MAC TECHNICAL MEMORANDUM 37

REAL-TIME SIMULATION OF MULTIDOMENSIONAL TURING MACHINES BY STORAGE MODIFICATION MACHINES

A. Schönhage

December 1973

This research was supported by the National
Science Foundation under research grant
GJ-34671.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

CAMBRIDGE MASSACHUSETTS 02139

REAL-TIME SIMULATION OF MULTIDIMENSIONAL TURING MACHINES BY STORAGE MODIFICATION MACHINES

A. Schönhage[1]

1. Introduction.

In [1] the author introduced a new machine model, now called the Storage Modification Machine (SMM).[2] It was claimed, but not proved, that SMM's can simulate all sorts of Turing machines -- those with multidimensional worktapes in particular -- in real time.

In the following sections we describe a real-time simulation technique for keeping track of the movements of a Turing machine read-write head in the plane in such a way that repeated visits to a square are properly recognized. This construction shows how to overcome the main difficulty in real-time simulation by SMM's of multidimensional storage devices, a difficulty which typically arises in the two-dimensional case. Furthermore, this will reinforce the intuition that such multidimensional worktapes do not possess any latent computational power.

2. The rules for real-time simulation.

We consider Turing machines with one input tape, one output tape, and an initially blank storage plane with one head in start position (0,0). In a single step such a machine

a) reads an input symbol,

b) accesses and shifts once on its two-dimensional worktape,

c) prints an output symbol,

d) advances the input and output tapes one position.

[1] This work was done at the Massachusetts Institute of Technology in August, 1973. This is a preliminary, informal report meant only as a working paper. A more careful report by the author will follow.

[2] An SMM constructs its own storage from an unlimited pool of nodes and arcs of bounded variety. Access is by following arcs, and an arc from an accessed node may be redirected to any other accessed node or to any new node.

An SMM simulates such a Turing machine "in real time" if it has the same input-output behavior as the Turing machine and there is a constant bound (not dependent on the input or time) on the number of SMM steps per Turing machine step. In other words, we allow bounded speed-up but no look-ahead.

3. The storage reference structure.

We can think of the squares of the storage plane to be represented as nodes which are arranged in a rectangular grid and which are connected to their neighbors by horizontal arcs W, N, E, S (West, North,...), as shown in Figure 1.

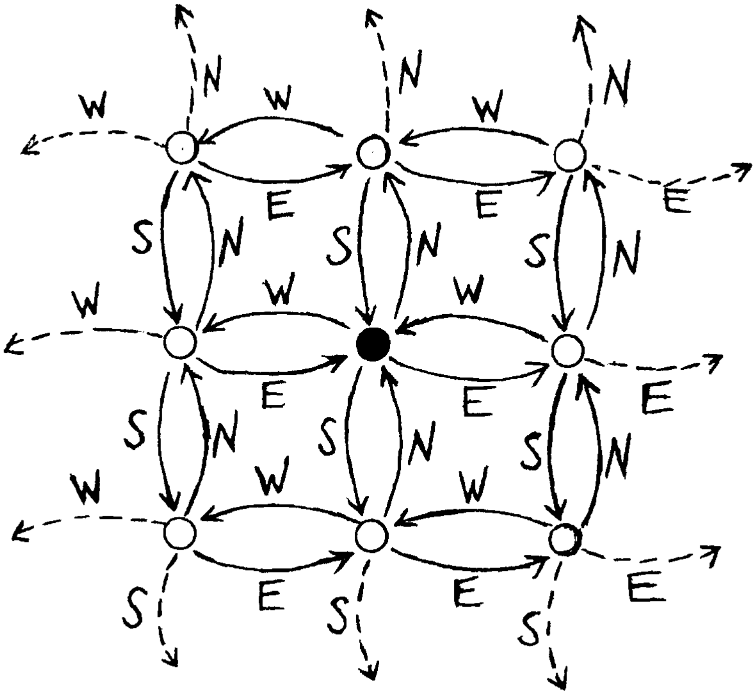

Fig. 1

It has to be understood, however, that this infinite structure does not preexist but has to be constructed in finite parts during the simulation. This requirement causes the main difficulty: how to decide in the case of a missing arc whether the node to which it should point has already been constructed.

In view of this question, we represent the plane by an infinite sequence of such grids, which we call floors, on successively reduced scales. The ground floor F^0 consists of the nodes M_z^0 ($z = (x,y) \in Z^2$) horizontally connected as shown in Figure 1. In the same way, for $k = 1,2,\ldots$, $F^k = \{M_z^k \mid z \in Z^2\}$.

From each M_z^k there is an upward arc U pointing to $M_{z'}^{k+1}$, where $z' = (x',y')$ is obtained from $z = (x,y)$ by $x' := \lfloor (x+1)/3 \rfloor$ $y' := \lfloor (y+1)/3 \rfloor$. Inversely, there are nine downward arcs $D_1,\ldots,D_9$ pointing from $M_{z'}^{k+1}$ to the different M_z^k 's whose U-arcs point to $M_{z'}^{k+1}$, according to the pattern shown in Fig. 2.

Fig.2

D_1	D_2	D_3
D_4	D_5	D_6
D_7	D_8	D_9

Thus, for example, D_8NW leads to the same node as D_4. In F^0 the D_i's can be used to encode the information contained in the squares of the storage plane.

Again we must stress that during the simulation only finite parts of the total structure described above are constructed. Arcs not yet properly connected will be pointing to their origin, thus indicating that something is missing.

Initially there will be only the node $M_{(0,0)}^0$ on the ground floor and no nodes on higher floors. The operations at each stage of the simulation will preserve the following: If the intended destination of any D_i- arc from an installed node M_z^k $(k \geq 1)$ has been installed, then all nine destinations have been installed along with the proper connections (upward, downward, and horizontal arcs) among all ten nodes. At each stage, furthermore, the only existing node without a properly installed U-arc will be the node $M_{(0,0)}^k$ on the top floor F^k reached so far.

The simulation process will consist of a sequence of stages. At stage t, floor k_t will be serviced, where the sequence $(k_1,k_2,k_3,\ldots)=(0,1,0,2,0,1,0,3,0,1,0,2,0,1,0,4,\ldots)$ is defined recursively by

$$k_t = \begin{cases} 0 & \text{for odd values of } t \\ k_{\frac{t}{2}}+1 & \text{for even values of } t \end{cases} \quad (t \geq 1)$$

A step of the Turing machine will be simulated each time the ground floor is serviced (i.e., in each odd stage). In the even stages the SMM will work on higher floors in order to prepare for the proper installation of the new arcs which are demanded dynamically by the movements of the storage head of the Turing machine. As servicing a floor will require only bounded time, the whole process will be a real-time simulation.

4. The stage counter.

In view of the real-time requirements, we provide immediate access to the floor to be serviced at each stage by the following construction (see Fig. 3 for illustration): At stage $t = 2\tau$ the "time pointer" T is directed to C_t, and the "half time pointer" H is directed to C_τ. (Initially both T and H point to C_0). The new node C_{t+1} has to be created, with "service pointer" P directed to floor-counting node K_0. Then after the ground floor is serviced, another new node C_{t+2} is created. Pointers H and T are reset to HU and TUU, respectively (indicated by the dashed lines in Fig. 3), and then TP is directed to HPU (to K_{j+1} in Fig. 3). (The first servicing of floor j, in stage 2^j, will have created K_{j+1}.) Finally, floor j+1 is serviced in stage $2\tau + 2$.

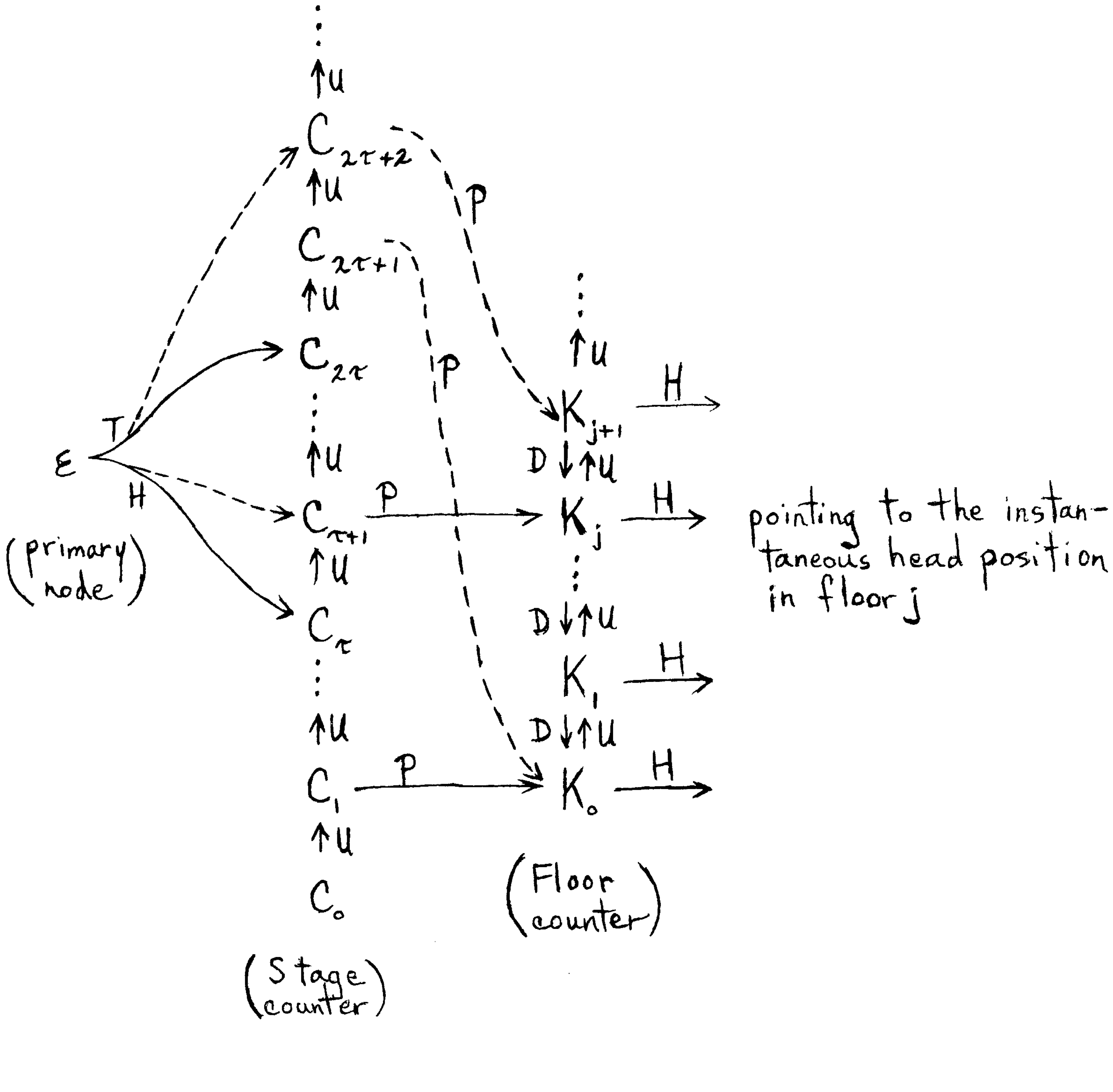
ε
T
H
(primary node)
C₂τ+2
C₂τ+1
C₂τ
Cτ+1
Cτ
C₁
C₀
U
P
(Stage counter)
Kj+1
Kj
K₁
K₀
D
U
H
pointing to the instantaneous head position in floor j
(Floor counter)

Fig. 3

5. The head movements.

The purpose of the H-pointers shown at the right-hand side of Fig. 3 is to simultaneously trace the movements of the Turing machine storage head on all floors already installed. Because of the reduced scales of the upper floors, however, larger delays in updating these pointers will be tolerable there.

6. Servicing floor j.

Servicing consists of *preparation* and *head adjustment*. On the ground floor preparation precedes head adjustment, but on higher floors it follows head adjustment. The preparation takes place around the instantaneous head position as recorded on floor j. Let us consider this node to be the enlarged one in the center of Fig. 1. Then preparation simply means to provide the other eight nodes shown in Fig. 1 and their mutual horizontal connections; i.e., it means to check whether they already are installed and, in the case that some of them are missing, to construct the missing elements. This can be achieved in a bounded number of steps provided that either

(*) { *all the nodes on the next floor* (floor j+1) *which can be reached by U-arcs from the nine nodes under preparation already have been installed along with all their horizontal connections*

or the enlarged node is the single node $M^{j}_{(0,0)}$ on the top floor F^{j}. (Note that the enlarged node need not happen to be in position D_5, Fig. 2). For, in the case that (*) holds, a single U-step leads to complete information on what parts of the neighborhood already exist.

In the exceptional case that floor j is the top floor so far and the enlarged node is the single node $M^{j}_{(0,0)}$ on that floor, the node $M^{j+1}_{(0,0)}$ and the the neighbors of $M^{j}_{(0,0)}$ are all constructed along with all their connecting arcs. To complete the introduction of floor j+1, node K_{j+1} is constructed along with the U-arc to it, the D-arc from it, and the H-arc from it to $M^{j+1}_{(0,0)}$.

Head adjustment on the ground floor takes place according to the head motion in the next step of the Turing machine computation. On other floors the head location TPH is updated to be the destination TPDHU of the U-arc from the head location on the floor below (floor j-1). (See Fig. 3.)

Obviously the purpose of preparation on the ground floor is to furnish that portion of the storage plane that may be needed for the simulation of the next move by the storage head of the Turing machine. The purpose of preparation on floor j+1, on the other hand, is to enable (by condition (*)) efficient preparation on floor j the next two times that floor is serviced. The reader should verify that condition (*) fails at no stage except for those at which the top floor is serviced. This can be shown inductively in conjunction with the additional assertion that each adjustment of a floor's H-pointer moves the pointer not more than a king's move in chess. The significant properties of the floor servicing sequence are these:

(i) The first service of floor j precedes the first service of floor j+1.

(ii) Floor j is serviced at most twice before floor j+1 is first serviced or between successive services of floor j+1.

7. Conclusion.

It should be obvious now how to generalize this method of real-time simulation to higher dimensions, to several heads in each of several planes, etc. Moreover, it is also possible to start with a finite amount of information already recorded in the storage plain (that is, to allow an off-line multidimensional input); then, of course, a corresponding rectangle of properly connected nodes on the ground floor should be provided as initial input to the SMM as well.

Reference: [1] A. Schönhage; Universelle Turing-Speicherung.

BIBLIOGRAPHIC DATA SHEET

1. Report No.
NSF-OCA-GJ34671 - TM-37

2.

3. Recipient's Accession No.

4. Title and Subtitle
Real-Time Simulation of Multidimensional Turing Machines
by Storage Modification Machines

5. Report Date : Issued
December 1973

6.

7. Author(s)
A. Schönhage

8. Performing Organization Rept. No. MAC TM-37

9. Performing Organization Name and Address
PROJECT MAC; MASSACHUSETTS INSTITUTE OF TECHNOLOGY:
545 Technology Square, Cambridge, Massachusetts 02139

10. Project/Task/Work Unit No.

11. Contract/Grant No.
GJ34671

12. Sponsoring Organization Name and Address
Associate Program Director
Office of Computing Activities
National Science Foundation
Washington, D. C. 20550

13. Type of Report & Period Covered: Interim
Scientific Report

14.

15. Supplementary Notes
This is a preliminary, informal report meant only as a working paper. A more careful report by the author will follow.

16. Abstracts

17. Key Words and Document Analysis. 17a. Descriptors

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement
Unlimited Distribution
Write Project MAC Publications

19. Security Class (This Report)
UNCLASSIFIED

20. Security Class (This Page
UNCLASSIFIED

21. No. of Pages
9

22. Price

THIS FORM MAY BE REPRODUCED

MIT/LCS/TM-37

REAL - TIME SIMULATION OF MULTIDIMENSIONAL TURING MACHINES BY STORAGE MODIFICATION MACHINES

A. Schonhage

December 1973