1, 0, 1).

In the remainder of this section we shall investigate the effect of removing the bounded-reversal restriction on the input or counters.

THEOREM 3.3. *The emptiness, infiniteness, disjointness, containment, universe, and equivalence problems (i.e. the F-problems) are undecidable for the following classes of machines: (a) DFCM(2, 0, ∞), (b) DFCM(1, ∞, ∞), (c) DFCM(1, 1, ∞).*

PROOF. The unsolvability of the F-problems for classes (a) and (b) follows from the result of Minsky [21], while that of (c) follows from Lemmas 3.3 and 3.4 below. □

LEMMA 3.3. *We can effectively find for arbitrary Turing machine M_1, machines M_2 and M_3 in DFCM(1, 0, ∞) and a homomorphism g_1 such that $T(M_1) = g_1(T(M_2) \cap T(M_3))$. Thus, the disjointness problem for the class DFCM(1, 0, ∞) is undecidable.*

PROOF. The proof of this result was implicit in [14]. □

LEMMA 3.4. *We can effectively find for arbitrary Turing machine M_1, a machine M' in DFCM(1, 1, ∞) and a homomorphism g' such that $T(M_1) = g'(T(M'))$ Hence, the F-problems for the class DFCM(1, 1, ∞) are undecidable.*

PROOF. Since only one input reversal is allowed, Lemma 3.3 does not translate directly. We describe the reduction.

Suppose M_2 and M_3 are the machines in DFCM(1, 0, ∞) such that $T(M_1) = g_1(T(M_2)$

[7] $\overline{T(M_1)} = \Sigma^* - T(M_1)$, where Σ is the alphabet of M_1

$\cap T(M_3))$. We can construct (see Section 3) a nondeterministic machine M_4 in NFCM(1, 0, ∞) such that $T(M_4) = (T(M_3))^R = \{x^R | x \text{ in } T(M_3)\}$.[8] Now modify M_4 by adding special symbols into its input alphabet. These symbols will be used to dictate the moves of the machine. Clearly, the new machine, call it M_5, is deterministic and $T(M_4) = g_2(T(M_5))$, where g_2 is the homomorphism that maps the special symbols into the null string and leaves the other symbols the same. We also modify M_2 into a machine M_6 by adding the special symbols into its inputs. Of course M_6 ignores the special symbols in its computation Then, $T(M_6) = g_2(T(M_2))$. By construction, M_5 and M_6 are in DFCM(1, 0, ∞), and $T(M_1) = g_1 g_2(T(M_6) \cap (T(M_5))^R)$. It is now trivial to construct from M_5 and M_6 a machine M' in DFCM(1, 1, ∞) such that $T(M_1) = g'\,(T(M'))$, where $g' = g_1 g_2$. Thus, the emptiness problem for the class DFCM(1, 1, ∞) is undecidable. The undecidability of the other problems follows directly. For example, to see that the infiniteness problem is undecidable, we need only note that for any machine M' in DFCM(1, 1, ∞), we can construct another machine M'' in DFCM(1, 1, ∞) accepting the set $\{d^i x | x \text{ in } T(M'), i \geq 1\}$, where d is a new symbol not in the alphabet of M'. Clearly, $T(M'')$ is infinite if and only if $T(M')$ is not empty Thus, the undecidability of the infiniteness problem follows from the unsolvability of the emptiness problem. □

The proof of our next result uses the undecidability of Hilbert's tenth problem [20]. Hilbert's tenth problem [15] is the problem of deciding for a given polynomial $P(x_1, \dots, x_n)$ with integer coefficients (i.e. a Diophantine polynomial) whether it has an integral root, i.e. integers $\alpha_1, \dots, \alpha_n$ such that $P(\alpha_1, \dots, \alpha_n) = 0$. We shall restrict $x_1, \dots, x_n$ to assume only nonnegative integers since $P(x_1, \dots, x_n)$ has a root in the integers if and only if one of the 2^n polynomials obtained by replacing some variables by their negative values has a root in the nonnegative integers

LEMMA 3.5. *Let $t(x_1, \dots, x_n) = s x_1^{i_1} \cdots x_n^{i_n}$ be a term of the polynomial $P(x_1, \dots, x_n)$, where $s = +$ or $-$, $i_1, \dots, i_n \geq 0$. Let $\Sigma = \{a_1, \dots, a_n, b\}$. We can construct a deterministic $(\infty, i_1 + \cdots + i_n)$-reversal-bounded 2-counter machine M_t which accepts an input of the form $a_1^{\alpha_1} \cdots a_n^{\alpha_n} b^{\beta}$, where $\alpha_1, \dots, \alpha_n$ are nonnegative integers, if and only if $\beta = \alpha_1^{i_1} \cdots \alpha_n^{i_n}$.*

PROOF. The exponents $i_1, \dots, i_n$ are stored in the states of M_t. Assume that each $i_j \geq 1$. (Otherwise, ignore the exponent.) M_t scans the $a_1^{\alpha_1}$ segment and stores integer α_1 in the first counter. Then it computes α_1^2 in the second counter by making α_1 passes on the $a_1^{\alpha_1}$ segment and adding α_1 to the second counter on each pass. Also, M_t decrements the first counter by 1 on each pass. Hence, a zero in the first counter indicates that M has made exactly α_1 passes on the $a_1^{\alpha_1}$ segment. By iterating the process and alternately switching the roles of the counters, M_t can compute $\alpha_1^{i_1}\alpha_2^{i_2} \cdots \alpha_n^{i_n}$ in one of the counters in $i_1 + i_2 + \cdots + i_n - 1$ counter reversals. (The input head makes the passes on the $a_j^{\alpha_j}$ segment when M_t is computing $\alpha_1^{i_1} \cdots \alpha_{j-1}^{i_{j-1}}\alpha_j^k$, $k = 1, 2, \dots, i_j$.) After M_t has computed $\alpha_1^{i_1}\alpha_2^{i_2} \cdots \alpha_n^{i_n}$ in one of the counters, it verifies that $\beta = \alpha_1^{i_1} \cdots \alpha_n^{i_n}$. □

THEOREM 3.4 *The F-problems are undecidable for the following classes of machines: (a) $\bigcup_{n \geq 1}$ DFCM(2, ∞, n), (b) $\bigcup_{k \geq 1}$ DFCM(k, ∞, 1).*

PROOF. It is sufficient to prove that the F-problems are unsolvable for class (a) since a multicounter machine which makes at most n reversals on each counter can be converted to an equivalent machine which makes at most 1 reversal on each counter.

Let $P(x_1, \dots, x_n)$ be a Diophantine polynomial. Let $P(x_1, \dots, x_n) = t_1(x_1, \dots, x_n) + \cdots + t_r(x_1, \dots, x_n)$ where each $t_j(x_1, \dots, x_n)$ is a term. For each $1 \leq j \leq r$, let $t_j(x_1, \dots, x_n) = s_j x_1^{i_{j1}} \cdots x_n^{i_{jn}}$, where s_j is $+$ or $-$, $i_{j1}, \dots, i_{jn} \geq 0$. We shall construct a deterministic (∞, l)-reversal-bounded 2-counter machine M_P accepting the language $L = \{a_1^{\alpha_1} \cdots a_n^{\alpha_n} b^{\beta_1} \# b^{\beta_2} \# \cdots \# b^{\beta_r} | \alpha_1, \dots, \alpha_n \geq 0, \beta_j = \alpha_1^{i_{j1}} \cdots \alpha_n^{i_{jn}}, s_1\beta_1 + \cdots + s_r\beta_r = 0\}$. The integers $i_{11}, \dots, i_{1n}, \dots, i_{r1}, \dots, i_{rn}$ and signs $s_1, \dots, s_r$ are stored in the states of M_P. M_P uses the technique described in Lemma 3.5 to check that $\beta_j = \alpha_1^{i_{j1}} \cdots \alpha_n^{i_{jn}}$ for $j = 1, 2, \dots, r$ and then verifies that $s_1\beta_1 + \cdots + s_r\beta_r = 0$. (The last step needs only 1

[8] x^R is the reverse of string x

reversal on the counter.) Clearly, M_P is (∞, l)-reversal-bounded where $l = \sum_{j=1}^{r} i_{j1} + \cdots + i_{jn}$, and $T(M_P) \neq \varnothing$ if and only if $P(x_1, \ldots, x_n)$ has a solution in the nonnegative integers. The undecidability of the emptiness problem now follows from the undecidability of Hilbert's tenth problem [20]. The unsolvability of the other F-problems follow easily. □

We are unable to extend Theorem 3.4 to the class $\bigcup_{n\geq 1}$ DFCM$(1, \infty, n)$. There does not seem to be a way to reduce Hilbert's tenth problem to the emptiness problem for this class. If, on the other hand, the emptiness problem is solvable, the proof would probably not involve semilinear sets since there are languages accepted by machines in DFCM$(1, \infty, 1)$ (e.g. $L = \{a^m b^{km} \mid k, m \geq 1\}$) whose Parikh maps are not semilinear.

The equivalence problem for the class DFCM(1, 0, ∞) is solvable [27]. However, the disjointness and containment problems are unsolvable. This follows from Lemma 3.3 and the fact that the class of languages defined by DFCM(1, 0, ∞) is closed under complementation.

4. *Unsolvable Problems Concerning Reversal Bounds*

In this section we shall look at some questions concerning bounds on input reversals and counter reversals. We shall see that almost all problems are unsolvable.

THEOREM 4.1. *Each of the following problems is recursively unsolvable for arbitrary machine M in DFCM*(2, 0, ∞):

(*a*) *Is M in DFCM*$(2, 0, n)$ *for some n?*

(*b*) *For a fixed n, is M in DFCM*$(2, 0, n)$?

PROOF. Let M_1 be an arbitrary machine in DFCM(2, 0, ∞) and d be a symbol not in the input alphabet of M_1. Construct a machine M_2 in DFCM(2, 0, ∞) which when presented with an input string of the form $d^i x$, $i \geq 1$, first scans the d^i segment and makes i reversals on each of the counters. M_2 then simulates the computation of M_1 on x and accepts $d^i x$ if and only if M_1 accepts x. If $T(M_1) = \varnothing$ then $T(M_2) = \varnothing$, and by definition, M_2 is in DFCM(2, 0, 0). If $T(M_1) \neq \varnothing$ then for each x in $T(M_1)$, M_2 accepts $d^i x$, $i = 1, 2, \ldots$. By construction, M_2 makes at least i reversals on each counter in processing $d^i x$. It follows that M_2 is in DFCM$(2, 0, n)$ for some n if and only if M_2 is in DFCM(2, 0, 0), and if and only if $T(M_1) = \varnothing$. The result follows, since the emptiness problem is unsolvable for the class DFCM(2, 0, ∞) (Theorem 3.3). □

If M is in NFCM(1, 0, ∞), we can decide if M is in NFCM$(1, 0, n)$ for some n. Moreover, such an n can be found effectively, if it exists. This follows from the decidability of similar questions concerning finite-turn pushdown machines [11]. The situation is different for machines in DFCM(1, 1, ∞) as stated in the next theorem.

THEOREM 4.2. *Same as Theorem* 4.1 *with DFCM*(2, 0, ∞) *replaced by DFCM*(1, 1, ∞).

PROOF. Same as in Theorem 4.1 plus the fact that the emptiness problem for the class DFCM(1, 1, ∞) is unsolvable (Theorem 3.3). □

For the class DFCM(1, ∞, ∞), we have

THEOREM 4.3. *Each of the following problems is recursively unsolvable for arbitrary machine M in DFCM*(1, ∞, ∞):

(*a*) *Is M in DFCM*$(1, m, n)$ *for some m and n?*

(*b*) *For fixed m and n, is M in DFCM*$(1, m, n)$?

PROOF. The proof is similar to that of Theorem 4.1. If M_1 is in DFCM(1, 1, ∞), construct a machine M_2 in DFCM(1, ∞, ∞) which when given an input of the form $d^i x$ makes i reversals on the input and i reversals on the counter, and then M_2 simulates M_1. The result follows from the unsolvability of the emptiness problem for the class DFCM(1, 1, ∞). □

Another result is this:

THEOREM 4.4. *The following problems are unsolvable for M in $\bigcup_{n\geq 1}$ DFCM*$(2, \infty, n)$:

(*a*) *Is M in DFCM*$(2, m, n)$ *for some m*?

(*b*) *For a fixed m, is M in DFCM*$(2, m, n)$?

PROOF Same as in Theorem 4.3 using the fact that the emptiness problem for machines in $\bigcup_{n\geq 1}$ DFCM$(2, \infty, n)$ is unsolvable (Theorem 3.4). □

We conclude this section with a decidable property.

THEOREM 4.5. *Let m, n be nonnegative integers. It is decidable to determine for an arbitrary machine M in NFCM*(k, ∞, ∞) *whether there is an input (not necessarily accepted) that will cause M to make either more than m input reversals or more than n counter reversals. Moreover, we can decide if there is an infinite number of such inputs.*

PROOF. Construct a machine M' in NFCM(k, m, n) which on input ¢x\$ simulates the computation of M on ¢x\$. M' accepts only those inputs that cause M to exceed either the input reversal bound or the counter reversal bound. The result follows since we can decide if $T(M')$ is empty or infinite (Theorem 3.1). □

5. *Reversal-Bounded Multicounter Machines Augmented by a Pushdown Store*

Some of the results of Sections 2 and 3 remain valid for a more general class of multicounter machines. These are machines augmented by a pushdown store. We denote by NPCM(k, m, n) the class of nondeterministic (m, n)-reversal-bounded k-counter machines augmented by an unrestricted pushdown store. DPCM(k, m, n) will denote the deterministic class. Other notations, e.g. NPCM(k, m, ∞), NPCM(k, ∞, n), etc., will also be used.

We begin with the following analogue of Theorem 2.1 for the class NPCM$(k, 0, n)$.

THEOREM 5.1. *Let M be in NPCM*$(k, 0, n)$ *and* $T(M) \subseteq \{a_1, \ldots, a_r\}^*$. *Then* $f^\alpha(T(M))$ *is a semilinear set effectively computable from M* ($\alpha = \langle a_1, \ldots, a_r\rangle$).

PROOF. The proof of Theorem 2.1 holds when the machines M and M' are augmented by a pushdown store. □

We cannot improve Theorem 5.1 by allowing M to be in NPCM(k, m, n), $m \geq 1$. To see this, consider the language $L = \{a^1b^2a^3b^4 \cdots a^{2k-1}b^{2k} | k \geq 1\}$. Clearly, L can be accepted by a deterministic pushdown machine (without counters) that makes exactly one reversal on the input But $f_{\langle a,b\rangle}(L) = \{(n^2, n^2 + n) | n \geq 1\}$, which is not semilinear.

Since the emptiness and infiniteness problems for semilinear sets are decidable [10], we have

THEOREM 5.2. *The emptiness and infiniteness problems for the class NPCM*$(k, 0, n)$ *are decidable*.

The following corollary is a generalization of Theorem 5 of [1].

COROLLARY 5.1. *Let M be in NPCM*(k, ∞, n). *Then* $T(M)$ *is a recursive set*.

PROOF Let x be an input to M. We can effectively construct a machine M_x in NPCM$(k, 0, n)$ such that $T(M_x) \neq \varnothing$ if and only if M accepts x. To do this, we encode the string ¢x\$ in the states of M_x. The actions of the input head of M on ¢x\$ are simulated by M_x in its states. M_x enters an accepting state if and only if M accepts x. The result follows from Theorem 5.2. □

Remarks.

(1) We can construct for every Turing machine M_1 a machine M_2 in DPCM(0, 1, 0) which makes at most three reversals on the pushdown such that $T(M_2) \neq \varnothing$ if and only if M_1 halts on an initially blank tape (See, e.g., the proof of [1, Theorem 1].) It follows that Theorem 5.2 does not hold for machines in DPCM(0, 1, 0), even if the machines are restricted to make at most three reversals on the pushdown.

(2) In [1] it is shown that if M_1 is a Turing machine, we can effectively find deterministic one-way pushdown machines M_2 and M_3 with the property that the pushdown store reverses only once, and $T(M_1) = g(T(M_2) \cap T(M_3))$ for some homomorphism g Hence, the containment and disjointness problems for this class of machines are undecidable. However, it is shown in [26] that the equivalence problem is decidable. The status of the equivalence problem for the full class of one-way deterministic pushdown machines is still open.

Every machine in the class NPCM(k, 0, n) has the property that the counters are reversal-bounded and the pushdown store is unrestricted. Now consider the class of one-way machines with one unrestricted counter and one pushdown store which makes at most one reversal.[9] Let NZ (DZ) denote the nondeterministic (deterministic) class. One might suspect that a result similar to Theorem 5.2 can be shown for the class DZ. However, we have the following negative result.

THEOREM 5.3. *Let M be a single-tape Turing machine. We can effectively construct a machine M'' in DZ such that $T(M) = g(T(M''))$ for some homomorphism g. Thus, the emptiness and infiniteness problems for the class DZ are unsolvable.*

PROOF. By Lemma 3.4, we can find for a given Turing machine M a machine M' in DFCM(1, 1, ∞) such that $T(M) = g(T(M'))$ for some homomorphism g. The desired machine M'' in DZ is constructed from M' as follows: M'' simulates the computation of M' on a given input and at the same time copies the input on the pushdown store. When the input head of M' reverses, M'' can continue the simulation using the pushdown store. Clearly, $T(M) = g(T(M''))$. □

COROLLARY 5 2. *The class of languages accepted by machines in NZ is precisely the class of recursively enumerable sets.*

PROOF. From the preceding theorem and the observation that NZ is closed under homomorphism. □

The class DZ contains only recursive sets as the following theorem shows. The result generalizes [1, Theorem 3].

THEOREM 5.4. *Let DZP(k, m, n) be the class of deterministic two-way k-pushdown store machines which operate in such a way that in every accepting computation the input head makes at most m reversals and each of the first $k - 1$ stores makes at most n reversals. (Thus, one pushdown store is unrestricted.) DZP(k, m, n) is effectively closed under complementation. It follows that DZP(k, m, n) and, hence, DZP(k, ∞, n) define only recursive sets.*

PROOF. We modify the construction of M_2 in the proof of Lemma 3.2. Cases (1)–(3) carry over with "any counter" replaced by "any of the first $k - 1$ pushdown stores." In case (4) the pushdown store may grow indefinitely or may be bounded in length but some pushdown configuration is repeated. M_2 can detect this situation using the technique of [16, Lemma 12.1] for showing closure under complementation of the class of languages accepted by deterministic one-way pushdown machines. □

COROLLARY 5.3. *Let M_1 be in DPCM(k, m, n). We can effectively construct a machine M_2 in DPCM(k, m, n) such that $T(M_2) = \overline{T(M_1)}$.*

To prove the next theorem, we need the following lemma.

LEMMA 5.1. *Let M_1 be in NPCM(k_1, 0, n_1) and M_2 be in NFCM(k_2, 0, n_2). Then we can find M_3 and M_4 in NPCM($k_1 + k_2$, 0, max$\{n_1, n_2\}$) such that $T(M_3) = T(M_1) \cup T(M_2)$ and $T(M_4) = T(M_1) \cup T(M_2)$.*

PROOF. The proof is similar to that of Lemma 3.1. The construction of $M_1 \otimes M_2$ in Section 2 still works for one-way machines even if one machine has a pushdown store. □

THEOREM 5.5. *The question, "Is $T(M_1) \subseteq T(M_2)$?" is decidable for*

(a) M_1 in NPCM(k_1, 0, n_1) and M_2 in DFCM(k_2, m_2, n_2),

(b) M_1 in NFCM(k_1, m_1, n_1) and M_2 in DPCM(k_2, 0, n_2).

PROOF. $T(M_1) \subseteq T(M_2)$ if and only if $T(M_1) \cap \overline{T(M_2)} = \varnothing$. (a) follows from Lemma 3.2, Theorem 2.2, Lemma 5.1, and Theorem 5.2. (b) follows from Corollary 5.3, Lemmas 3.2 and 5 1, and Theorem 5.2. □

The next result is immediate from Theorem 5.5.

COROLLARY 5.4. *It is decidable to determine for arbitrary M_1 in DFCM(k_1, m_1, n_1) and M_2 in DPCM(k_2, 0, n_2) whether $T(M_1) = T(M_2)$.*

[9] We ignore the case when the pushdown store is also unrestricted or the case when there are two or more unrestricted counters Such machines are as powerful as Turing machines [21]. (See also Theorem 3 3)

The remainder of this section concerns machines in NPCM(k, m, n) that accept bounded languages.

LEMMA 5.2. *Let M be in NPCM(k, m, n) and $T(M) \subseteq a_1^* \cdots a_r^*$, where $a_1, \ldots, a_r$ are distinct symbols. Then $f_\alpha(T(M))$ is a semilinear set effectively computable from M ($\alpha = \langle a_1, \ldots, a_r\rangle$).*

PROOF. We may assume that M is in normal form (see Lemma 2.1). It is sufficient to construct a machine M' in NPCM($r(m + 1) + k$, 0, n) such that $T(M) = T(M')$, by Theorem 5.1.

Let $a_1^{i_1} \cdots a_r^{i_r}$ be an input to M. Since M is in normal form, each segment $a_j^{i_j}$ is scanned exactly $m + 1$ times. M' will have $r(m + 1)$ counters in addition to the k counters that will simulate the counters of M. M' begins by reading the input and storing integer i_j in each of the counters in the jth set of $m + 1$ counters, $j = 1, 2, \ldots, r$. Then M' simulates the computation of M on ¢$a_1^{i_1} \cdots a_r^{i_r}$ \$ using the $r(m + 1)$ counters whose initial values are $i_1, \ldots, i_1, \ldots, i_r, \ldots, i_r$. The jth set of $m + 1$ counters is used in simulating the computation of M on the segment $a_j^{i_j}$. A left-to-right (or right-to-left) scan of $a_j^{i_j}$ by M is simulated by using one of the counters in the jth set of counters whose value is i_j. Another scan of $a_j^{i_j}$ at a later time will use a different counter in the jth set. It is clear that $T(M') = T(M)$. Moreover, M' is deterministic if M' is. □

THEOREM 5.6. *Let M be in NPCM(k, m, n) and $T(M) \subseteq w_1^* \cdots w_r^*$. Then $f_\alpha(T(M))$ is a semilinear set effectively computable from M ($\alpha = \langle w_1, \ldots, w_r\rangle$).*

PROOF. Given M and $w_1, \ldots, w_r$, we can construct another machine M' in NPCM(k, m, n) such that $T(M') = \{a_1^{i_1} \cdots a_r^{i_r} \mid w_1^{i_1} \cdots w_r^{i_r}$ *in* $T(M)\}$, where $a_1, \ldots, a_r$ are distinct symbols. M' need only code $w_1, \ldots, w_r$ in the states and simulate the computation of M on w_i in the states. □

COROLLARY 5.5. *The emptiness, infiniteness, disjointness, containment, and equivalence problems for machines in NPCM(k, m, n) which accept subsets of $w_1^* \cdots w_r^*$ are decidable.*

PROOF. This follows from Theorem 5.6 and the fact that the said problems are decidable for semilinear sets [10]. For example, to determine if $T(M_1) \subseteq T(M_2)$ we can find the semilinear sets Q_i such that $Q_i = f_\alpha(T(M_i))$, $\alpha = \langle w_1, \ldots, w_n\rangle$, $i = 1, 2$. Then $T(M_1) \subseteq T(M_2)$ if and only if $Q_1 \subseteq Q_2$ which is decidable [10]. □

6. *Decidable Properties of Multitape Multicounter Machines*

We apply the results of the previous sections to some decision problems concerning two-way multitape multicounter machines These are machines with t (greater than or equal to 1) input tapes (with endmarkers), each with an independent two-way read head. Thus, the sets accepted by such machines are t-tuples of strings, i.e. relations. We omit the formal definitions We shall use the notation NMFCM(k, m, n, t), NMPCM(k, m, n, t), etc., to denote the multitape classes of machines. Thus, a machine in NMFCM(k, m, n, t) is a nondeterministic two-way t-tape k-counter machine which operates in such a way that in every accepting computation each input head makes at most m reversals and each counter makes at most n reversals. Machines in NMPCM(k, m, n, t) are provided with a single unrestricted pushdown store.

Multitape machines (without counters) have been studied in several places [3, 5, 6, 13, 23, 24]. Properties of the finite-state variety can be found in [3, 5, 6, 23, 24] while those of the pushdown type can be found in [13]. It is well known that the equivalence problem for nondeterministic one-way t-tape ($t \geq 2$) (finite-state) machines without counters is undecidable [24] and so is the containment problem for the deterministic case [24]. The status of the equivalence problem for the deterministic case is still open, although the problem is known to be decidable for $t \leq 2$ [3]. For the deterministic two-way varieties (without counters), one-input reversal is sufficient to make the emptiness problem unsolvable [23].

Here, we shall show that the emptiness, containment, and equivalence problems are decidable for some restricted classes of multitape machines.

We begin with the following lemma. The lemma is a generalization of a similar result for one-way multitape machines without counters [23].

LEMMA 6.1. *Let M_1 be in NMPCM$(k, 0, n, t)$ (NMFCM$(k, 0, n, t)$) and $1 \le i \le t$. We can effectively construct a machine M_2 in NPCM$(k, 0, n)$ (NFCM$(k, 0, n)$) such that $T(M_2) = \{x_i | \text{for some } x_1, \dots, x_t \text{ in } \Sigma^*, (x_1, \dots, x_t) \text{ is in } T(M_1)\}$.*

PROOF. M_2 simulates the computation of M_1 on $(¢x_1\$, \dots, ¢x_t\$)$ by guessing the strings $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_t$. We omit the details. □

Lemma 6.1 does not hold for two-way machines, as the following proposition shows.

PROPOSITION 1. *There is a machine M in DMFCM$(0, 1, 0, 3)$, i.e. M is a deterministic two-way 3-tape machine without counters which makes at most one reversal on each input tape such that the set $L_1 = \{x_1 | (x_1, x_2, x_3) \text{ in } T(M)\}$ cannot be accepted by any machine in NPCM(k, m, n) for any k, m, n.*

PROOF. Consider the set $W = \{a^i, x, x) | x = a^1b^2a^3b^4 \cdots a^{2k-1}b^{2k}, i = 1 + 3 + \cdots + 2k - 1, k \ge 1\}$. Clearly, W can be accepted by a machine in DMFCM(0, 1, 0, 3). Now $L_1 = \{x_1 | (x_1, x_2, x_3) \text{ in } W\} = \{a^{n^2} | n \ge 1\}$. L_1 is not semilinear and, by Theorem 5.6, L_1 cannot be accepted by any machine in NPCM(k, m, n). □

COROLLARY 6.1. *There is a set Y accepted by a machine in DMFCM$(0, 1, 0, 2)$ such that $L_1 = \{x_1 | (x_1, x_2) \text{ in } Y\}$ cannot be accepted by any machine in NFCM(k, m, n) for any k, m, n.*

PROOF. Let $Y = \{(x, x) | x = a^1b^2a^3b^4 \cdots a^{2k-1}b^{2k}, k \ge 1\}$. Then $f_{\langle a,b \rangle}(L_1) = \{(n^2, n^2 + n) | n \ge 1\}$ is not semilinear. By Theorem 2.3, L_1 cannot be accepted by any machine in NFCM(k, m, n). □

Remark. Proposition 1 is not true for machines in DMFCM(0, 1, 0, 2), for we can show that if M is in NMFCM$(k, 1, n, 2)$ then the set $L_1 = \{x_1 | (x_1, x_2) \text{ in } T(M)\}$ can be accepted by a machine M' in NPCM$(k, 1, n)$. Briefly, M' operates as follows: Given $¢x_1\$$, M' simulates the computation of M on $(¢x_1\$, ¢x_2\$)$ by guessing the symbols of x_2 and storing them in the pushdown store. M' uses the pushdown store when M reverses on the second tape.

From Lemma 6.1, we obtain the following theorem. Again, the proof technique has been used in [23].

THEOREM 6.1. *The emptiness and infiniteness problems for machines in NMPCM$(k, 0, n, t)$ are decidable.*

PROOF. Let M be in NMPCM$(k, 0, n, t)$. For each $1 \le i \le t$, let $L_i = \{x_i | (x_1, \dots, x_n) \text{ in } T(M)\}$. By Lemma 6.1, we can effectively construct M_i in NPCM$(k, 0, n)$ such that $T(M_i) = L_i$. Then $T(M) = \varnothing$ if and only if $T(M_1) = \varnothing$. $T(M)$ is infinite if and only if $T(M_i)$ is infinite for some $1 \le i \le t$. The result follows since the emptiness and infiniteness problems for machines in NPCM$(k, 0, n)$ are decidable. □

In contrast to Theorem 6.1, we have the following proposition, which was shown in [23].

PROPOSITION 2. *The emptiness and infiniteness problems are undecidable for machines in DMFCM$(0, 1, 0, 2)$ (i.e. deterministic 2-tape machines without counters that make at most one reversal on each tape).*

LEMMA 6.2. *Let M_1 be in NMFCM(k, m, n, t). Suppose $T(M_1) \subseteq \Sigma^* \times \times_{i=1}^{t-1} a_1^* \cdots a_r^*$,[10] where $\Sigma = \{a_1, \dots, a_r\}$ is the input alphabet of M_1. We can effectively construct for some k', m', n' a machine M_2 in NFCM(k', m', n') such that $T(M_2) = \{x_1\#x_2\# \cdots \#x_t | (x_1, \dots, x_t) \text{ is in } T(M_1)\}$. If M_1 is deterministic then so is M_2. (# is a new symbol not in Σ.)*

PROOF. We may assume that M_1 is in normal form. (Extend the definition of normal form to multitape machines.) We describe the construction of M_2 for the case $t = 2$. M_2 will have $r(m + 1)$ counters in addition to the k counters needed to simulate the k counters of M_1. Given an input $¢y\$$, M_2 first checks that y is of the form $x\#a_1^{i_1} \cdots a_r^{i_r}$ for some x in Σ^* and $i_1, \dots, i_r \ge 0$. While checking, M_2 stores integer i_j in each of

[10] $\times_{i=1}^{t-1} a_1^* \cdots a_r^*$ denotes the Cartesian product of $a_1^* \cdots a_r^*$ $t - 1$ times

the counters of the jth set of $m + 1$ counters, $j = 1, 2, \ldots, r$. M_2 then moves its input head to the left endmarker and begins simulating the computation of M_1 on $(¢x\$, ¢a_1^{i_1} \cdots a_r^{i_r}\$)$. Computation of M_1 on the tape $¢a_1^{i_1} \cdots a_r^{i_r}\$$ is simulated using the $m + 1$ copies of $i_1, \ldots, i_r$ stored in the $r(m + 1)$ counters. (See Lemma 5.2.) □

We note that Lemma 6.2 remains valid even if $T(M_1) \subseteq \Sigma^* \times \times_{i=1}^{t-1} w_1^* \cdots w_r^*$, where $w_1, \ldots, w_r$ are strings in Σ^*. (See the proof of Theorem 5.6.) Thus from Theorems 3.1 and 3.2 we have the next result.

THEOREM 6.2. (*a*) *The emptiness, infiniteness, and disjointness problems are decidable for machines in NMFCM*(k, m, n, t) *satisfying the property that they only accept subsets of* $\Sigma^* \times \times_{i=1}^{t-1} w_1^* \cdots w_r^*$. (*b*) *Moreover, the containment and equivalence problems are decidable for deterministic such machines.*

We now show that Theorem 6.2(b) does not hold for the nondeterministic case, even if no counters are allowed. In [24] it is shown that the equivalence problem for nondeterministic one-way t-tape finite-state machines is undecidable ($t \geq 2$). However, the proof in [24] does not remain valid for the class of machines whose inputs come from $\Sigma^* \times \times_{i=1}^{t-1} w_1^* \cdots w_r^*$. Our next result, which is rather surprising, shows that the universe problem is unsolvable even for a very restricted class of machines.

THEOREM 6.3. *Let* $\mathcal{T} = \{M | M$ *is a nondeterministic one-way 2-tape finite-state machine such that* $T(M) \subseteq \{0, 1\}^* \times 1^*\}$. *The universe, containment, and equivalence problems for the class* $\mathcal{T}$ *are undecidable.*

PROOF. It is sufficient to show the undecidability of the universe problem. We shall show how we can reduce the halting problem for single-tape Turing machines to the universe problem for the class $\mathcal{T}$. Specifically, we shall show that if M is a Turing machine, we can construct a machine M'' in $\mathcal{T}$ such that $T(M'') = \{0, 1\}^* \times 1^*$ if and only if M does not halt on an initially blank tape.

Let M be a single-tape Turing machine and K be its set of states. Assume without loss of generality that M's tape alphabet consists of 0, 1, b (for blank). We may also assume that M never overwrites a symbol by a blank. Hence, any configuration of M can be written as $bxqyb$, where x, y are in $\{0, 1\}^*$ and q in K. The initial configuration is bq_0b, where q_0 is the initial state of M. We assume that q_0 is not a halting state. We shall construct a 2-tape machine M' with input alphabet $\Sigma = \{0, 1, b, \#\} \cup K$ (# is a new symbol) such that $T(M') = \Sigma^* \times 1^*$ if and only if M does not halt. By standard coding techniques, we can easily modify M' to a machine M'' in $\mathcal{T}$ satisfying $T(M'') = \{0, 1\}^* \times 1^*$ if and only if M does not halt. First, we describe the construction of two machines, M_1 and M_2.

Let $R = \{(x, 1^r) | r \geq 0, x = \#\mathrm{ID}_1\# \cdots \#\mathrm{ID}_k\#$ for some $k \geq 2$ and configurations $\mathrm{ID}_1, \ldots, \mathrm{ID}_k$ of M, ID_1 is the initial configuration of M, ID_k is a halting configuration of M, and $r = |x|\}$. Clearly, we can construct a 2-tape finite-state machine M_1 such that $T(M_1) = (\Sigma^* \times 1^*) - R$

Next, we construct a 2-tape finite-state machine M_2 which accepts a 2-tuple of the form $(x, 1^r)$ if (1) $x = \#\mathrm{ID}_1\# \cdots \#\mathrm{ID}_k\#$ for some $k \geq 2$ and configurations $\mathrm{ID}_1, \ldots, \mathrm{ID}_k$ of M, (2) ID_1 is the initial configuration of M, (3) ID_k is a halting configuration of M, and either (4) $r \neq |x|$ or (5) $r = |x|$ and for some $i < k$, ID_{i+1} is not the proper successor of ID_i. M_2 needs only its finite-state control to check (1), (2), and (3). Now M_2 may guess that $r \neq |x|$ and check the guess by reading across both tapes, thus verifying (4). Otherwise, M_2 does the following (H_1 and H_2 denote the tape heads of $x = \#\mathrm{ID}_1\# \cdots \#\mathrm{ID}_i\#\mathrm{ID}_{i+1}\# \cdots \#\mathrm{ID}_k\#$ and 1^r, respectively): M_2 moves H_1 and H_2 to the right simultaneously until H_1 reaches the # immediately to the left of some ID_i, $1 \leq i < k$. Then M_2 moves H_1 some number s of squares to the right and guesses that an "error" occurs in positions s, $s + 1$, or $s + 2$ of ID_i and ID_{i+1}. M_2 uses its finite-state control to remember these symbols of ID_i as it moves H_1 and H_2 to the right, stopping when H_1 is at the next #. Now, M_2 moves H_1 and H_2 to the right, moving H_2 two places for each move of H_1. At some point, M_2 guesses that the number t of squares crossed by H_1 is s and checks the symbols at positions t, $t + 1$, $t + 2$ on tape 1 to see if they are appropriate for the successor of ID_t if $t = s$. Then M_2 moves H_1 and H_2 to the right at the same speed. If H_1 and H_2 reach the right end of their respective tapes at the same time, either $r \neq |x|$ or $r = |x|$ and $t = s$. Therefore, if H_1 and H_2 reach the end of their tapes at the same time and the t, $t + 1$, $t + 2$ symbols were not appropriate, M_2 accepts.

Now $T(M_1) \cup T(M_2) = \Sigma^* \times 1^*$ if and only if $R \subseteq T(M_2)$. But from the construction of M_2, $R \subseteq T(M_2)$ if and only if M does not halt. The desired machine M' is now constructed from M_1 and M_2 so that $T(M') = T(M_1) \cup T(M_2)$. □

Remark. Theorem 6.3 holds even if the machines have no endmarkers. In the proof the machines can simply guess the ends of the tapes.

In view of Theorem 6.3 the following result is the best possible.

THEOREM 6.4. *The emptiness, infiniteness, disjointness, containment, and equivalence problems are decidable for machines in NMPCM(k, m, n, t) with the property that they only accept subsets of* $\times_{i=1}^{t} w_1^* \cdots w_r^*$.

PROOF. The proof is similar to that of Theorem 6.2, this time using Corollary 5.5. □

7. *Conclusions*

We have shown that the F-problems (i.e. emptiness, infiniteness, disjointness, containment, universe, and equivalence problems) are decidable for the class of deterministic two-way multicounter machines with reversal-bounded input and counters. This result is the best possible in that dropping the bounded-reversal restriction on the input or counters makes all the F-problems undecidable We have also investigated the boundary points between decidability and undecidability of various decision questions for several related classes of machines, in some instances improving previously known results. Among the interesting questions that remain unresolved are the following:

(a) Which of the F-problems are decidable for the class $\bigcup_{n \geq 1}$ DFCM(1, ∞, n)?

(b) Are the emptiness, infiniteness, and disjointness problems decidable for the class $\bigcup_{n \geq 1}$ NFCM(1, ∞, n)? Note that the universe problem is already undecidable for the class NFCM(1, 0, 1).

(c) By Theorem 3.4, the F-problems are unsolvable for machines in $\bigcup_{n \geq 1}$ DFCM(2, ∞, n) accepting only bounding languages (i.e. subsets of $a_1^* \cdots a_r^*$ for some $r \geq 1$ and distinct symbols $a_1 \ldots, a_r$). In a forthcoming paper, we shall show that the F-problems are decidable for machines in the class $\bigcup_{k,n \geq 1}$ DFCM(k, ∞, n) whose input alphabet consists only of one letter. Does this latter result generalize to the nondeterministic case?

(d) Is the equivalence problem for the class of deterministic one-way t-tape finite-state machines decidable? The case $t = 1$ is trivial, and the case $t = 2$ has already been shown decidable [3].

(e) Is the equivalence problem for the class of deterministic one-way pushdown machines decidable? [26] contains a proof that for reversal-bounded pushdown machines, the problem is decidable.

ACKNOWLEDGMENTS. In an earlier version of this paper, M_2 of Theorem 6.3 was a 3-tape machine with $T(M_2) \subseteq \Sigma^* \times 1^* \times 1^*$, and the theorem was only proved for 3-tape machines. I am grateful to one of the referees for pointing out a modification of our earlier construction of M_2 leading to the 2-tape machine described here.

REFERENCES

1 BAKER, B , AND BOOK, R Reversal-bounded multipushdown machines *J Comptr and Syst Sci. 8* (1974), 315-332

2 BAR-HILLEL, Y , PERLIS, M , AND SHAMIR, E. On formal properties of simple phrase structure grammars *Zeitschrift fur Phonetik Sprachwissenshaft und Kommunikationsforschung 14* (1961), 143-172

3 BIRD, M The equivalence problem for deterministic two-tape automata *J Comptr. and Syst. Sci. 7* (1973), 218-236

4 DAVIS, M *Computability and Unsolvability* McGraw-Hill, New York, 1958

5 ELGOT, C , AND MEZEL, J On relations defined by generalized finite automata *IBM J. Res and Develop 9* (1965), 47-68

6 ELGOT, C , AND RUTLEDGE, J RS-machines with almost blank tape *J ACM 11* (1964), 313-337.

7 FISCHER, P , MEYER, A , AND ROSENBERG, A Counter machines and counter languages *Math. Syst Theory 2* (1968), 265-283

8 FRIEDMAN, E The inclusion problem for simple languages *Theoretical Comptr Sci. 1* (1976), 297-316

9 GINSBURG, S , AND GREIBACH, S Deterministic context-free languages *Inform and Contr 9* (1966), 620-648

10 GINSBURG, S , AND SPANIER, E Bounded Algol-like languages *Trans. Amer. Math. Soc. 113* (1964), 333-368

11 GINSBURG, S , AND SPANIER, E Finite-turn pushdown automata *SIAM J. Contr. 4* (1966), 429-453

12 GREIBACH, S Remarks on the complexity of nondeterministic counter languages *Theoretical Comptr Sci 1* (1976), 269-288

13 HARRISON, M., AND IBARRA, O Multi-tape and multi-head pushdown automata *Inform and Contr 13* (1968), 433-470

14 HARTMANIS, J , AND HOFCROFT, J What makes some language theory problems undecidable. *J Comptr and Syst Sci 4* (1970), 368-376

15 HILBERT, D Mathematische probleme Vortrag Gehalten auf dem Int Math Kongress zu Paris 1900 *Nachr Akad. Wiss Gottingen Math -Phys* (1900), 253-297, translation in *Bull Am Math Soc 8* (1901-1902), 437-479

16 HOFCROFT, J , AND ULLMAN, J *Formal Languages and Their Relation to Automata* Addison-Wesley, Reading, Mass , 1969

17 IBARRA, O A note on semilinear sets and bounded-reversal multihead pushdown automata *Inform Proc Letters 3* (1974), 25-28

18 IBARRA, O , AND KIM, C A useful device for showing the solvability of some decision problems Proc Eighth ACM Symp on Theory of Comptng , 1976, pp 135-140, *J Comptr and Syst Sci 13* (1976), 153-160

19 KORENJAK, A , AND HOPCROFT, J Simple deterministic languages Conf Rec IEEE 7th Annual Symp. on Switching and Automata Theory, 1966, pp 36-46

20 MATIJASEVIČ, Y Enumerable sets are Diophantine *Soviet Math Dokl 11* (1970), 354-357.

21 MINSKY, M Recursive unsolvability of Post's problem of Tag and other topics in the theory of Turing machines *Annals of Math 74* (1961), 437-455

22 PARIKH, R On context-free languages *J ACM 13* (1966), 570-581

23 RABIN, M , AND SCOTT, D Finite automata and their decision problems *IBM J Res. and Develop 3* (1959), 114-125

24 ROSENBERG, A Nonwriting extensions of finite automata Ph D Th , Harvard U , Cambridge, Mass , 1965

25 SUDBOROUGH, I Bounded-reversal multi-head finite automata languages *Inform. and Contr. 25* (1974), 317-328

26 VALIANT, L The equivalence problem for deterministic finite-turn pushdown automata *Inform and Contr 25* (1974), 123-133

27 VALIANT, L , AND PATTERSON, M Deterministic one-counter automata *J Comptr and Syst Sci 10* (1975), 340-350

RECEIVED SEPTEMBER 1976, REVISED MARCH 1977