On Non-Determinancy in Simple Computing Devices

J. Hartmanis*

Received October 21, 1971

Summary. This paper studies one-tape Turing machines with $k$ read-only heads which are restricted to the original input. The main result shows that if any set accepted by such a 3-head non-deterministic Turing machine can be accepted by a deterministic Turing machine with more read-only heads, then the deterministic and non-deterministic context-sensitive languages are identical. Several related results are derived and some tantalizing open problems are discussed.

Introduction and Preliminaries

This paper deals with the problem of non-determinacy in some very simple computing devices and shows that if non-determinacy can be eliminated for these devices by using a "more complicated" device of the same type, then non-determinacy can be eliminated in a wide class of computing devices, including the linearly bounded automata. Thus these results give a very simple class of computing devices, for which we cannot yet answer the question about the power of non-determinism, and for which a deeper understanding of these problems may yield a solution of the classic and embarrassing problem about non-deterministic context-sensitive languages [1, 2].

The first class of computing devices considered consists of one-tape Turing machines with $k$ read-only heads, $k = 1, 2, 3, \ldots$, which are restricted to the original input. We assume that the input is placed between special end-markers and thus the $k$ heads can read the input but cannot go past the end markers. For sake of brevity we will refer to these devices as $k$-head automata. The input is accepted if, after starting the automaton in its starting state with all heads on the left endmarker, the automaton enters an accepting state and halts. The input is rejected if the automaton enters a rejecting state and halts or if it does not halt. Note that the automaton can sense when two heads are on the same tape square.

Let $C_D^k$ denote the set of all languages accepted by deterministic $k$-head automata and let $C_N^k$ denote the class of all languages accepted by non-deterministic $k$-head automata.

The main result of this paper shows that if every non-deterministic 3-head automaton can be replaced by a deterministic automaton with more heads, that is

$$C_N^3 \subseteq \bigcup_{k=1}^{\infty} C_D^k,$$

then the deterministic and non-deterministic context-sensitive languages are the same. In other words, if the non-determinacy for all 3-head automata can be

* This research has been supported in part by National Science Foundation Grant GJ-155.
eliminated by using more heads, then any non-deterministic linearly bounded automaton (l.b.a.) can be replaced by a deterministic l.b.a. The proof of this result is achieved by a three fold reduction of this problem and contains a novel encoding of many head positions on an input into one head position on an extended input.

Surprisingly, it looks that even the simple sounding problem of determining whether there exists a set accepted by a non-deterministic 3-head automaton which cannot be accepted by any deterministic k-head automaton, appears to be very difficult.

We conclude this part of the paper by deriving a result which shows that adding more heads to an automaton increases its computational power and discussing several open problems.

Finally, we indicate the general applicability of the proof techniques used and indicate how they can be applied to the many-tape—two-tape problem for Turing machines.

Reduction to Machines with Three Heads

In this section we prove that if every non-deterministic 3-head automaton can be replaced with a deterministic automaton with (possibly) more heads, then the deterministic and non-deterministic context-sensitive languages are the same.

The proof proceeds through several reductions. The first reduction is by means of a known result [3].

Lemma. If every set accepted by a non-deterministic \( L(n) = \log n \)-tape bounded Turing machine can be accepted by a deterministic \( L(n) = \log n \)-tape bounded Turing machine, then the deterministic and non-deterministic context-sensitive languages are the same.

Proof. For the sake of completeness we describe a \( L(n) = \log n \)-tape bounded Turing machine and outline the proof. A \( L(n) = \log n \)-tape bounded T.M. is a T.M. with two tapes: the input tape on which between end markers is placed the input and on which the T.M. can move a two-way, read-only head; the working tape of the T.M. is initially blank and on it the machine can move a two-way read-write head which cannot visit more than \( \lfloor \log n \rfloor \) tape squares for an input of length \( n \). The accepting and rejecting of inputs for this machine is defined in the same way as for a \( k \)-head automaton. For a more detailed discussion of \( \log n \)-tape bounded T.M.'s see [2, 3, 4].

Assume that any non-deterministic \( L(n) = \log n \)-tape bounded T.M. can be replaced by a deterministic \( L(n) = \log n \)-tape bounded T.M. Let \( A \subseteq \Sigma^* \) be a set accepted by a non-deterministic linearly bounded automaton \( M \). We will show that then \( A \) can also be accepted by a deterministic l.b.a. To do this let \( \# \) be a new symbol not in \( \Sigma \) and let

\[
T(A) = \{ w | w = u \#; u \in A \text{ and } |u| + t = 2^{\lfloor |u| \rfloor} \},
\]

where \( |u| \) designates the length of the sequence \( u \). Thus we are just attaching enough markers to each sequence \( u \) in \( A \) to increase its length exponentially. Observe now that \( T(A) \) can be accepted by a non-deterministic \( \log n \)-tape bounded Turing machine \( M_1 \). The machine \( M_1 \) operates as follows: for a given input \( wM_1 \)