ON THE COMPUTATIONAL POWER OF REVERSAL-BOUNDED MACHINES[†]

by

Ronald V. Book
Department of Mathematics
and Computer Science Program
University of California
Santa Barbara, CA 93106, USA

and

Chee K. Yap
Department of Computer Science
Yale University
New Haven, CO 06520, USA

Introduction

A multitape Turing machine whose behavior is restricted so that each read-write head can change directions only a bounded number of times (independent of the input) is *reversal-bounded*. The number of reversals made by a Turing machine's read-write head during a computation has been studied as a dynamical measure of computational complexity [6,12,14,18] and restricted acceptors with a bound on the number of reversals have been used in automaton theory to characterize certain classes of languages [1,5,8,9,11]. In this paper we investigate classes of languages accepted by reversal-bounded machines and present two sets of results. The results show that when the reversal-bounded restriction is imposed, nondeterministic machines are more powerful than deterministic ones, and for deterministic time-bounded machines, the additional restriction to reversal-bounded operation greatly limits their computational power.

The power of nondeterminism has been a source of problems in automata theory for many years. In only a very few cases of restricted models of computation is it known whether the nondeterministic mode of operation is strictly more powerful than the deterministic mode, e.g., for finite-state acceptors nondeterminism adds no additional power while for pushdown store acceptors it does. It is known [1] that nondeterministic reversal-bounded acceptors are more powerful than deterministic reversal-bounded acceptors when no other restrictions are placed on computational resources. Here we show that this is also true when the machines are forced to operate in linear time or in real time, that is, there is languages accepted in real time by nondeterministic reversal-bounded acceptors that cannot be accepted in linear time by deterministic reversal-bounded acceptors. (Notice that the analagous statement about Turing machines with no restrictions on reversals has not been proven.) Further, for any time bound deterministic reversal-bounded computation is strictly weaker than nondeterministic reversal-bounded computation and the class P_{REVBD} of languages accepted in polynomial time by deterministic reversal-bounded machines is not equal to the class NP_{REVBD} of languages accepted in polynomial time by nondeterministic reversal-bounded machines. However, these facts become uninteresting when

[†] This research was supported in part by the National Science Foundation under Grant MCS-76-05744.

it is realized that if a deterministic acceptor is reversal-bounded, then it may as well be reversal-bounded and operate in linear time.

The restriction that machines be reversal-bounded is a severe one for deterministic machines. It is shown here that for machines operating in real time or in linear time there are languages accepted by machines with no restriction on reversals that cannot be accepted by any deterministic reversal-bounded machine. Further, any language over a one letter alphabet (tally language) accepted by a deterministic reversal-bounded machine must be in fact a regular set.

Preliminaries

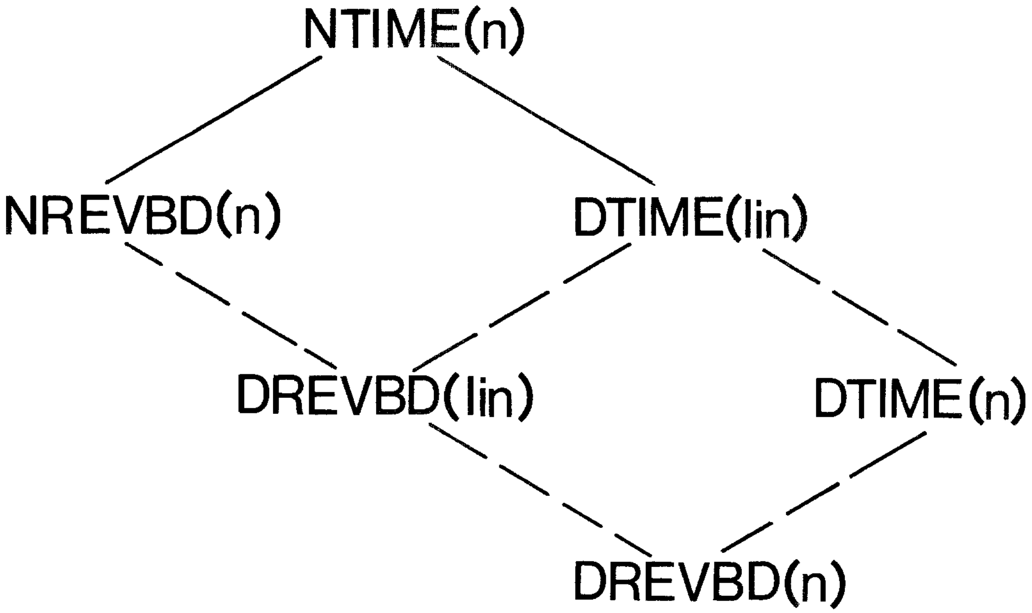
We consider multitape Turing acceptors having a one-way read-only input tape and some number of tapes with two-way read-write heads as auxiliary storage. The set of input strings accepted by a machine M is denoted by L(M). A reversal of a Turing machine tape is a step in a computation in which the read-write head changes directions, e.g., after a number of transitions in which the head moves only to the right, a transition occurs in which the head moves to the left. A Turing acceptor M is reversal-bounded if there exists a fixed constant k such that in every computation each read-write head makes at most k reversals. It is sufficient to consider the case of $k = 1$ [5] as long as one allows an increase in the number of storage tapes, and in this case there is no loss of running time. Further, it is easy to see that allowing two-way action on the input tape adds no computational power as long as the read head's behavior is also reversal-bounded.

Let DREVBD be the class of languages accepted by deterministic multitape Turing acceptors that are reversal-bounded. Let DREVBD(n)(DREVBD(lin)) be the class of languages accepted in real time (linear time) by deterministic multitape Turing acceptors that are reversal-bounded. Let NREVBD(n)(NREVBD(lin) be the class of languages accepted in real time (linear time) by nondeterministic multitape Turing machines that are reversal-bounded. The class NREVBD(n) has also been referred to as $\mathcal{L}_{BNP}$ [2,4].

In Reference [1] it was shown that DREVBD is a class of recursive sets and that a language is recursively enumerable if and only if it is accepted by a nondeterministic multitape Turing acceptor that is reversal-bounded. Hence there is a strong difference between the deterministic and nondeterministic modes of operation for reversal-bounded machines.

In Reference [5] it was shown that if a nondeterministic multitape Turing acceptor is reversal-bounded and operates in linear time, then there is another reversal-bounded acceptor that operates in real time and accepts the same language, i.e., NREVBD(lin) = NREVBD(n).

Let DTIME(n)(DTIME(lin)) be the class of languages accepted in real time by deterministic multitape Turing acceptors and let NTIME(n)(NTIME(lin)) be the class of languages accepted in real time (linear time) by nondeterministic multitape Turing



C_1 - - C_2 indicates $C_1 \subsetneq C_2$

C_1 — C_2 indicates $C_1 \subseteq C_2$

acceptors. For these classes there is no restriction on the number of reversals the machines may make. It is known that a language is accepted in linear time by a nondeterministic multitape Turing acceptor if and only if it is accepted in real time by such a machine i.e., NTIME(lin) = NTIME(n) [3].

The main results of this paper are summarized by the figure; that is, we show how the classes DREVBD(n) and DREVBD(lin) compare to each other and to DTIME(n), DTIME(lin), and NREVBD(n). We can conclude that in the case of deterministic machines, the restriction that all read-write heads be reversal-bounded is a very strong restriction. This constrasts with the results in References [2,4,5] on the class NREVBD(n) and other classes specified by nondeterministic machines that are both reversal-bounded and time-bounded.

Main Results

In this section we establish the relationships shown in the figure. We begin by showing that some of the obvious inclusions are in fact proper inclusions.

It is known [13,15] that there exists a deterministic context-free language L such that L can be accepted by a deterministic pushdown store acceptor that makes only one reversal on its pushdown store but L cannot be accepted in real time by any deterministic multitape Turing acceptor. Thus the language L can be recognized by a deterministic reversal-bounded machine that operates in linear time ($L \in$ DREVBD(lin)), but not by any machine that operates in real time ($L \notin$ DTIME(n) so $L \notin$ DREVBD(n)). Thus we see that DREVBD(n) $\underset{\neq}{\subset}$ DREVBD(lin), DREVBD(lin) $\not\subseteq$ DTIME(n), and DTIME(n) $\underset{\neq}{\subset}$ DTIME(lin).

Now we turn to a technical lemma that is very useful in showing how DREVBD relates to other classes.

Lemma 1: Let M_1 be a deterministic multitape Turing machine that is reversal-bounded. Let Σ be a finite alphabet and let d and # be two symbols not in Σ. Suppose that the language L accepted by M_1 is such that (i) $L \subseteq \Sigma^*\{d\}^*\{\#\}$ and (ii) for each string $w \in \Sigma^*$, if $wd^p\#$ and $wd^q\#$ are in L, then $p = q$. Further, suppose that in M_1's computation on any input string $wd^p\#$, either M_1 transfers into an accepting state when it first reads # or M_1 does not accept $wd^p\#$. Then one can construct a deterministic reversal-bounded machine M_2 such that the language accepted by M_2 is $\{w\# \mid$ there exists a p such that $wd^p\#$ is in $L\}$.

Informally, we interpret Lemma 1 as saying that DREVBD is closed under "removal of padding" (as made precise by the statement of Lemma 1). The classes DTIME(n), DTIME(lin), and NTIME(n) do not have this property since for every recursively enumerable set L there exist a finite alphabet Σ, symbols d and # not in Σ, and a partial recursive function f from Σ^* to the natural numbers such that $\{wd^{f(w)}\# \mid w \in L\}$ is in DTIME(n). Hence, if DTIME(n) were closed under "removal of padding," then every recursively enumerable set would be in DTIME(n). This leads to the next result.

Theorem 2: There are languages accepted in real time by deterministic multitape Turing machines that cannot be accepted by any deterministic reversal-bounded machine. That is, DTIME(n) $\not\subseteq$ DREVBD. Hence, DTIME(n) and DREVBD are incomparable.

Corollary:

(i) There are languages accepted in real time by deterministic multitape Turing machines that cannot be accepted in real time by any deterministic reversal-bounded machine, that is, DREVBD(n) $\subsetneq$ DTIME(n).

(ii) There are languages accepted in real time by deterministic multitape Turing machines that cannot be accepted in linear time by any deterministic reversal-bounded machine, that is, DTIME(n) $\not\subseteq$ DREVBD(lin). Hence, DTIME(n) and DREVBD(lin) are incomparable.

(iii) There are languages accepted in linear time by deterministic multitape Turing machines that cannot be accepted in linear time by any deterministic reversal-bounded machine. That is, DREVBD(lin) $\subsetneq$ DTIME(lin).

Recall that a language is recursively enumerable if and only if it is accepted by a nondeterministic reversal-bounded multitape acceptor [1]. Using the proof techniques of Reference [1], it can be shown that every recursively enumerable set can be represented, with suitable padding, as a language in NREVBD(n) and hence that NREVBD(n) is not closed under "removal of padding." This leads to the next result.

Theorem 3: There are languages accepted in real time by nondeterministic reversal-bounded machines that cannot be accepted by any deterministic reversal-bounded machine, that is, NREVBD(n) $\not\subseteq$ DREVBD.

Corollary: There are languages accepted in real time by nondterministic reversal-bounded machines that cannot be accepted in linear time by any deterministic reversal-bounded machine, that is, DREVBD(lin) $\subsetneq$ NREVBD(n).

Corollary: None of the classes DREVBD, DREVBD(lin), or DREVBD(n) is closed under nonerasing homomorphism.

If we interpret a language to be an encoding of a logical predicate, then we can interpret the second corollary to Theorem 3 as saying that the classes of predicates recognized by deterministic reversal-bounded machines (without time bound or or in linear time or in real time) are not closed under bounded existential quantification.

Recall that it has not been shown that DTIME(lin) $\neq$ NTIME(n) (although we believe that this inequality is true) and thus the first corollary to Theorem 3 provides a stronger result than what is known about time-bounded Turing machines with no bound on reversals.

The above results establish all the relationships between classes illustrated in the figure.

Properties of DREVBD

Now we present some results on languages accepted by deterministic reversal-bounded machines. These results indicate that the imposition of a finite bound on the number of reversals allowed is a severe restriction on the power of acceptance by deterministic machines.

Recall that a "tally" language is a language on a one letter alphabet. In Reference [19] it is shown that every tally language in DREVBD(n) is a regular set. The next result is stronger.

Theorem 4: Let a and # be two symbols. Let L be a language over $\{a\}$ with # as an endmarker, i.e., $L \subseteq \{a\}^*\{\#\}$. If L is accepted by a deterministic reversal-bounded acceptor, then L is a regular set, that is, the tally languages in DREVBD are only regular sets.

The proof of Theorem 4 depends on a more general result on encodings of integers. Theorem 4 is reminiscent of the result [16,17] that numbers of the form 2^n cannot be represented with just two counters (and no input tape) where initially one counter contains n and the other counter is empty.

The power of Theorem 4 becomes apparent when one notes that it can be used to show the following inequalities already established above: DREVBD(n) $\neq$ DTIME(n); DREVBD(lin) $\neq$ DTIME(lin); DREVBD(lin) $\neq$ NTIME(n); DREVBD $\neq$ DTIME(n); DREVBD $\neq$ DTIME(lin); DREVBD $\neq$ NTIME(n).

Recall that if M_1 is a multitape Turing machine that has more than one read-write head on some of its tapes, then there is another machine M_2 with only one read-write head per tape (but possibly more tapes) such that M_2 accepts the same language as M_1 and M_2 uses the same amount of time as M_1 [7,12]. This is true even if M_1 operates in real time [7]. Using Theorem 4 we can show that this result cannot be extended to the case where all read-write heads are reversal-bounded.

Corollary: There is a tally language (with endmarker) that can be accepted by a deterministic reversal-bounded machine if one of the work tapes has two reversal-bounded read-write heads but this language cannot be accepted by any deterministic reversal-bounded machine with only one head per tape.

The language $\{a^{n^2}\# \mid n \geq 1\}$ is an example of language satisfying the corollary.

It is known [1] that the class DREVBD is a class of recursive sets. However Dr. Richard Lipton has pointed out a stronger result.

Theorem 5: A language is accepted by a deterministic reversal-bounded machine if and only if it is accepted in linear time by such a machine, that is, DREVBD = DREVBD(lin).

Properties of NREVBD(n)

There are important differences between the deterministic and nondeterministic modes of operation under the restriction to finite reversal bounds. While the results on nondeterministic reversal-bounded machines in References [1,5] contrast strongly with the properties of DREVBD presented in this paper, distinctions between DREVBD(n) and NREVBD(n) can also be made.

First, note that the tally language $\{a^{n^2} \mid n \geq 1\}$ is a nonregular set that is in NREVBD(n) so that the statement analogous to Theorem 4 for NREVBD(n) is not true. In fact, the next result shows that the tally languages (with or without endmarker) in NREVBD(n) can encode a great deal of information.

Theorem 6: The following are equivalent:

(a) Every tally language in NREVBD(n) is accepted in polynomial time by a deterministic multitape Turing machine.

(b) Every language accepted in exponential time by a nondeterministic Turing machine is also accepted in exponential time by a deterministic Turing machine.

It has not been shown that NREVBD(n) $\neq$ NTIME(n) although it has been conjectured [4,5] that this inequality is true. Every context-free language is in NREVBD(n) if and only if the Dyck set on two letters is in NREVBD(n) if and only if NREVBD(n) = NTIME(n).

The exact relationship between DTIME(n) and NREVBD(n) or between DTIME(lin) and NREVBD(n) is not known. We believe that NREVBD(n) $\not\subseteq$ DTIME(lin) for otherwise the class P of languages accepted in polynomial time by deterministic machines would be equal to the class NP of languages accepted in polynomial time by nondeterministic machines. If DTIME(n) $\subseteq$ NREVBD(n) or DTIME(lin) $\subseteq$ NREVBD(n), then the Dyck set on two letters would be in NREVBD(n) so that NREVBD(n) = NTIME(n). We conjecture that the classes DTIME(lin) and NREVBD(n) are incomparable.

Note that every context-free language can be accepted by a nondeterministic reversal-bounded machine that runs in time n^2 and thus every language in NTIME(n) can be accepted by such a machine, so that $\mathrm{NREVBD}(n) \subseteq \mathrm{NTIME}(n) \subseteq \mathrm{NREVBD}(n^2)$.

Other Remarks

In Reference [9] two restrictions were placed on pushdown store acceptors. One restriction required the strings written on the pushdown store to be elements in a bounded language. The other restriction was on the number of reversals. Both of these restrictions can be placed on multitape Turing machines producing multicounter acceptors in the first case and reversal-bounded machines in the second.

Unrestricted multicounter machines accept all the recursively enumerable sets so that some resource must be restricted if smaller classes are to be obtained. Thus, time-bounded multicounter acceptors have been studied. Of interest here is the fact that the class P_{counter} of languages accepted in polynimial time by deterministic multicounter acceptors is not equal to the class NP_{counter} of languages accepted in polynomial time by nondeterministic multicounter acceptors [10]. This leads us to consider the analogous question for reversal-bounded acceptors: is the class P_{REVBD} of languages accepted in polynomial time by deterministic reversal-bounded machines equal to the class NP_{REVBD} of languages accepted in polynomial time by nondeterministic reversal-bounded machines?

We settle this question very simply. First, it is easy to see that $NP_{\mathrm{REVBD}} = NP$, that is, a language is accepted in polynomial time by a nondeterministic reversal-bounded machine if and only if it is accepted in polynomial time by a nondeterministic (unrestricted) multitape Turing machine. Second, recall that $\mathrm{NTIME} \subseteq NP$ and from above note that $\mathrm{DREVBD}(\mathrm{lin}) \subseteq P_{\mathrm{REVBD}} \subseteq \mathrm{DREVBD} = \mathrm{DREVBD}(\mathrm{lin}) \underset{\neq}{\subset} \mathrm{NREVBD}(\mathrm{lin}) \subseteq \mathrm{NTIME}(n)$. Thus, $P_{\mathrm{REVBD}} = \mathrm{DREVBD}(\mathrm{lin}) \underset{\neq}{\subset} NP = NP_{\mathrm{REVBD}}$.

Proposition: The class P_{REVBD} of languages accepted in polynomial time by deterministic reversal-bounded machines is not equal to the class NP_{REVBD} of languages accepted in polynomial time by nondeterministic reversal-bounded machines.

References

1. B. Baker and R. Book, Reversal-bounded multipushdown machines, JCSS 8 (1974), 315-322.

2. R. Book, Simple representations of certain classes of languages, submitted for publication.

3. R. Book and S. Greibach, Quasi-realtime languages, Math. Systems Theory 4 (1970), 97-111.

4. R. Book and M. Nivat, Linear languages and intersections of classes of languages, in preparation.

5. R. Book, M. Nivat, and M. Paterson, Reversal-bounded acceptors and intersections of linear languages, SIAM J. Computing 3 (1974), 283-295.

6. P. Fischer, The reduction of tape reversals for off-line one-tape Turing machines, JCSS 2 (1968), 136-147.

7. P. Fischer, A. Meyer, and A. Rosenberg, Real-time simulation of multihead tape units, JACM 19 (1972), 590-607.

8. S. Ginsburg and E. Spanier, Finite-turn pushdown machines, SIAM J. Control 4 (1966), 429-453.

9. S. Greibach, An infinite hierarchy of context-free languages, JACM 16 (1969), 91-106.

10. S. Greibach, Remarks on the complexity of nondeterministic counter languages, Theoretical Computer Science 1 (1976), 269-288.

11. S. Greibach, Visits, crosses and reversals for nondeterministic off-line machines, Info. Control, to appear.

12. J. Hartmanis, Tape-reversal bounded Turing machine computations, JCSS 2 (1968), 117-135.

13. J. Hartmanis and R. Stearns, On the computational complexity of algorithms, Trans. Amer. Math. Soc. 117 (1965), 285-306.

14. T. Kameda and R. Vollmar, Note on tape reversal complexity of languages, Info. Control 17 (1970), 203-215.

15. A. Rosenberg, Real-time definable languages, JACM 14 (1967), 645-662.

16. R. Schroeppel, A two counter machine cannot calculate 2^n, M.I.T. AI Memo. 257 (1973).

17. F. Yao, Computation by 2-counter machines, unpublished paper.

18. A. Barashka, Concerning the number of reversals of the stack head of a two-way pushdown automaton, Cybernetics 4 (1975), 564-570.

19. R. Book, On languages with a certain prefix property, Math. Systems Theory 10 (1977), to appear.