A Principle of Least Computational Action
(preliminary version)

AKHILESH TYAGI
Department of Computer Science
University of North Carolina
Chapel Hill, NC 27599-3175

Abstract

The units of energy-time product in the VLSI model of computation correspond to action. Motivated by the principle of least action, we propose a definition for computational action and develop a principle of least computational action. Any VLSI algorithm satisfying the principle of least computational action is also energy-time and AT^2 optimal.

1 Introduction

A VLSI system can be viewed as an information *engine* that processes input data in accordance with a predetermined algorithm so that the output is in the desired form. In doing so, it uses certain amount of space, time and energy. These resources (space, time and energy) to certain extent are interchangeable. For instance, the seminal work of Thompson [Tho79], Brent and Kung [BK80] demonstrates an area-time trade-off. Energy can also be traded for time as shown in Aggarwal *et al.* [ACR88]] and Tyagi [Tya89]. Consider the 2-dimensional space formed by area and energy. A VLSI computation corresponds to a path in the time projection of area, energy space. A given VLSI algorithm can be realized along a large variety of paths with in this area, energy, time space. Which of these paths result in energy-time (ET) and area-time (AT^2)* optimal computations? Note that ET optimality is based on the global use of energy and time. Is it possible to state some local optimality criteria that ensure global optimality? This paper is an attempt to answer these questions. The work reported here is still in ongoing stage and hence it is only a preliminary report.

The cost of resources needed to realize a function in VLSI can be bounded from below by several complexity measures, primary one being the communication or information complexity of f denoted by $I(f,n)$. The information complexity $I(f,n)$ is the minimum of number of bits that need to be exchanged between two processors each holding roughly half the input bits, to compute f cooperatively, over all equal sized partitions of input bits. All the known AT^2 and ET bounds are stated in terms of $I(f,n)$. One key technique, introduced by Thompson [Tho79], is to cut the VLSI chip so that the set of input bits is equally partitioned on two sides. The heart of the argument is to relate the length of this cut to a function of area and time as well as to $I(f,n)$. The reason for outlining this lower bound technique is to draw the reader's attention to the fact that the central role of wires in a VLSI system is to transmit information between different parts of the system. The total cost of the system, whether it is area, time or energy, seems to be dominated by the cost of wires. Hence the design of a VLSI system consists of dealing with these area, energy and time trade-offs. The principle of least computational action developed in this paper is another way of describing the optimal part of area, energy and time space.

Let us first introduce the principle of least action followed by an intuitive description of computational action. When an object moves from point A to point B in space, which of the many possible paths will it take? The laws of motion in Newtonian mechanics provide an answer to this question. There is an alternative way of capturing the characteristics of this motion. In Physics, *action* is defined as the path integral of *kinetic energy* minus *potential energy*. The principle of least action states that the object will take a path that minimizes action. It can be shown that in the absence of any frictional forces, the principle of least action encapsulates the laws of Newtonian mechanics. A similar notion of action can also be defined for quantum mechanics. An interesting question is whether some VLSI computation attribute can serve the role of computational action in characterizing optimal VLSI computations? In Section 3 we propose a communication switching energy based candidate for computational action. Section 4 closes the paper with some conclusions and open questions.

2 Background and Model

VLSI Model

The model of VLSI computation is essentially the same as the one described by Thompson [Tho79]. A computation is abstracted as a communication graph. A communication graph is very much like a flow graph with the primitives being some basic operators that are realizable as electrical devices. Two communicating nodes are adjacent in this graph. A layout can be viewed as a convex embedding of the communication graph in a Cartesian grid. Each grid point can either have a processor or a wire passing through. A wire cannot go through a grid

*Note that ET optimality implies AT^2 optimality since E has an upper bound of AT.

0-8186-3420-0/93 $03.00

point with a processor unless it is a terminal of the processor at that grid point. The number of layers is limited to some constant γ. Thus both the fanin and fanout are bounded by 4γ. Wires have unit width and bandwidth and processors have unit area. The initial data values are localized to some constant area, to preclude an encoding of the results. The input words are read at the designated nodes called input ports. The input is synchronous and each input bit is available only once. The input and output conventions are where-determinate (the locations of input/output ports are pre-designated) but need not be when-determinate (the times when input/output data become valid can be determined by the input value).

The following is an enhancement of this model to account for energy, taken from Tyagi [Tya88] and Kissin [Kis82]. We assume that whenever a wire of length l changes state, it consumes $\Theta(l)$ switching energy. This has the following justification. Let u be the unit switching energy that is required to switch a wire of unit length. Almost all of this energy is dissipated in the resistance of the transistor switching the wire. Let us analyze the amount of switching energy required to change the state of a wire with length l, width w and capacitance C. This transition involves transfer of charge equal to CV through a potential difference of V. The energy required to do this is $CV^2/2$. The capacitance C of the wire equals $\frac{\epsilon A}{d}$, where ϵ is the permittivity of the dioxide, $A = wl$ is the area of the wire, and d is the depth of the dioxide. Then the switching energy is $CV^2/2 = \frac{V^2 \epsilon w}{2d} l$. In a typical layout design, the widths of all the wires are within a constant factor of the minimum metal width for a process. For a given process, d and ϵ are constant. The supply voltage V, currently is in the range 3-5 Volts and is not expected to be lowered significantly in order for a chip to remain noise-immune. Thus dioxide depth d, wire width w, permittivity ϵ and the supply voltage V can all be absorbed into a proportionality constant permitting us to conclude that *the energy needed to switch a wire of length l is $\Theta(l)$.*

For the lower bound arguments, we do not account for the switching energy consumed by the processors. In the upper bound case, the computations performed by a processing node are assumed to be primitive, *i.e.* these gates have constant fanin (4γ). Their switching energy is at most a small constant (4γ) times the output wire switching energy. Hence the wire switching energy also constitutes an upper bound within a constant factor. Note that this assumes that the circuit is well-designed to the extent that no race conditions exist, which can increase the energy consumption exponentially over a well-designed circuit.

For lack of space we will only give an informal definition for switching energy. We distinguish between the combinational and sequential cases, which determines the upper bound on the number of times a wire can switch for each input instance. When a wire can switch at most once, we refer to it as *uniswitch model* (USM), which corresponds to combinational circuit. When there is no such upper bound on the amount of switching per wire, we are dealing with *multiswitch model* (MSM).

Let the wire switching energy $E_W(C, s, \vec{x})$ be the switching energy consumed by the wires when the input $\vec{x}$ is applied to a circuit C with the set of wires W in state s. Then $E_W(C, s, \vec{x})$ is given by $u \sum_{w_j \in W} k_j \times length(w_j)$, where u is the unit switching energy and wire w_j switches k_j times. Note that all the following definitions apply to both USM and MSM.

Definition 1 *The worst case energy consumption for a circuit C, $E_w(C)$, is defined to be $max_{s,\vec{x}} E_W(C, s, \vec{x})$, where the maximum is taken over all (state, input vector) pairs.*

The average case energy consumption is defined in a similar way. It is the average of switching energy consumed for all possible initial state, next input vector combinations.

Definition 2 *The average case energy consumption for a circuit C is defined to be its energy consumption averaged over all initial states and all input vectors. Thus $E_a(C) = \sum_{s,\vec{x}} E_W(C, s, \vec{x}) / (|S_i| \, |I|)$, where S_i is the set of initial states and I is the set of input vectors.*

Background

We describe some relevant concepts in this section.

Information Complexity:

We first introduce the notion of *information complexity* of a function $f(x_n x_{n-1} \ldots x_1) = y_m y_{m-1} \ldots y_1$. Let $C_f = (V, W)$ be a circuit to compute f. Consider a partition $\pi = \left\{ \pi^L = \{x_{i_1}, x_{i_2}, \ldots x_{i_{\lceil n/2 \rceil}}\}, \right.$ $\left. \pi^R = \{x_{j_1}, x_{j_2}, \ldots x_{j_{\lfloor n/2 \rfloor}}\} \right\}$ that divides the set of input bits into two equal-sized sets. The chip C_f can possibly be partitioned in such a way that one partition contains the input ports corresponding to π^L and the other one contains the input ports for π^R. Let $I(f, C_f, \pi, \vec{x}, n)$ be the number of bits that need to be exchanged between π^L and π^R when the input bits are assigned values according to $\vec{x} \in \{0,1\}^n$. Note that $I(f, C_f, \pi, \vec{x}, n)$ is ∞ for all the partitions π that cannot be realized in the chip C_f. The information complexity of f, $I(f, n)$, is the minimum number of bits exchanged between any two almost equal-sized partitions of the input bits over all implementations, which is $\min_{C_f} \min_{\pi \in \Pi_n} \max_{\vec{x}} I(f, C_f, \pi, \vec{x}, n)$. Π_n is the set of all approximately equal sized partitions of n bits such that each set in the partition contains at least n/c bits for a constant $c > 1$. We will use I to denote $I(f, n)$ in the following, whenever the function f is implicit. Many techniques were developed to derive lower bounds on $I(f, n)$ for specific functions [[AA80], [AUY83], [BK80], [JK84], [LS81], [MS82], [PS84], [Tho79], [Yao79]]. A particularly interesting class of functions: *transitive functions*, was introduced by Vuillemin [Vui83]. A transitive function embeds a transitive permutation group computation. Some examples of transitive functions include shifting, integer multiplication and linear transforms. Vuillemin also showed that $I(f, n)$ for a transitive function is $\Omega(n)$.

Spatial Entropy:

Spatial entropy is a concept introduced by Mead [[MC80], pages 366-370]. He argues that logical entropy quantized by the log of number of decisions needed for a computation (equivalent to depth of a circuit) is only part of the story. A physical system to implement the computation

also has to deal with *spatial entropy*, entropy of data being in the wrong place. It costs physical resources in area, energy and time to move data around before logical gates can operate to reduce the logical entropy of the computing system. To quote Mead:

> *" In any physical system, the logical entropy treated by classical complexity theory is only part of the story. There is also a spatial entropy associated with a computation. Spatial entropy may be thought of as a measure of data being in the wrong place, just as logical entropy is a measure of data being in the wrong form. Data communications are used to remove spatial entropy, just as logical operations are used to remove logical entropy."*

The logic gates remove the logical entropy, while the wires in the circuit remove the spatial entropy. We quantized the notion of spatial entropy (see Rajgopal [Raj92] for more details) as follows. Recall that **entropy** of a probability distribution $\{p_i\}$ s.t. $\sum_{i=0}^{n-1} p_i = 1$, $H(P) = \sum_{i=0}^{n-1} p_i \log\left(\frac{1}{p_i}\right)$, measures the information (# of bits) in the event. If p_0^w and p_1^w are the probabilities (frequency) of a wire w in a combinational circuit being in the state 0 and 1 respectively then the information transmitted on w is $H(w) = p_0^w \log\left(\frac{1}{p_0^w}\right) + p_1^w \log\left(\frac{1}{p_1^w}\right)$. Let the length of wire w be $l(w)$. A quantitative measure of spatial entropy of a circuit C containing wires W then is $\sum_{w \in W} H(w)\, l(w)$, *i.e.*, total **information-distance product**. Note that the 1-probability of a wire w, p_1^w depends on the input assignment and hence let $S_{\vec{x}}(C)$ refer to the spatial entropy of a circuit C on the input assignment $\vec{x}$. The average case spatial entropy is the average of $S_{\vec{x}}$ over all the 2^n input assignments, $\frac{\sum_{\vec{x} \in \{0,1\}^n} S_{\vec{x}}}{2^n}$.

Note that a more general definition of spatial entropy can charge cost $f(l)$ for sending one bit of information through a distance l for some function f. The definition we have provided uses an identity function $f(x) = x$. This definition makes spatial entropy equivalent to switching energy for a CMOS like VLSI model, where energy cost is linear in the length of the wire. As we have shown [Tya89], transmission of k information bits causes $\Omega(k)$ switching. Hence the switching energy contribution for this transmission is $\Theta(kl)$. However its spatial entropy contribution is also kl.

3 Computational Action

In this section, we define computational action and propose a principle of least computational action. The intent is to keep it in the form of action from Newtonian mechanics. Note that the action attributes are characteristics of how information is gained in the circuit embedding within the geometric constraints of 2-dimensions.

Spatial entropy is going to serve the role of kinetic energy and distance in the following sense. Let $se(t)$ and $se(t+1)$ be the spatial entropy of the computation at time t and $t+1$. Then the information traveled distance $se(t+1) - se(t)$ in one time unit giving rise to information velocity $se(t+1) - se(t)$. $se(t+1) - se(t)$ is also the energy needed for this information gain. This will be the action component corresponding to kinetic energy. We say that a conservative information force is present when the following is true. Let us say that the wire carrying the information at time t has length l. Note that we are assuming that the wire lengths are small enough that they can be charged or discharged within one clock cycle. The situation changes if we go to a delay model linear or quadratic in wire length. The wire of length l can represent up to $\log l$ information since the packet of information sent from one end could be in any of the l discrete positions. This corresponds to how far has the voltage profile propagated in a CMOS wire. Only k bits of information about the problem might have been gained by this charge transfer of potential information $\log l$. We say that there is an information force equivalent to $k - \log l$ in this situation. If there were m wires active between time t and $t+1$ then the potential energy is $mk - \log l$ assuming similar wire lengths for all of them. Rationale behind this decision is that this much energy can be gained in the reverse phase of a reversible computation. Note that the force is present only if one end of the active wire was not occupied by valid information. If both ends of a wire have valid data, it is considered as a single information object with larger mass. Now the computational action can be defined as $\sum_{t=0}^{T} \Delta_t se - \text{potential energy}_t$. The principle of least computational action states that given certain amount of time T and logical entropy of the problem, a circuit that minimizes computational action is an optimal circuit.

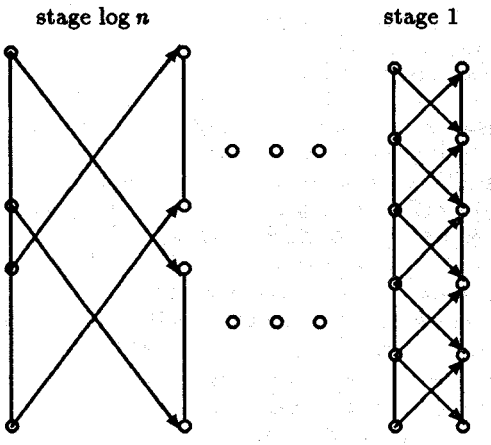

Figure 1: A Barrel Shifter

Let us illustrate this principle with examples of *shifter* designs. We first describe 3 designs for a shifter.

barrel shifter: An n-bit barrel shifter with $\log n$ stages is shown in Figure 1. The first stage wires are assumed to have length $n/2$, the second stage wires have length $n/4$ and in general the ith stage wires, for $1 \leq i \leq \log n$, have length $n/2^i$. Then the total spatial entropy and energy of this shifter is $\approx n^2$.

square shifter: A square shifter is a snaked-around version of the linear chain of shift registers, as described in Ullman [[Ull84], page 69] and illustrated

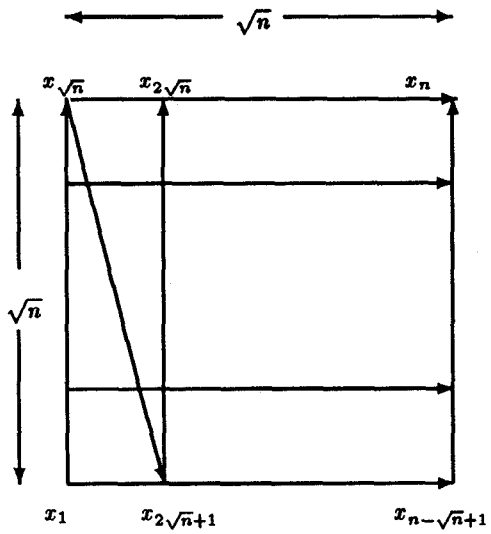

Figure 2: A Square Shifter

in Figure 2. The n bits to be shifted are stored along a $\sqrt{n} \times \sqrt{n}$ array. The rows are numbered bottom-up from 1 to $\sqrt{n}$ and columns left-right from 1 to $\sqrt{n}$. The least significant input bit x_1 is stored at $(1,1)$ position, $x_{\sqrt{n}}$ at $(\sqrt{n},1)$, $x_{\sqrt{n}+1}$ at $(1,2)$, and x_n at $(\sqrt{n},\sqrt{n})$. The first half control bits $c_{\log n} \ldots c_{(\log n)/2+1}$ specify the horizontal shift amount and the least significant half bits specify the vertical shift amount. It takes area $\sqrt{n} \times \sqrt{n}$ and time $O(\sqrt{n})$. This is also a sequential circuit. The expected vertical and horizontal shift amount is $\sqrt{n}/2$. Each horizontal and vertical wire carries $\sqrt{n}/2$ bits of information during shifting. Assuming all the wire lengths to be 1, the total spatial entropy and energy of this shifter is $n^{3/2}$.

hybrid barrel-square shifter: We can design a hybrid shifter which is ET optimal and works in time $3 \log n$. The n input bits are present in a $\log n \times n/\log n$ array initially which is connected along both rows and columns (similar to square-shifter). The horizontal shift between any of the $\log n$ columns corresponds to a shift by $n/\log n$ bits. Hence top $\log \log n$ stages of barrel shifter can be accounted for by decoding MSB $\log \log n$ bits of shift amount and moving the data in the array along that many columns. This takes time $\log n$ and energy $n \log n$. This array is connected to a $n/\log n \times n/\log n$ barrel shifter. Each column of array is shifted into barrel shifter (which is pipelined) in turn and the whole shifting is done in time $3 \log n$. The energy cost for barrel shifter is $n^2/\log n$ leading to total energy and spatial entropy $n^2/\log n$.

Consider the principle of least computational action as applied to the barrel shifter (in Figure 1). Let us compute $\Delta_t se$. At time t, $se(t)$ is given by $n^2/2^t$. Hence $\Delta_t se$ is $n^2/2^{t+1}$. In a barrel shifter since data moves in a wave, at time t data is traversing the wires in stage $t+1$ for $0 \leq t < \log n$. Hence there are many unoccupied nodes in this circuit at any given time. This gives rise to information force. At time t, the wires are of length $n/2^{t+1}$. Each stage removes logical entropy equal to one bit. Hence the information force is $1 - \log n + t + 1 = t + 2 - \log n$. There are n active wires at any given time (this is analogous to information mass) leading to potential energy $n(t + 2 - \log n)$. The computational action for a barrel shifter is $\sum_{t=0}^{\log n - 1} n^2/2^{t+1} - n(t + 2 - \log n)$ which is $\frac{(n^2-n)}{2} + \frac{n \log^2 n}{2}$. Accomplishing barrel shift with fanin 2 gates in $\log n$ time cannot be done without information force. A circuit with information force is not ET optimal as it adds a positive value to the computational action. We will justify these statement in a while.

Now let us consider the square shifter. It gains n in se for each time unit and there is no information force since each node in the circuit is occupied all the time. All the n nodes containing valid data act as a single object of mass n. Hence its computational action is $\sum_{t=0}^{\sqrt{n}-1} n$ which is $n^{3/2}$. The hybrid shifter is an example of a barrel shifter which works fast ($3 \log n$) and yet does not have information force for most time units. The computational action for the first $\log n$ time units (when computation is limited to the array) is $n \log n$. Once the $n/\log n$-bit barrel shifter startes getting filled at time $\log n + 1$, there exists a decreasing force until time $2 \log n$ which increases again as the shifter is emptied between time $2 \log n + 1$ and $3 \log n$.

An interesting observation that can be made from the form of principle of least computational action is that when there is no information force, a circuit that covers spatial entropy at a uniform rate (uniform power dissipation) is the optimal one. This is because then the computational action reduces to the sum of spatial entropy gained at each time unit. Geometrically, the wires to remove spatial entropy are built in the circuit. A circuit that removes different amounts of spatial entropy at different time units must have some unused wires which will give rise to information force. Hence the power dissipation must be at a uniform rate. Note that the square shifter has a uniform power consumption proportional to n and the hybrid shifter also has almost uniform power consumption of $n^2/\log n$. They are both ET optimal as well.

Can we prove that a circuit that minimizes computational action is indeed ET optimal? Note that the first term (kinetic energy) adds up to total energy (spatial entropy) of the circuit. If a circuit minimizes this term for a computation over time T, then it must also be minimizing the ET product for that function. The presence of information force only increases computational action. Hence unless the wires are short (preferably unit length wires to gain unit information), computational action tends to increase. Note that in two dimensional space, a node can communicate with a constant number of neighbors. However, if we wish to lower the time of computation we need to propagate information fast along long wires. This is the problem with barrel shifter. We will provide details of space constrained communication lower bounds in the detailed version of this paper.

The ET lower bounds derived in Tyagi [Tya89] provide a lower bound on computational action as well. In particular, a function with information complexity I will have

an energy lower bound of $I^{3/2}$ and ET lower bound of I^2. If the circuit is combinational (USM), it implies that an information force ought to exist. The lower bounds for such computations are I^2 for both E and ET.

4 Conclusions

The objective of this paper was to be able to capture the global AT^2 and ET optimality with a statement about how certain circuit attribute (called computational action) relates between two consecutive time units. We proposed one definition for computational action and a corresponding principle of least computational action. We believe that circuits satisfying the principle of least computational action are also ET optimal. This work is still in progress and hence the results reported are preliminary.

Suresh Rajgopal and I are also working on a similar notion of action in the domain of Boolean cube representation of a Boolean function. We hope to be able to develop a principle of least computational action based on an attribute on the *on-set* and *off-set* distribution. Some preliminary work along these lines is reported in Rajgopal [Raj92].

References

[AA80] H. Abelson and P. Andrae. Information Transfer and Area-Time Tradeoffs for VLSI Multiplication. *CACM*, January 1980.

[ACR88] A. Aggarwal, A. K. Chandra, and P. Raghavan. Energy Consumption in VLSI Circuits. In *Proceedings of ACM Symposium on Theory of Computing*, pages 205–216. ACM-SIGACT, 1988.

[AUY83] A. Aho, J.D. Ullman, and M. Yannakakis. On Notions of Information Transfer in VLSI Circuits. In *Proceedings of ACM Symposium on Theory of Computing*, pages 133–139. ACM-SIGACT, 1983.

[BK80] R.P. Brent and H.T. Kung. The Chip Complexity of Binary Arithmetic. In *Proceedings of ACM Symposium on Theory of Computing*, pages 190–200. ACM-SIGACT, 1980.

[JK84] J. Ja'Ja' and V.K.P. Kumar. Information Transfer in Distributed Computing with Applications to VLSI. *Journal of the ACM*, pages 150–162, January 1984.

[Kis82] G. Kissin. Measuring Energy Consumption in VLSI Circuits: a Foundation. In *Proceedings of ACM Symposium on Theory of Computing*, pages 99–104. ACM-SIGACT, 1982.

[LS81] R.J. Lipton and R. Sedgewick. Lower Bounds for VLSI. In *Proceedings of ACM Symposium on Theory of Computing*, pages 300–307. ACM-SIGACT, 1981.

[MC80] C. Mead and L. Conway. *Introduction to VLSI Systems*. Addison-Wesley, Reading, Mass., 1980.

[MS82] K. Mehlhorn and E.M. Schmidt. Las Vegas is Better than Determinism in VLSI and distributed computing. In *Proceedings of ACM Symposium on Theory of Computing*, pages 330–337. ACM-SIGACT, 1982.

[PS84] C. Papadimitriou and M. Sipser. Communication Complexity. *Journal of Computer and System Sciences*, pages 260–269, 1984.

[Raj92] S. Rajgopal. *Spatial Entropy — A Unified Attribute to Model Dynamic Communication in VLSI Circuits*. PhD thesis, Department of Computer Science, University of North Carolina, Chapel Hill, October 1992.

[Tho79] C.D. Thompson. Area-Time Complexity for VLSI. In *Proceedings of ACM Symposium on Theory of Computing*, pages 81–88. ACM-SIGACT, 1979.

[Tya88] A. Tyagi. *The Role of Energy in VLSI Computations*. PhD thesis, Department of Computer Science, University of Washington, Seattle, 1988. *Available as UWCS Technical Report Number 88-06-05.*

[Tya89] A. Tyagi. Energy-Time Trade-Offs in VLSI Computations. In *Proceedings of the Ninth Conference on Foundations of Software Technology & Theoretical Computer Science*, pages 301–311. Lecture Notes in Computer Science #405, Springer-Verlag, 1989. *An extended version submitted to IEEETC.*

[Ull84] J.D. Ullman. *Computational Aspects of VLSI*. Computer Science Press, Rockville, Md., 1984.

[Vui83] J. Vuillemin. A Combinatorial Limit to the Computing Power of VLSI Circuits. *IEEE Transactions on Computers*, pages 294–300, March 1983.

[Yao79] A.C. Yao. Some Complexity Questions Related to Distributed Computing. In *Proceedings of ACM Symposium on Theory of Computing*, pages 209–213. ACM-SIGACT, 1979.