Simulations among Multidimensional Turing Machines

Michael Conrad Loui

August 1980

© Michael Conrad Loui 1980

The author hereby grants to M.I.T. permission to reproduce and
to distribute publicly copies of this thesis document in whole or in part.

This report was prepared with the support of the National Science Foundation Grant No. MCS-77-19754 and the Fannie and John Hertz Foundation.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LABORATORY FOR COMPUTER SCIENCE

CAMBRIDGE MASSACHUSETTS 02139

[illegible]

by

Michael Conrad Loui

Submitted to the Department of Electrical Engineering and Computer Science
on August 12, 1980
in partial fulfillment of the requirements for the degree of
[illegible]

Abstract

This thesis presents three independent papers: nearly optimal on-line simulations among multidimensional Turing machines, a space bound for one-tape multidimensional Turing machines, and new proofs in the pebble game.

For all $d \geq 1$, all $e > d$, and all $\epsilon > 0$, every deterministic multihead e-dimensional Turing machine of time complexity $T(n)$ can be simulated on-line by a deterministic multihead d-dimensional Turing machine in time $O(T(n)^{1 + 1/d - 1/e + \epsilon})$. This simulation almost achieves the known lower bound $\Omega(T(n)^{1 + 1/d - 1/e})$ on the time required.

Every nondeterministic d-dimensional Turing machine with one worktape head of time complexity $T(n)$ can be simulated by a deterministic Turing machine of space complexity $(T(n) \log T(n))^{d/(d+1)}$. The proof includes a generalization of counting sequences.

An overlap argument is used to derive two new [illegible]. Every directed acyclic graph G with n vertices and bounded indegree can be pebbled with $O(n/\log n)$ pebbles. Furthermore, if $S \geq O(n/\log n)$, then G can be pebbled with S pebbles in time $S2^{O(n/S)}$.

Key Words: automata, computational complexity, [illegible], Turing machine, nondeterminism, time complexity, tape complexity, pebble game, time-space tradeoff, overlap.

THESIS SUPERVISOR: Albert R. Meyer
TITLE: Professor of Computer Science and Engineering

ACKNOWLEDGMENTS

For the last three years Albert Meyer has supervised my doctoral studies with patience and wisdom. Under his tutelage I have honed my research skills. I have benefited immeasurably from his advice on lecturing and technical writing. His exemplary fastidiousness bespeaks his high standards of excellence.

Harold Abelson and Leonard Adleman, the readers on my thesis committee, suffered through early drafts of this thesis. I appreciate their perspicacious recommendations. In addition, Professor Adleman suggested the generalization of crossing sequences used in Chapter 2.

I value the encouragement of Nicholas Pippenger and Joel Seiferas, who have taken particular interest in my work and have generously shared their time with me. Dr. Pippenger noticed the extension of the proof of Theorem 3.1 to the proof of Theorem 3.2. Moreover, Professor Seiferas and Robert Melville provided a copy of a paper by Grigoriev [4] that is related to Chapter 1.

My colleagues in the Theory of Computation Group have contributed to an intellectually exciting environment. Neil Immerman and Andrea LaPaugh suggested expository improvements to parts of this thesis. Daniel Weise devised the pebbling strategy in the proof of Lemma 3.1, and Christos Papadimitriou provided the proof of Lemma 4.1. I also thank Alan Baratz, Errol Lloyd, Jeffrey Jaffe, Gary Miller, Ron Pinter, Vaughan Pratt, Ronald Rivest, Adi Shamir, Alan Sherman, Michael Sipser, and Karl Winklman for enlightening discussions.

I am fortunate that Ronald Book, James Munkres, and Alan Willsky guided my early professional development. I hope to emulate the clarity of their writing and the effectiveness of their teaching.

My family and friends have been kind and encouraging throughout my graduate study.

I am grateful for the fellowship provided by the Fannie and John Hertz Foundation for my five years at M.I.T. My research has also been supported by the National Science Foundation under Grant No. MCS-77-19754.

[illegible]

For the last three years Albert Meyer has supervised my doctoral studies with patience and wisdom. Under his tutelage I have honed my research skills. I have benefited immeasurably from his advice on lecturing and technical writing. His exemplary fastidiousness bespeaks his high standards of excellence.

Harold Abelson and Leonard Adleman, the readers on my thesis committee, [illegible] early drafts of this thesis. I appreciate their perspicacious recommendations. [illegible] Professor Adleman suggested the [illegible]

I value the encouragement of Nicholas Pippenger and Joel Seiferas, who have been particular interest in my work and have generously shared their time with me. Dr. Pippenger [illegible] extension of the proof of Theorem 3.1 to the proof of Theorem 3.2. [illegible] and Robert Melville provided a copy of a paper by Grigoriev [4] that is related to Chapter [illegible]

My colleagues in the Theory of Computation Group have [illegible] exciting environment. Neil Immerman and [illegible] parts of this thesis. Daniel Weise devised the pebbling strategy in the proof of [illegible] Charles Papadimitriou provided the proof [illegible] Jeffrey Jaffe, Gary Miller, Ron Pinter, Vaughan Pratt, Ronald Rivest, [illegible] Michael Sipser, and Karl Wiedman for enlightening discussions.

I am fortunate that Ronald Book, James Munkres, and Alan Willsky guided my early professional development. I hope to emulate the clarity of their writing and the effectiveness of their teaching.

My family and friends have been kind and encouraging throughout my graduate study.

I am grateful for the fellowship provided by the Fannie and John Hertz Foundation for my five years at M.I.T. My research has also been supported by the National Science Foundation under Grant No. MCS-77-19754.

PREFACE

The four chapters of this thesis were written independently and may be read seapartely. Each has its own introduction, terminology, and notations, but all references have been collected at the end of the thesis.

Chapter 1 presents an on-line simulation of a deterministic multihead e-dimensional Turing machine of time complexity $T(n)$ by a deterministic multihead d-dimensional machine of time complexity $O(T(n)^{1 + 1/d - 1/e + \varepsilon})$ for all $\varepsilon > 0$. In Theorem 1.2 the ε in the exponent is replaced by $o(1)$. This simulation nearly achieves the known lower bound $\Omega(T(n)^{1 + 1/d - 1/e})$ on the time required.

Continuing the study of multidimensional machines, Chapter 2 presents an off-line simulation of a nondeterministic d-dimensional machine with one worktape head that runs in time $T(n)$ by a deterministic machine in space $(T(n) \log T(n))^{d/(d+1)}$. An anonymous referee noticed the simulation by an alternating Turing machine in time $O((T(n) \log T(n))^{d/(d+1)})$ (Theorem 2.3). This chapter has been accepted for publication in *Theoretical Computer Science*. An earlier version appeared as Technical Memorandum TM-145 of the M.I.T. Laboratory for Computer Science [14].

Chapter 3 uses an overlap argument to derive new proofs in the pebble game. We develop a strategy that uses $O(n/\log n)$ pebbles to pebble every directed acyclic graph with n vertices and bounded indegree. A variation of this strategy uses S pebbles to pebble the graph in at most $2^{2^{O(n/S)}}$ steps. This note on the pebble game will appear in *Information Processing Letters*.

Chapter 4 recommends further research on automata with nonsequential storage structures. It includes a novel geometric argument that suggests a time-space tradeoff for simulating a multidimensional Turing machine by a tree machine.

Chapter 1. Simulations among Multidimensional Turing Machines

1.1. Background

Introduced by Hartmanis and Stearns [5], multidimensional Turing machines are natural generalizations of conventional Turing machines. Hennie and Grigoriev [3, 6] established a lower bound of $\Omega(T(n)^{1 + 1/d - 1/e})$ on the time required by a multihead d-dimensional Turing machine to simulate an e-dimensional machine of time complexity $T(n)$ on-line. We present a simulation that nearly achieves this bound.

Theorem 1.1. For all $d \geq 1$, all $e > d$, and all $\varepsilon > 0$, every multihead e-dimensional Turing machine of time complexity $T(n)$ can be simulated on-line by a multihead d-dimensional Turing machine in time $O(T(n)^{1 + 1/d - 1/e + \varepsilon})$.

For the case $d = 1$, Pippenger and Fischer [22] devised an optimal simulation that runs in time $O(T(n)^{2 - 1/e})$ on-line. Grigoriev [3] described an on-line simulation in time $O(T(n)^{1 + 1/d})$ when $e = d + 1$; even in this special case, Theorem 1.1 provides a better upper bound. Also, Grigoriev proved that every storage modification machine of time complexity $T(n)$ can be simulated on-line by a d-dimensional machine in time $O(T(n)^{1 + 1/(d+1)})$; since every multidimensional Turing machine can be simulated in real time by a storage modification machine [27], every e-dimensional machine can be simulated on-line by a d-dimensional machine in time $O(T(n)^{1 + 1/(d+1)})$. The time required by our simulation is smaller, however.

Grigoriev [4] demonstrated that every nondeterministic e-dimensional machine can be simulated off-line by a nondeterministic d-dimensional machine in time $O(T(n)^{1 + 1/d - 1/e + \varepsilon})$ for every $\varepsilon > 0$. We consider only deterministic machines. Our simulation can be modified to yield this result about nondeterministic machines.

Paul, Seiferas, and Simon [19] studied simulations among multidimensional machines with limited numbers of worktape heads. They established nonlinear lower bounds on the time required to simulate multidimensional machines on-line by machines with fewer worktape heads. Furthermore, they presented simulations of multitape multidimensional machines by machines with just two worktapes having one head each. In contrast, we present a simulation by a machine with more worktape heads.

1.2. Simulation

Let us review definitions for multidimensional Turing machines. To each cell of a d-dimensional worktape assign in the usual way a d-tuple of integers called the *coordinates* of the cell. The coordinates of adjacent cells differ in just one component by ± 1. The *origin* is the cell whose coordinates are all zero. A *d-dimensional Turing machine* has a finite-state control, a read-only input tape, a write-only output tape, and a finite number of d-dimensional worktapes, each of which has a finite number of heads. At each step the machine reads the symbols in the cells on which the input and worktape heads are positioned, writes symbols on these worktape cells and possibly on the output tape too, and shifts each worktape head in one of $2d + 1$ possible directions – either to one of $2d$ adjacent cells or to the same cell. Initially, all worktape cells hold blanks, and every worktape head is positioned on the origin of its tape. Leong and Seiferas [12] proved that every d-dimensional Turing machine can be simulated in real time by a d-dimensional machine having just one head on each of its worktapes.

Fix integers $d \geq 1$ and $e > d$, a positive real number ϵ, and a finite alphabet Δ. To establish Theorem 1.1, it suffices to exhibit an on-line simulation of a particular e-dimensional machine **E** with worktape alphabet Δ by a d-dimensional machine **D** in time $O(n^{1 + 1/d - 1/e + \epsilon})$. Machine **E** has one head on one worktape and operates in real time as follows. At each step it reads another input symbol, called a *command*, that has the form $\langle b, \delta \rangle$, where $b \in \Delta$, and δ is one of the $2e + 1$ directions in which the worktape head can shift. Suppose **E** is in a configuration in which the cell y scanned by the worktape head contains b'. When **E** then reads the input symbol $\langle b, \delta \rangle$, it writes b on y, writes b' on its output tape, and shifts the worktape head in direction δ. Call symbols of Δ *responses*. Let Σ be the set of commands for **E**. Machine **E** defines a function from Σ^* to Δ^* that maps a string of commands into a string of responses of the same length. Machine **D** *simulates* **E** *on-line* in time $T(n)$ if it computes this function in time $T(n)$ on inputs of length n, and for every input, for every j, machine **D** produces the jth response before reading the $(j + 1)$st command.

On a given input string of commands, when **E** has processed just the first τ commands, we say that **E** is *at time* τ. When **D** has processed only the first τ commands (and produced the first τ output responses), **D** is *at simulated time* τ.

Consider a string of n commands. For simplicity we describe a simulation in which n is available off-line. The simulation can be converted routinely to an on-line simulation with time loss of only a

constant factor [19]. (For $n' = 1, 2, 4, 8, \ldots$, machine D repeats the simulation *ab initio* using n' for the length of the input until n' is at least as large as n.)

On a d-dimensional worktape, a *box* is a set of cells that form a d-dimensional cube. The *volume* of a box is the number of cells in it. The *base cell* of a box is the cell whose coordinates are the smallest. Cell y is *at distance* s from cell z if the shortest rectilinear path from y to z has length s; equivalently, a worktape head requires exactly s steps to move from y to z. Cell y is *well within* box C of side s if it is at distance strictly greater than $s/3$ from every cell outside C. The *relative position* of a cell y with respect to a box C is the list of coordinates of y when the base cell of C is taken as the origin; if y is at distance s from the base cell of C, then its relative position can be specified by a binary string of length proportional to $\log s$. Write $x(h, C)$ for the cell at relative position h with respect to box C. The *relative position* of a box in C is the relative position of its base cell with respect to C.

Set

$$c_0 = 8^d 5^e,$$

$$L = de\,(\log n)/\log c_0,$$

$$c = c_0^{1/de},$$

$$s_L = n,$$

$$s_{L-1} = n^{1/e}.$$

We may assume that c is an integer and that ε is sufficiently small so that

$$d\varepsilon \leq 1/d - 1/e + \varepsilon \tag{1.1}$$

and

$$c^{\varepsilon/d-1} > 5^e. \tag{1.2}$$

Note that $c^L = n^{1/e}$ and $c_0^{L/d} = n^e$. For each $i < L - 1$, set