CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Scholarship@Claremont

Claremont Colleges
Scholarship @ Claremont

All HMC Faculty Publications and Research | HMC Faculty Scholarship

1-1-1982

Probabilistic Simulations

Nicholas J. Pippenger
Harvey Mudd College

Recommended Citation
Pippenger, Nicholas. "Probabilistic Simulations." ACM Symp. on Theory of Computing, 14 (1982), 17-26.

This Conference Proceeding is brought to you for free and open access by the HMC Faculty Scholarship at Scholarship @ Claremont. It has been accepted for inclusion in All HMC Faculty Publications and Research by an authorized administrator of Scholarship @ Claremont. For more information, please contact scholarship@cuc.claremont.edu.

PROBABILISTIC SIMULATIONS

(Preliminary Version)

Nicholas Pippenger
IBM Research Laboratory
San Jose, CA 95193

1. Introduction

The results of this paper concern the question of how fast machines with one type of storage media can simulate machines with a different type of storage media. Most work on this question has focused on the question of how fast one deterministic machine can simulate another. In this paper we shall look at the question of how fast a probabilistic machine can simulate another. This approach should be of interest in its own right, in view of the great attention that probabilistic algorithms have recently attracted. It has, however, two additional claims to interest. Firstly, a result concerning a probabilistic question can lead to an improved result concerning a traditional deterministic question. Specifically, we shall give an improved simulation of deterministic time-bounded multidimensional machines by deterministic space-bounded machines; the proof is probabilistic although the final result is not. Secondly, the use of probabilistic methods opens the way to allied disciplines and allows their power to be brought to bear on our problems. Specifically, we shall use game-theoretic and information-theoretic ideas (which are in turn based on probability theory).

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1982 ACM 0-89791-067-2/82/005/0017 $00.75

In this paper, all machines will have a one-way read-only input tape and a one-way write-only output tape. By "simulation", we shall mean on-line simulation. In addition to their input and output tapes, machines may have one or more storage media (which may be multidimensional or tree-structured), each with one or more access heads. By an "ℓ-dimensional machine" or a "tree machine", we shall mean a machine whose storage media are all ℓ-dimensional or tree-structured, respectively. More specifically, an ℓ-dimensional storage medium will have cells corresponding to points in $\{0, 1, \ldots, \}^{\ell}$ and 3^{ℓ} shifts. The distance (minimum number of shifts needed to travel between) two cells $a = \text{saveUB.}|1|, \ldots, a_{\ell})$ and $a' = (a'_1, \ldots, a'_{\ell})$ is given by the metric

$$d(a, a') = \max_{1 \le j \le \ell} |a_j - a'_j|.$$

A tree-structured storage medium will have cells corresponding to points in $\{0, 1\}^{*}$ and 3 shifts. The distance between two cells a and a' is given by the metric

$$d(a, a') = \|a\| + \|a'\| - 2\|\ell cp(a, a')\|,$$

where $\|a\|$ denotes the length of a and $\ell cp(a, a')$

denotes the longest common prefix of a and a'.

By a "probabilistic machine", we shall mean one that may flip coins but that always gives correct outputs. By the "running time" of such a machine, we shall mean the maximum (over all inputs) of the average (over coin flips) of the number of steps. (Babai [1] has suggested the term "Las Vegas" for probabilistic algorithms that always give correct outputs, as distinguished from "Monte Carlo" algorithms, which may give incorrect outputs.)

There is an alternate way of defining probabilistic machines and their running times that is often convenient. A "probabilistic machine" is one that may flip coins, always gives either correct outputs (success) or an initial segment of the correct outputs (failure), and succeeds with probability at least 1/2. The "running time" of such a machine is the maximum (over inputs and coin flips) of the number of steps. The equivalence (to within constant factors) of these definitions can be shown by routine methods.

We shall present probabilistic simulations for the situation of a probabilistic machine (called the "host") simulating a deterministic machine (called the "guest"), but these simulations have immediate corollaries in which the guests may also be probabilistic. We shall present simulations for the situation in which the guest has a single access head on a single storage medium, but these simulations have immediate corollaries in which the guest may have any number of access heads on any number of storage media. Finally, we shall assume that the number of steps taken by the guest in known in advance to the host. This assumption can be eliminated by routine methods (see Galil [4]).

2. An Upper Bound for Tree Machines

Our first result concerns the simulation of multidimensional machines by tree machines. A multidimensional machine running in time T can obviously be simulated by a deterministic tree machine running in time $O(T \log T)$. Reischuk [14] improved this to $T \exp O(\log^* T)$.

Theorem 1: A multidimensional machine running in time T can be simulated by a probabilistic tree machine running in time $O(T)$.

Proof: Let the guest run for $T = 2^\tau$ steps. Let $y = (y_1, \ldots, y_\ell)$ be a uniformly distributed random point in $\{0, \ldots, T-1\}^\ell$ (obtained from $\ell\tau$ independent unbiased coin flips). The position of the head of the guest can be regarded as a point $a = (a_1, \ldots, a_\ell)$ in $\{0, \ldots, T-1\}^\ell$. Let $b = a + y$ (that is for $1 \le j \le \ell$, let $b_j = a_j + y_j$ modulo T). For $1 \le j \le \ell$, let $b_{j,1}$ (most significant), ..., $b_{j,\tau}$ (least significant) in $\{0, 1\}$ be the binary digits of b_j. Define the map $f_y : \{0, \ldots, T-1\}^\ell \to \{0, 1\}^{\ell\tau}$ by

$$f_y(a) = b_{1,1} \cdots b_{\ell,1} \cdots b_{1,\tau} \cdots b_{\ell,\tau}.$$

Let the symbol stored in cell a of the guest be stored at cell $f_y(a)$ in the host. Let g denote the metric of the guest and let h denote the metric of the host. At each step the guest shifts from a cell a to a cell a' satisfying

$$g(a, a') \le 1.$$

A simple calculation shows that

$$\text{ave}_y\ h(f_y(a),\ f_y(a')) \leq 2\Sigma_{0\leq j<\ell\tau}(j+1)2^{-\lfloor j/\ell \rfloor}.$$

This sum is O(1), independently of T. Thus the host, in average time O(1), can shift from $f_y(a)$ to $f_y(a')$ when the guest shifts from a to a'. This allows the host, in average time O(T), to simulate T steps by the guest.□

It is natural to ask if one could use a deterministic storage mapping function f instead of the random storage mapping function f_y in this proof. De Millo, Eisenstat and Lipton [3] have shown that one cannot: for any function f: $\{0, \ldots, T-1\}^2 \rightarrow \{0, 1\}^*$, there exist points a and a' such that

$$g(a, a') \leq 1$$

but

$$h(f(a), f(a')) = \Omega(\log T).$$

The idea of using a random storage mapping function f_y instead of a deterministic storage mapping function f is due to Carter and Wegman [2], who introduced it in the context of hashing functions, which map a large random-access storage medium into a smaller one. We have adapted their idea to the context of multidimensional and tree-structured storage media, exhibiting an appropriate random storage mapping function and formalizing the result in terms of simulations.

It is sometimes of interest to regard randomization as a resource: to count the number of coin flips used by a probabilistic machine. In this simulation, the number is particularly small; with care, T steps by the guest can be simulated with O(log T) coin flips by the host.

Theorem 1 has consequences for the problem of simulating a time-bounded machine by a space-bounded machine. (In the remainder of this section, all machines are deterministic and all simulations are off-line.) Hopcroft, Paul and Valiant [7] showed that a one-dimensional machine running in time T can be simulated by a machine running in space O(T/log T). (Space is a sufficiently robust complexity measure that it is unnecessary to specify the storage media of space-bounded machines.) Paul and Reischuk [12] showed that a tree machine can be simulated in space O(T/log T) and that a multidimensional machine can be simulated in space O(T log log T/log T). By combining Reischuk's simulation of a multidimensional machine by a tree machine (cited above) with Paul and Reischuk's simulation of a tree machine by a space-bounded machine, a multidimensional machine can be simulated in space T (exp $O(\log^* T)$)/log T. The next result shows that it can be simulated in space O(T/log T).

Corollary 1.1: A deterministic multidimensional machine running in time T can be simulated off-line by a deterministic machine running in space O(T/log T).

This corollary is obtained by combining Theorem 1 with Paul and Reischuk's simulation of a tree machine in space O(T/log T) (cited above), and observing that the space-bounded machine can exhaustively search for a storage mapping function that does at least as well as the expectation (a sequence of O(log T) coin flips can certainly be represented in space O(T/log T).

3. Range Reduction for Multidimensional Machines

This section describes a result that will be needed in the following section. By the range of a computation we shall mean the maximum distance moved by any head away from its original position at any step of the computation. A machine running in time T always runs in range T, but for some types of machine it is possible to substantially reduce this range bound without substantially increasing the time bound. Paul and Reischuk [12] showed that a tree machine running in time T can be simulated by a tree machine running in time O(T) and in range O(log T). It would be of interest (as will be seen in the next section) to obtain the analogous result for multidimensional machines: that a ℓ-dimensional machine running in time T can be simulated by a ℓ-dimensional machine running in time O(T) and in range $O(T^{1/\ell})$. The closest approximation to this which has thusfar been obtained is time T exp $O((\log T)^{1/2})$ and range $T^{1/\ell}$ exp $O((\log T)^{1/2})$, which can be obtained as a corollary to a result of Loui [9]. For probabilistic simulations we can improve these bounds significantly.

Theorem 2: An ℓ-dimensional machine running in time T can be simulated by a probabilistic ℓ-dimensional machine running in time $O(T(\log T)^{1/\ell})$ and in range $O((T \log T)^{1/\ell})$.

The proof of this theorem will be obtained by combining three simulations that involve a new type of machine, which will be called a mulilayer machine. A multilayer machine is a machine having one or more multilayer storage media (which may be tree-structured or multidimensional). Each cell of a multilayered storage medium is capable of holding an unlimited number of symbols, one on each of an unlimited number of layers. The layer to be read or written is selected in a direct-access fashion by writing the index of the desired layer on a special one-dimensional layer selection tape.

Since direct access to layers is much more powerful than local access to cells, multilayer machines are interesting only when access to layers is restricted in some way, as measured by one or more of three new resources that will be introduced here for this purpose. By the change of a computation we shall mean the number of times that a new layer is selected. By the breadth of a computation we shall mean the number of different layers written upon during the computation. Finally, by the height of a computation we shall mean the maximum number of layers written upon in any one cell during the computation.

Proposition 2.1: An ℓ-dimensional machine running in time T can be simulated by a probabilistic multilayer machine running in time O(T), range $O(T^{1/\ell})$ and change $O(T^{1-1/\ell})$.

Proof: Let the guest run for $T = 2^{\ell\rho}$ steps. Let $R = 2^{\rho}$. Let y be a uniformly distributed random point in $\{0, \ldots, R-1\}^{\ell}$. The position of the head of the guest can be regarded as a point a in $\{0, \ldots, T-1\}^{\ell}$. Define the maps e_y: $\{0, \ldots, T-1\}^{\ell} \to \{0, \ldots, T/R\}^{\ell}$ and f_y: $\{0, \ldots, T-1\}^{\ell} \to \{0, \ldots, R-1\}^{\ell}$ by

$$a + y = e_y(a)R + f_y(a).$$

Let the symbol stored at cell a of the guest be stored at cell $f_y(a)$ of layer $e_y(a)$ of the host. It is easy to check that T steps by the guest can be simulated by the host in average time O(T), range R = $O(T^{1/\ell})$ and average change $O(T/R) = O(T^{1-1/\ell})$.□

Proposition 2.2: An ℓ-dimensional multilayer machine running in time T, range $R = O(T^{1/\ell})$ and change $C = O(T^{1-1/\ell})$ can be simulated by a probabilistic ℓ-dimensional multilayer machine running in time O(T), range $O(T^{1/\ell})$, height O(log T/log log T) and breadth $O((\log T)^{\ell-1})$.

Sketch of Proof: We shall begin with a simulation that meets the time, range and height bounds. We shall then indicate how to modify this simulation to also meet the breadth bound.

For each layer e of the guest that is written upon, let y_e be an independent uniformly distributed random point in $\{0, \ldots, R-1\}^\ell$. Let the symbol stored at cell a (in $\{0, \ldots, R-1\}^\ell$) of layer e of the guest be stored at cell $a + y_e$ (in $\{0, \ldots, 2R-1\}^\ell$) of layer e in the host.

The value of y_e for each layer e that is written upon can be kept in a directory (on a single additional layer) comprising $C = O(T^{1-1/\ell})$ records of length c = O(log T). If the directory uses universal hashing [2], it will fit in volume O(Cc) = O(T) and thus in range $O(T^{1/\ell})$; it can be accessed once in expected time $O(T^{1/\ell})$, and thus it can be accessed C times in expected time O(T). With probability at least 7/8, the time spent accessing this directory will be O(T).

Consider the height of the resulting computation. Let $p_{a,e}$ denote the probability that cell a of layer e of the host is nonblank. It is easy to see that

$$\Sigma_e\, p_{a,e} \le T/R^\ell = O(1)$$

for each cell a. It follows that the probability that cell a has H nonblank layers is $O(1)^H/H!$. Thus by choosing

$$H = O(\log T/\log\log T)$$

we can ensure that with probability at least 7/8, each of the O(T) cells of the host has at most H nonblank layers, so that the height bound is met.

To modify the simulation so that the breadth is also small, partition the cells of the host into ℓ-cubes of side $L = \lfloor \log T \rfloor$, using a grid whose origin is a uniformly distributed random point in $\{0, \ldots, L-1\}^\ell$. Let $q_{b,e}$ denote the probability that some cell in cube b of layer e of the host is nonblank. It is not hard to see that

$$\Sigma_e\, q_{b,e} \le (CL^\ell + T\ell L^{\ell-1})/R^\ell$$

$$= O(L^{\ell-1})$$

for each cube b. It follows that the probability that cube b has B nonblank layers is $O(L^{\ell-1})^B/B!$. Thus by choosing

$$B = O(L^{\ell-1})$$

we can ensure that with probability at least 7/8, none of the O(T) cubes have more than B nonblank layers.

Within each cube, the layers of the host can be reassigned so that at most B layers of the host are nonblank (so that the breadth bound is met). The host will change layers whenever it shifts from one cube to another. It is easy to see that this will happen an average of $O(T/L) = O(T/\log T)$ times.

For each cube, the reassignment of layers can be kept in a directory (on a single additional layer) comprising $B = O((\log T)^{\ell-1})$ records of length $b = O(\log T)$. If the directories use universal hashing [2], they will each fit in volume $O(Bb) = O((\log T)^{\ell})$ and thus in range $O(\log T)$; they can be accessed once in expected time $O(\log T)$, and thus they can be accessed $O(T/\log T)$ times in expected time $O(T)$. With probability at least 7/8, the time spent accessing these directories will be $O(T)$.

It is easy to check that with probability at least $1 - 1/8 - 1/8 - 1/8 - 1/8 = 1/2$, the host runs in time $O(T)$, range $O(T^{1/\ell})$, height $O(\log T/\log \log T)$ and breadth $O((\log T)^{\ell-1})$.□

Proposition 2.3: An ℓ-dimensional multilayer machine running in time T, range R, height H and breadth B can be simulated by a probabilistic ℓ-dimensional machine running in time $O(T(H \log B)^{1/\ell})$ and range $O(R(H \log B)^{1/\ell})$.

Proof: For each cell of the guest, the symbol in each layer can be kept in a directory comprising H records of length $O(\log B)$. If these directories use universal hashing [2], each directory will fit in volume $O(H \log B)$, and thus in range $O((H \log B)^{1/\ell})$; these directories will thus fit in range $O(R(H \log B)^{1/\ell})$. Each directory can be accessed once in expected time $O((H \log B)^{1/\ell})$, and thus these directories can be accessed T times in expected time $O(T(H \log B)^{1/\ell})$.□

These three simulations can be combined by routine methods to yield a simulation fulfilling Theorem 2.

4. An Upper Bound for Multidimensional Machines

The results of this section concern the the simulation of ℓ-dimensional machines by k-dimensional machines, where $k < \ell$. Hennie [6] showed that a deterministic one-dimensional machine requires time $\Omega(T^{2-1/\ell})$ to simulate an ℓ-dimensional machine running in time T. Pippenger and Fischer [13] showed that an ℓ-dimensional machine can be simulated by a deterministic one-dimensional machine in time $O(T^{2-1/\ell})$. Grigor'ev [5] observed that Hennie's argument yields the result that a deterministic k-dimensional machine requires time $\Omega(T^{1+1/k-1/\ell})$ to simulate an ℓ-dimensional machine running in time T. Loui [9] showed that an ℓ-dimensional machine can be simulated by a deterministic k-dimensional machine in time $O(T^{1+1/k-1/\ell}(\log T)^{m})$, where m depends on k and ℓ and $m \to \infty$ as $k \to \infty$ or $\ell \to \infty$. We shall obtain a significantly faster probabilistic simulation.

Theorem 3: An ℓ-dimensional machine running in time T and in range R can be simulated by a probabilistic k-dimensional machine running in time $O(TR^{\ell/k-1})$.

Proof: Let the guest run for $T = 2^\tau$ steps in range $R = 2^{k\rho}$. Let $y = (y_1, \ldots, y_\ell)$ be a uniformly distributed random point in $\{0, \ldots, R-1\}^\ell$ (obtained from $k\ell\rho$ independent unbiased coin flips). The position of the head of the guest can be regarded as a point $a = (a_1, \ldots, a_\ell)$ in $\{0, \ldots, R-1\}^\ell$. Let $b = a + y$ in $\{0, \ldots, R-1\}^\ell$ (that is, for $1 \le j \le \ell$, let $b_j = a_j + y_j$ modulo R). For $1 \le j \le \ell$, let $b_{j,1}$ (most significant), ..., $b_{j,k\rho}$ (least significant) in {0, 1}be the binary digits of b_j. For $1 \le i \le k$ and $1 \le j \le \ell\rho$, define $c_{i,j}$ in {0, 1} by the identity

$$c_{1,1}\cdots c_{k,1} \cdots c_{1,\ell\rho}\cdots c_{k,\ell\rho} = b_{1,1}\cdots b_{\ell,1} \cdots b_{1,k\rho}\cdots b_{\ell,k\rho}$$

in $\{0, 1\}^{k\ell\rho}$. Let $Q=2^{\ell\rho}$. For $1 \le i \le k$, let c_i in {0, ..., Q-1} be the number with binary digits $c_{i,1}$ (most significant), ..., $c_{i,\ell\rho}$ (least significant). Define the map f_y: $\{0, \ldots, R-1\}^\ell \to \{0, \ldots, Q-1\}^k$ by

$$f_y(a) = (c_1, \ldots, c_k).$$

Let the symbol stored in cell a of the guest be stored in cell $f_y(a)$ of the host. Let g be the metric of the guest and let h be the metric of the host. At each step, the guest shifts from a cell a to a cell a' satisfying

$$g(a, a') \le 1.$$

A simple calculation shows that

$$\text{ave}_y\, h(f_y(a), f_y(a')) \le \Sigma_{0 \le j \le k\ell\rho}\, 2^{\lfloor j/k \rfloor - \lfloor j/\ell \rfloor}$$

This sum is $O(R^{\ell/k-1})$, independently of T. Thus the host, in average time $O(R^{\ell/k-1})$, can shift from $f_y(a)$ to $f_y(a')$ when the guest shifts from a to a'. This allows the host, in average time $O(TR^{\ell/k-1})$, to simulate T steps by the guest.□

If in Theorem 3 we use the trivial bound $R \le T$, we obtain only the poor result that an ℓ-dimensional machine running in time T can be simulated by a probabilistic k-dimensional machine running in time $O(T^{\ell/k})$. If, however, we first apply Theorem 2, we obtain the following result.

Corollary 3.1: An ℓ-dimensional machine running in time T can be simulated by a probabilistic k-dimensional machine running in time $O(T^{1+1/k-1/\ell}(\log T)^{1/k})$.

This simulation improves Loui's in two respects. Firstly, it is faster: the factor $(\log T)^m$ in Loui's result exceeds $(\log T)^{k^3}$ when $\ell=k+1$. Secondly, it is simpler: it makes no use of recursion, and the processes of range reduction and dimension reduction are separated, whereas they are intertwined in Loui's simulation, since a certain amount of each must be accomplished at each level of the recursion. Loui's simulation, of course, has the merit of being deterministic.

5. A Lower Bound for Multidimensional Machines

The purpose of this secion is to extend Hennie's [6] and Grigor'ev's [5] lower bounds from deterministic hosts to probabilistic ones.

Theorem 4: A probabilistic k-dimensional machine requires time $\Omega(T^{1+1/k-1/\ell})$ to simulate an ℓ-dimensional machine running in time T.

For the proof of this theorem, we shall need a proposition concerning random variables. If C and B are random variables, we shall let $C \to B$ (read "C determines B") denote the event that C assumes a value with which only one value of B is compatible.

Proposition 4: If C is a random variable assuming c distinct values, and if $B_1, \ldots, B_N$ are mutually independent uniformly distributed random variables assuming $b_1, \ldots, b_N$ distinct values respectively, then

$$\Sigma_{1 \le n \le N} P(C \to B_n) \log b_n \le \log c.$$

Sketch of Proof: The proof, which is information-theoretic in nature, is based on the following inequalities. Firstly, if B_n is a uniformly distributed random variable assuming b_n distinct values, then

$$P(C \to B_n) \log b_n \le I(C; B_n),$$

where $I(C; B_n)$ denotes the mutual information between C and B_n. Secondly, if $B_1, \ldots, B_N$ are mutually independent random variables, then

$$\Sigma_{1 \le n \le N} I(C; B_n) \le I(C; B),$$

where $B = (B_1, \ldots, B_N)$. Thirdly,

$$I(C; B) \le H(C),$$

where H(C) denotes the entropy of C. Finally, if C assumes c distinct values, then

$$H(C) \le \log c,$$

which completes the proof.□

Proof of Theorem 4: Let the guest G be an ℓ-dimensional machine with a single access head on a single ℓ-dimensional storage medium. Let each input symbol read by G command it to write a 0 or 1 in the cell currently scanned by the head, write as an output the symbol currently scanned by the head, or shift the access head in one of the 3^ℓ possible ways.

Let the host H be a probabilistic k-dimensional machine that simulates G. Let $U_{x,y}$ be the number of steps taken by H when the input is the finite string x and the coin flips are as specified by the appropriate initial segment of the infinite binary string y. We shall show that

$$\max_x \operatorname{ave}_{y,q} = \Omega(T^{1+1/k-1/\ell}),$$

where the maximum is over all input strings x of length T and the average is over all infinite binary strings y (with the usual uniform probability distribution q).

Let p be an arbitrary probability distribution on the input strings of length T. Then

$$\begin{aligned} \max_x \operatorname{ave}_{y,q} U_{x,y} &\ge \operatorname{ave}_{x,p} \operatorname{ave}_{y,q} U_{x,y} \\ &= \operatorname{ave}_{y,q} \operatorname{ave}_{x,p} U_{x,y} \\ &\ge \inf_y \operatorname{ave}_{x,p} U_{x,y}. \end{aligned}$$

(This inequality has a simple game-theoretic interpretation: in a two-person zero-sum game, if one player must announce a probability distribution on his moves, after which the other player must announce his move, it cannot be a disadvantage to be the second player.) Thus it will suffice to exhibit a probability distribution p on the input strings of length T such that

$$\inf_y \operatorname{ave}_{x,p} U_{x,y} = \Omega(T^{1+1/k-1/\ell}).$$

A random input string x of length T is chosen according to the probability distribution p as follows. First, choose $N = 2^{\ell\rho}$ independent uniformly distributed random variables $Y_1, \ldots, Y_N$ in {0, 1}.

Second, choose $M = 2^{(\ell-1)\rho}$ independent uniformly distributed random variables $X_1, \ldots, X_M$ in $\{1, \ldots, N\}$. The string x will be the concatenation of a "storage phase" x_0 and M "retrieval phases" $x_1, \ldots, x_M$. Let $R = 2^\rho$. The storage phase, of length (2N-1) + (R-1), causes the values of $Y_1, \ldots, Y_N$ to be written in the cells $\{0, \ldots, R-1\}^\ell$ and returns the head to the origin. For $1 \le m \le M$, the m-th retrieval phase, of length (R-1) + 1 + (R-1), causes Y_{X_m} to be written as an output and returns the head to the origin. The length of x is thus $T = (2N+R-2) + M(2R-1) = \Theta(N)$.

For $1 \le m \le M$, let the random variable W_m denote the number of steps taken by H between reading the first symbol of x_m and writing the m-th output. We shall show that

$$E(\Sigma_{1 \le m \le M} W_m) = \Omega(N^{1+1/k-1/\ell}).$$

Since $M = \Omega(N^{1-1/\ell})$, it will suffice to show that

$$E(W_m) = \Omega(N^{1/k}).$$

We have

$$E(W_m) = \Sigma_{w \ge 0} P(W_m > w)$$

$$= \Sigma_{w \ge 0} [1 - P(W_m \le w)],$$

so it will suffice to show that

$$P(W_m \le w) = O(w^k).$$

We also have

$$P(W_m \le w) = \Sigma_{1 \le n \le N} P(W_m \le w | X_m = n)\ P(X_m = n)$$

$$= \Sigma_{1 \le n \le N} P(W_m \le w | X_m = n)/N,$$

so it will suffice to show that

$$\Sigma_{1 \le n \le N} P(W_m \le w | X_m = n) = O(w^k).$$

For $1 \le m \le M$, let the random variable Z_m denote the configuration of H just before reading the first symbol of x_m and, for $w \ge 0$, let $Z_{m,w}$ denote that portion of Z_m accessible within w steps.

If $X_m = n$, the event $W_m \le w$ implies the event $Z_{m,w} \to Y_n$. Thus it will suffice to show that

$$\Sigma_{1 \le n \le N} P(Z_{m,w} \to Y_n | X_m = n) = O(w^k).$$

The event $Z_{m,w} \to Y_n$ depends only upon $Y_1, \ldots, Y_N$ and $X_1, \ldots, X_{m-1}$, and the event $X_m = n$ is independent of these random variables. Thus $P(Z_{m,w} \to Y_n | X_m = n) = P(Z_{m,w} \to Y_n)$, and it will suffice to show that

$$\Sigma_{1 \le n \le N} P(Z_{m,w} \to Y_n) = O(w^k).$$

Since H is a k-dimensional machine, $Z_{m,w}$ assumes exp $O(w^k)$ distinct values. Thus Propostion 4 completes the proof.□

The information-theoretic argument used in the proof of Proposition 4 is related to arguments used by Paul [11] and others in the context of deterministic machines. The principal difference is that we use Shannon's information measure [15] rather than Kolmogorov's [8].

The game-theoretic argument used in the proof of Theorem 4 is related to arguments used by Yao [16] in the context of decision trees. The inequality we use is the easier and more general half of von Neumann's minimax theorem [10].

6. References

[1] L. Babai, "Monte Carlo Algorithms in Graph Isomorphism Testing", preprint.

[2] J. L. Carter and M. N. Wegman, "Universal Classes of Hash Functions", J. Comp. and Sys. Sci., 18 (1979) 143-154.

[3] R. A. DeMillo, S. C. Eisenstat and R. J. Lipton, "Preserving Average Proximity in Arrays", Comm. ACM, 21 (1978) 228-231.

[4] Z. Galil, "Two Fast Simulations which Imply Some Fast String Matching and Palindrome Recognition Algorithms", Info. Proc. Let., 4 (1976) 85-87.

[5] D. Yu. Grigor'ev, "Imbedding Theorems for Turing Machines of Different Dimensions and Kolmogorov Algorithms", Sov. Math. Dokl., 18 (1977) 588-592.

[6] F. C. Hennie, "On-Line Turing Machine Computations", IEEE Trans. on Comp., 15 (1966) 35-44.

[7] J. E. Hopcroft, W. J. Paul and L. G. Valiant, "On Time versus Space", J. ACM, 24 (1977) 332-337.

[8] A. N. Kolmogorov, "Three Approaches to the Quantitative Definintion of Information", Prob. of Info. Trans., 1 (1965) 1-7.

[9] M. C. Loui, "Simulations among Multidimensional Turing Machines", IEEE Symp on Found. of Comp. Sci., 22 (1981) 58-67.

[10] J. von Neumann, "Zur Theorie der Gesellschaftsspiele", Math. Ann., 100 (1928) 295-320.

[11] W. J. Paul, "Kolmogorov Complexity and Lower Bounds", Found. Comp. Theory, 2 (1979) 325-334.

[12] W. J. Paul and R. Reischuk, "On Time versus Space, II", J. Comp. and Sys. Sci., 22 (1981) 312-327.

[13] N. Pippenger and M. J. Fischer, "Relations among Complexity Measures", J. ACM, 26 (1979) 361-381.

[14] R. Reischuk, "A Fast Implementation of a Multidimensional Storage into a Tree Storage", Automata, Lang. and Prog., 7 (1980) 531-542.

[15] C. E. Shannon, "A Mathematical Theory of Communication", Bell Sys Tech. J., 27 (1948) 379-423, 623-656.

[16] A. C. Yao, "Probabilistic Computations--Toward a Unified Measure of Complexity", IEEE Symp. on Found. of Comp. Sci., 18 (1977) 222-227.