REMARK ON THE COMPUTATIONAL POWER OF A TURING MACHINE VARIANT

Lionel E. DEIMEL, Jr.*
School of Information and Computer Science,
Georgia Institute of Technology, Atlanta, Georgia 30332, USA

Received 9 July 1974
Revised version received 7 October 1974

automata theory theory of computation

1. Introduction

In [1], Fischer describes minor variations on the Turing machine (TM) model of a computing device – the Post machine (PM), S-machine (SM), and D-machine (DM). Every action of a TM involves three operations – a state change, a print (overprint) operation, and a tape head movement. Each action of the variant machines, however, involves only two operations, one required and one selected from the remaining two. The PM changes state and either prints a symbol or moves its tape head; the SM prints and either changes state or moves its tape head; and the DM moves its tape head and either changes state or prints. A U-machine, which performs only one of the three operations at a time, is also defined. Fischer shows that given a particular TM, it is possible to find a PM, SM, DM, and UM which simulate it. Thus, one concludes that all five models are in some sense equivalent – a number of restrictions on Turing's original model has no essential effect on the power of the model. We here wish to show that a slight restriction on the DM distinguishes it from other models by its inability to simulate the other automata in a special way.

2. Machine definitions

We follow Fischer's practice in defining machines by use of transition tables.

Definition 1: A *TM* is a set of quintuples (transitions) of the form (q_i, S_j, S_k, D, q_l), where q_i is the present state of the machine; S_j is the symbol under scan; S_k is the symbol to be printed on the scanned square (replacing S_j); $D \in \{L, R, N\}$, where L and R indicate movement of the tape head one square left and right, respectively, and N denotes no movement of the tape head; and q_l represents the new internal state.

The machine halts if no transitions apply. The machine is deterministic, that is, no two quintuples begin with the same state-symbol pair. The sets of states and symbols are each finite. The machine is started in some initial state q_1 scanning an infinite tape containing the symbol S_0 (or blank) on all but a finite number of squares.

Definition 2: A *DM* is a set of quadruples of the form (q_i, S_j, X, D), where q_i, S_j, and D are as above and X is either a state (representing a state change) or a symbol (representing the printing of the symbol). Otherwise, the DM is defined as a TM.

By a *configuration* of a Turing-type automaton (TM, PM, etc.), we mean the total state of the machine as specified by its internal state, (finite) tape expression, and scanned square.

3. Simulation

Various definitions of simulation have been proposed by Fischer and others [1, 2]. Our concern will

* Graduate student.

be with a very restrictive definition. The definition is formalized in the appendix. We state informally a slightly more restrictive version of it here:

By simulation of one Turing-type automaton by another, we mean that

(1) Both machines start in the same configuration,
(2) Given both machines in the same configuration, the simulating machine achieves the next configuration of the simulated machine in a finite number of steps and passes through no configuration which is a possible configuration of the simulated machine in between, and
(3) The simulating machine halts if and only if the simulated machine halts.

The definition requires not only that the simulating machine be substitutable for the simulated machine (timing considerations aside), but also that simulation be step-by-step. Thus, by looking only at the simulating machine, we can monitor the exact course of computation of the simulated machine.

4. Some simulation results

Fischer proves the well-known result that the N option (no head movement) in the TM formulation is superfluous if we exclude TM's which may stay on one square and overprint without moving forever. (Such machines are of little interest, of course). He also shows that, given an N-free TM, one can find a PM, SM, and UM which simulate it. (Most of the constructions used conform to the simulation definition under consideration. Those which do not may be made to do so at the price of more involved constructions). Is the N option necessary in the definition of the other machines? In fact, it is not allowed in the definitions of the PM, SM, and UM, although it is in that of the DM. We wish to establish whether this is a necessary characteristic of the DM.

Definition 3: An N-free DM will be called a *restricted DM.*

Theorem 1: Every N-free TM modified so as to maintain a right end maker, say \$, on its tape can be simulated by a restricted DM.

Proof: Clearly, any TM may be modified in the manner described. Input tapes are to be as before except that the rightmost non-blank symbol, which must be to the right of the initial head position, is now \$. The modified machine moves the end marker right as necessary. (It need never move \$ to the left). The modified TM performs essentially the same computation as before.

The quintuples of the modified TM, (q_i, S_j, S_k, D, q_l), $D \neq N$, may be divided into three classes:

(1) $j = k$, i.e, no new symbol is written,
(2) $i = l, j \neq k$, i.e., a new symbol is written but the state is unchanged,
(3) $i \neq l, j \neq k$, i.e., a new symbol is written and the state is changed.

The first two cases are easily mirrored by the simulating machine:

(1) (q_i, S_j, q_l, D)
(2) (q_i, S_j, S_k, D)

In order to simulate transitions of form (3), we expand our alphabet to include primed and doubly primed symbols for each symbol S other than \$. New states are also introduced:

(3) (q_i, S_j, q_{ij}, R)
(q_{ij}, S, S', R), for all $S \neq \$$
$(q_{ij}, \$, q'_{ij}, L)$
(q'_{ij}, S', S'', L)
(q'_{ij}, S_j, S_k, R)
(q'_{ij}, S'', S'', R)
$(q'_{ij}, \$, q''_{ij}, L)$
(q''_{ij}, S'', S, L)
(q''_{ij}, S_k, q_l, D)

Since the DM cannot change state, print a new symbol, and move to a different square all at once, it is necessary to effect the state and symbol changes sequentially. The DM accomplishes these changes by first moving right to the end marker and returning to print the new symbol, then moving right to the end marker and properly altering its state and head position upon return to the new symbol. The restricted DM clearly simulates the modified TM in the desired manner.

Corollary 1.1: There is a universal restricted DM.

Proof: Let U be a universal TM modified so that a right end marker is maintained. If we confine the input tapes of U to those containing descriptions of TM's which always move their tape heads but which never change state and print a new symbol on the same transition (i.e., we confine the descriptions to those of restricted DM's), U will be universal over the

restricted DM's. It follows from theorem 1 and the existence of U that there exists a restricted DM universal over the class of restricted DM's.

Theorem 2: There exists an N-free TM which cannot be simulated (in the sense of our definition) by a restricted DM.

Proof: Consider the TM T defined as follows:

(q_1, S_0, S_1, q_2, R)

where q_1 is the initial state and S_0 is the blank. When started on a square containing anything but a blank, T halts. When started on a square containing a blank, it prints S_1, moves right in state q_2, and halts.

Consider the operation of T when presented with a blank tape – T prints an S_1 and halts in state q_2 to the right of S_1. A simulating restricted DM D must have a quadruple beginning with q_1, S_0. D must move right or left and either print or change state. Assume it changes state to q_1'. The configuration of D is exactly as before with q_1 renamed q_1' – it is still scanning a blank type. Clearly D may change states as many times as we wish, but at some point it must instead print a symbol. Assume at some point D prints symbol S_a. Since the tape head must move and the state cannot change, the action of D is governed by the same quadruple as before, as it is scanning another square containing a blank. This means D prints S_a and moves to another square containing S_0, ad infinitum. Thus, it is impossible for D to print a single symbol and halt when presented with a blank tape. No restricted DM can simulate T.

Theorem 2 makes clear the need for the end marker and complex construction of theorem 1.

The full implications of corollary 1.1 and theorem 2 are unclear. Obviously, the ability to perform all computable tasks and the ability to do them in a certain way are not the same thing. The representation of the tasks may be crucial. Note that the coding of an input tape with an end marker from one without such a marker is not a TM-computable operation – the machine cannot know when it has seen the last non-blank square on the tape in order to place the end marker. In fact, however, an end marker as such is not actually required, so long as some delimiter of known position is always on the tape. Placing such a delimiter is TM-computable but not, as should be clear from theorem 2, restricted-DM-computable.

One additional result may be noted. It is possible to simulate any N-free TM with a PM, SM, and UM which never explicitly represent in their transition table a null action (overprinting with the same symbol, "changing" to the same state, N head movement). The result may be extended to the DM only by requiring the use of an end marker or its equivalent, in which case theorem 1 may be easily modified to produce the desired simulation.

References

[1] P.C. Fischer, On formalisms for Turing machines, J. ACM 12 (1965) 570–580.

[2] G.T. Herman, Simulation of one abstract computing machine by another, C. ACM 11 (1968) 803, 813.

Appendix

Given machines M_1 and M_2 with state sets K_1 and K_2, respectively, and a bijective function $f: K_1 \to K_2' \subseteq K_2$ which maps the initial state of M_1 to the initial state of M_2. Define the function $g: K_2 \to K_1 \cup \{\emptyset\}$ such that $g(q) = f^{-1}(q)$ if $q \in K_2'$ and $g(q) = \emptyset$ otherwise.

Definition 4: Configurations of M_1 and M_2 are said to be *corresponding configurations* if they differ at most in state and if $f(q_1) = q_2$, where q_1 and q_2 are the states of M_1 and M_2, respectively. (Thus $g(q_2) = q_1$).

Definition 5: M_2 is said to simulate (or more properly, be capable of simulating) M_1 if and only if, when M_1 is started in any initial configuration and M_2 is started in the corresponding configuration:

(1) M_2 halts if and only if M_1 halts

(2) If M_1 achieves configuration C_{n+1} immediately following configuration C_n, then M_2, from the configuration corresponding to C_n, in a finite number of steps achieves the configuration corresponding to C_{n+1} without achieving any corresponding configuration of M_1 in between.