RANDOM WALKS, UNIVERSAL TRAVERSAL SEQUENCES, AND THE COMPLEXITY OF MAZE PROBLEMS

Romas Aleliunas[1]
Richard M. Karp[2]
Richard J. Lipton[2]
Laszlo Lovász[3]
Charles Rackoff[1]

I. INTRODUCTION

It is well known that the reachability problem for directed graphs is logspace-complete for the complexity class NSPACE(log n), and thus holds the key to the open question of whether DSPACE(log n) = NSPACE(log n) ([3,4,5,6]). Here as usual DSPACE(log n) is the class of languages that are accepted in log n space by deterministic Turing Machines, while NSPACE(log n) is the class of languages that are accepted in log n space by non-deterministic ones. The reachability problem for undirected graphs has also been considered ([5]), but it has remained an open question whether undirected graph reachability is logspace-complete for NSPACE(log n). Here we derive results suggesting that the undirected reachability problem is structurally different from, and easier than, the directed version. These results are an affirmative answer to a question of S. Cook.

Cook's original question was about connected n-vertex graphs G which are regular of some fixed degree d. At each vertex v, let the edges incident with v be given the distinct labels $0,1,\cdots,d-1$. *Warning:* no assumption is made about the consistency of the labelling - each endpoint of an edge may label it in a different manner. A sequence σ in $\{0,1,\cdots,d-1\}^*$ is said to *traverse* G from v if, by starting at v and following the sequence of edge labels σ, one covers all the vertices of G. Call σ *n-universal* if it traverses every n-vertex G with degree d, from every starting point v. Cook's original question was: are there always "short" (i.e., of length polynomial in n) n-universal sequences?

The motivation for this question lies in the attempt to prove lower bounds on the space complexity of the reachability problem. This attempt focuses its attention on n-vertex directed graphs for some fixed n. Then one attempts to prove that *any* machine with limited space cannot solve the reachability problem for these graphs. One approach is to try to prove an even stronger result: the reachability problem cannot be solved *non-uniformly* in logspace. This would mean that, even if we allowed a sequence $\{T_n\}$ of two-way finite automata, where T_n was responsible only for solving the reachability problem on input strings of length n, the number of bits needed to store the internal state of T_n would grow faster than $\log n$. Many of the methods used to attack this problem work equally as well on undirected graphs; hence it is natural to consider the non-uniform complexity of the reachability problem for undirected graphs. If short n-universal sequences always exist, then not only is the non-uniform complexity of the undirected problem logspace, but the machines that demonstrate this are very simple. Each machine T_n does nothing more than non-adaptively follow a set of move instructions and then stop! One of our main results is that universal sequences do exist.

Theorem: There is an n-universal sequence of length $O(n^3 \log n)$. (The implied constant depends only on the fixed degree d.)

The proof of this theorem depends on an analysis of random walks in undirected graphs. Consider a random walk that starts at a vertex v of a graph G, and whenever it reaches any vertex w, chooses an edge at random from those edges incident with w, and traverses it. Let T be the expected number of edges traversed before all of G has been visited.

Theorem: If G has n vertices and e edges, then

$$T \leq 2e(n-1) \quad .$$

Thus, there is a polynomial time logspace probabilistic algorithm for traversing an undirected graph. In a similar vein, we present polynomial-time logspace probabilistic algorithms for testing whether a graph is bipartite and for the reachability question for undirected mazes. It should be noted that each of these algorithms will have a small probability of failure which can be made arbitrarily small - in a sense one can trade accuracy for space.

The existence of short ($O(n^3 \log n)$) universal sequences is obtained from the random walk results by the probabilistic method of Erdös. The method consists in showing that the expected number of pairs G,v that a random sequence of length $cn^3 \log n$ fails to traverse is < 1; hence, some universal sequence of this length must exist.

Our analysis of random walks on graphs is also the starting point for an investigation of certain interception games. We show that, when an individual (*the mouse*) moves among the vertices of a graph but is oblivious of its pursuer (*the cat*), it can be intercepted in polynomial expected time even when the cat follows an extremely simple random walk policy. The bound on the expected time depends on the complexity of the mouse's motion,

[†]This research was supported in part by NSF Grants MCS77-09906 and MCS77-12517 and ONR Grant N00014-75-C-0752.

[1]Dept. of Computer Science, University of Toronto
[2]Computer Science Div., University of California, Berkeley
[3]Bolyai Institute, University of Szeged, Hungary

but in no case exceeds $O(e^2n)$, even when the mouse computes its trajectory nonrecursively and/or probabilistically.

Our probabilistic graph traversal algorithms raise a number of questions about complexity classes. If a problem is in DSPACE(log n) then it is clearly solvable deterministically in log n space and polynomial time. Similarly, the restriction to polynomial time is superfluous with respect to nondeterministic logspace. With respect to probabilistic acceptance the situation appears to be quite different. Let RSPACE(log n) denote the class of languages recognized by probabilistic algorithms operating in logspace. Let $\mathrm{RSPACE}^{\mathrm{poly}}(\log n)$ denote the class of languages recognized by probabilistic algorithms operating in logspace and polynomial time. It is known [1] that RSPACE(log n) = NSPACE(log n). The results stated earlier imply that the undirected reachability problem is in $\mathrm{RSPACE}^{\mathrm{poly}}(\log n)$ and that the directed reachability problem is in this class of languages if and only if $\mathrm{RSPACE}^{\mathrm{poly}}(\log n) = \mathrm{RSPACE}(\log n)$. It remains an open question whether or not these two classes are equal.

2. RANDOM WALKS ON UNDIRECTED GRAPHS

Let G be an undirected connected graph with n vertices. Let d(i) denote the number of edges incident with vertex i and let $\Delta(i,j)$ denote the distance between vertices i and j. Consider a random walk on G, i.e., a Markov chain in which the states are the vertices of G, and the transition probability p_{ij} is given by:

$$p_{ij} = \begin{cases} 0, & \text{if } \{i,j\} \text{ is not an edge} \\ \frac{1}{d(i)}, & \text{otherwise} \end{cases}$$

Let T(i,j) denote the expected number of transitions until j is reached, starting from i; in particular, T(i,i) is the mean recurrence time of vertex i. Also let n denote the number of vertices and e the number of edges of G.

Lemma 1: for each vertex i of G,

$$T(i,i) = \frac{2e}{d(i)} .$$

Proof: (cf. [2]). Let $\pi(i)$ be the stationary probability of vertex i. Then $\pi = (\pi(1),\cdots,\pi(n))$ is the unique vector satisfying

$$\pi P = \pi \text{ and } \pi(1) + \cdots + \pi(n) = 1.$$

By direct substitution it is easy to confirm that $\pi(i) = \frac{d(i)}{2e}$. Then, from the fact that the mean recurrence time of a state is the reciprocal of its stationary probability, $T(i,i) = \frac{1}{\pi(i)} = \frac{2e}{d(i)}$. □

Lemma 1 has the following useful interpretation. Let {i,j} be any edge of G. Then the long-run frequency with which this edge is traversed from v to w is $\frac{1}{2e}$.

Lemma 2: If vertices i and j are adjacent in G, $T(i,j) + T(j,i) \leq 2e$ with equality if and only if {i,j} is a cut edge of G.

Proof: From the observation that all transitions have the same long-run frequency, it follows that T(i,j) + T(j,i), the expected number of transitions during a round trip from i to j and back, is exactly 2e times the expected number of occurrences of any particular transition during such a round trip. The result then follows by noting that the expected number of transitions from i to j along edge {i,j} during a round trip is ≤ 1, with equality if and only if {i,j} is a cut edge of G. □

Lemma 3: for all i and j, $T(i,j) + T(j,i) \leq 2e\Delta(i,j)$.

Proof: By induction on $\Delta(i,j)$ and Lemma 2. □

Define $T(i,\cdot)$, the *traversal time* starting from i, as the expected number of transitions required to visit all the vertices of G, starting from i.

Theorem 4: $T(i,\cdot) \leq 2e(n-1)$.

Proof: Let H be any spanning tree of G. Then there is a walk beginning and ending at i which traverses each edge of H exactly once in each direction. Let the successive vertices visited in this walk be $i_0, i_1, i_2, \cdots, i_{2n-2}$ where $i = i_0 = i_{2n-2}$. Clearly, $T(i,\cdot)$ is less than or equal to the expected time required for the Markov chain to visit the vertices $i_0, i_1, i_2, \cdots, i_{2n-2}$ in the indicated order. But this expectation is

$$T(i_0,i_1) + T(i_1,i_2) + \cdots + T(i_{2n-3},i_{2n-2})$$
$$= \sum_{\{i,j\} \in H} T(i,j) + T(j,i)$$

and by Lemma 2 this is $\leq 2e(n-1)$. □

This theorem supplies, of course, the bound promised earlier. Note, this bound depends critically on the undirected nature of the graphs. For arbitrary directed graphs T(i,j) can be finite but exponential in size.

3. RANDOM WALKS ON EULERIAN DIGRAPHS

A strongly connected directed graph G with indegree equal to outdegree at every vertex is an *Eulerian digraph*. In this section we will show that random walks on these digraphs enjoy many of the same properties as random walks on undirected graphs.

The notation used is essentially the same as in the last section. One difference is that d(i) is now the outdegree of vertex i. Note, since any undirected graph can be considered as an Eulerian digraph the last section's results are logically subsumed by the following lemmas.

Lemma 5: For each vertex i, $T(i,i) = \frac{e}{d(i)}$.

Lemma 6: If $<i,j>$ is an edge, then $T(i,j)+T(j,i) \leq e$.

The proofs of Lemmas 5 and 6 are virtually identical with those of Lemmas 1 and 2.

Let $\Delta(i,j)$ denote the distance between i and j in the *undirected* graph in which every directed edge $<i,j>$ of G is replaced by an undirected edge $\{i,j\}$.

Lemma 7: $T(i,j)+T(j,i) \leq e \cdot \Delta(i,j)$.

Proof: Use Lemma 6, induction on $\Delta(i,j)$ and the "triangle inequality" $T(i,k)+T(k,j) \geq T(i,j)$. □

We now turn our attention to an application of Lemma 7. In particular we will construct a probabilistic algorithm which operates in logspace and polynomial time and accepts those strings that are the descriptions of labelled graphs such that the connected component that contains vertex 1 is *not* bipartite (i.e., contains an odd cycle). The exact details of how graphs are encoded as strings is unimportant provided we can do the following within the bounds of logspace:

1) determine those vertices which are adjacent to a given vertex;

2) remember or "mark" one vertex with a special marker.

The probabilistic algorithm then proceeds as follows:

Initially it "marks" vertex 1 and sets a parity counter to 0. It then simply executes a random walk as described in section 2. At each step the counter is incremented 1 mod 2. If it ever returns to the marked vertex and the counter is odd (i.e., contains the value 1) then it stops and accepts the graph as not bipartite.

Note, the algorithm has available to it only random 0's and 1's with equal probability. It can, however, simulate a random d-way choice in logspace in an unbiased way. To say that this algorithm works means that if the component of G containing 1 is not bipartite then with probability $>\frac{1}{2}$ it will determine this in polynomial time. The key to seeing this is to examine a derived digraph $\tilde{G}$. $\tilde{G}$ has the vertex set of all pairs $<v,k>$ where v is a vertex of G and $k=0,1$; if $<x,y>$ is an edge of G then

$$<<x,i>,<y,i+1>> \quad (i=0,1)$$

is an edge of $\tilde{G}$ ($i+1$ is taken mod 2 of course). The importance of $\tilde{G}$ is: a run of the above probabilistic algorithm can be viewed as a random walk on $\tilde{G}$ starting at $<1,0>$ and terminating if $<1,1>$ is ever reached. Clearly, $\tilde{G}$ is an Eulerian digraph if and only if $\tilde{G}$ is strongly connected. The latter holds exactly when the component of G containing vertex 1 is not bipartite. Since $\tilde{G}$ has $2e$ edges and $2n$ vertices, it follows that if $<1,0>$ can reach $<1,1>$ it can do so in less than $2n$ steps; by Lemma 7 it follows that the expected time to reach $<1,1>$ is at most $4en$. Then Markov's Theorem shows that in time $9en$ the probability that the algorithm discovers an odd cycle is $>\frac{1}{2}$. This completes the argument that the algorithm is correct.

4. UNIVERSAL SEQUENCES

Recall that a sequence σ in $\{0,1,\cdots,d-1\}^*$ is *n-universal* if given any n-vertex connected graph of fixed degree d and any vertex v, starting at v and following the labels of the edges of G as described by σ one visits all the vertices of G. The existence of such sequences of length $o(n^3 \log n)$ is a consequence of the main theorem on random walks on undirected graphs.

Theorem 8: There is an n-universal sequence of length $O(n^3 \log n)$. (The implied constant depends only on the fixed degree d.)

Proof: Let $\tilde{\sigma}$ be a random sequence from $\{0,1,\cdots,d-1\}^*$ of length $2dn(n-1)(dn+2)\lceil \log_2 n \rceil$. Here we of course insist that each element of this sequence is selected independently and moreover that each of the d labels is equally likely. Also let $\tilde{X}_{G,v}$ be a family of random variables indexed by n-vertex connected regular graphs (with degree d) with a distinguished vertex v which is defined as follows:

$$\tilde{X}_{G,v} = \begin{cases} 1, & \text{if starting at } v \text{ in } G \\ & \text{and following } \tilde{\sigma} \text{ one fails} \\ & \text{to visit all of } G; \\ 0, & \text{otherwise} \end{cases}$$

Let $\tilde{Y} = \sum \tilde{X}_{G,v}$ where the sum is over all such graphs and vertices. By the definition of $\tilde{Y}$, if $\tilde{Y}$ equals 0 then $\tilde{\sigma}$ is n-universal: this follows since $\tilde{Y}$ counts the number of "counter examples" to the assertion "$\tilde{\sigma}$ is n-universal." The key to the probabilistic method is to show that, when $\tilde{\sigma}$ is a random sequence, the expectation of $\tilde{Y}$, $E[\tilde{Y}]$, is <1; this will clearly imply that a universal sequence of the required length exists. Since expectation is linear,

$$E[\tilde{Y}] = \sum E[\tilde{X}_{G,v}].$$

We shall show that, for any G and v,

$E(X_{G,v}) \leq 2^{-(dn+2)\lceil \log_2 n \rceil} \leq n^{-dn+2}$. To see this, consider $\tilde{\sigma}$ as the concatenation of random sequences $s_1, s_2, \cdots, s_{(dn+2)\lceil \log_2 n \rceil}$ each of length $2dn(n-1)$. By Theorem 4 the expected traversal time for G from vertex v is $\leq dn(n-1)$. Hence, by Markov's theorem, the probability that s_1 traverses G from v is $\geq \frac{1}{2}$. Similarly, the probability that s_2 traverses G, starting from whatever vertex w is occupied at the end of s_1, is $\geq \frac{1}{2}$, etc. The probability that none of the sequences $s_1, s_2, \cdots$ traverse G is

$\leq 2^{-(dn+2)\lceil \log_2 n \rceil} \leq n^{-(dn+2)}$. Thus, $E[\tilde{X}_{G,v}] \leq n^{-dn+2}$. Since the number of labelled n-vertex graphs with degree d is less than n^{dn+1} it follows that $E[\tilde{Y}] \leq n^{dn+1} \cdot n^{-dn+2} \leq n^{-1} < 1$. □

By contrast with this result, universal sequences for the traversal of n-vertex directed graphs in which each vertex has outdegree d must be of length at least $d^n + n-1$.

5. INTERCEPTION PROBLEMS

In this section we consider certain problems of strategy related to the following situation. Two individuals, *the cat* and *the mouse*, are moving from vertex to vertex of a connected graph. Each move requires one unit of time. The cat's object is to intercept the mouse, i.e., to occupy the same vertex at the same time. Neither individual senses the other's position until interception actually occurs. In this respect the situation differs from conventional pursuit games in which each individual can modify his trajectory according to his observations of the other's motion.

We will assume throughout that the cat's behavior is very simple, and requires no global knowledge of the graph: he merely alternates between phases in which he delays, or waits, at a fixed vertex, and phases in which he follows a random walk on the graph. Various policies for the cat will differ according to the stochastic experiment he conducts in order to decide when to switch between waiting and walking.

The Case of a Finite-State Mouse

We first consider the case where the mouse is a deterministic finite-state automaton. Each of his moves and each of his state transitions is determined by his present state and by information local to the vertex he currently occupies; this information is in the form of labels on the edges incident with that vertex.

We assume that

(i) there is a set $\{1,2,\cdots,d\}$ of possible labels.

(ii) for each ordered pair $<v,e>$ such that v is a vertex and e is an edge incident with v, there is an edge label $L(<v,e>)$; if edge e joins vertices v and w, the labels $L(<v,e>)$ and $L(<w,e>)$ are not necessarily equal.

(iii) no label appears twice at the same vertex; i.e., if e and e' are distinct edges incident with v, then $L(<v,e>) \neq L(<v,e'>)$.

It follows from these assumptions that each vertex has degree $\leq d$.

The behavior of the mouse is determined by:

a finite set of states Q;

an initial state q_0;

an initial vertex v_0;

a *transition function* $\delta: Q \times D \to \{1,2,\cdots,d\} \times Q$, where D is the family of nonempty subsets of $\{1,2,\cdots,d\}$. This function satisfies the following restriction: if $\delta(q,S) = x$ then $x \in S$.

When the mouse is at vertex v in state q, it observes the set of labels

$$S = \{L(<v,e>) \mid e \text{ is incident with } v\} .$$

Then, if $\delta(q,S) = x,q'$, it moves along the unique edge $\tilde{e}$ such that $L(<v,\tilde{e}>) = x$, and it enters state q' .

We consider a simple policy for the cat. It starts at some vertex w_0, and at each step it uses the following stochastic decision rule:

if the current vertex v has degree d then

(i) with probability $\frac{2}{d+2}$, remain at vertex v;

(ii) with probability $\frac{d}{d+2}$, move along a randomly chosen edge incident with v.

Thus, the cat can be regarded as executing a random walk on a graph H obtained from G by adding a loop at each vertex (a loop contributes 2 to the degree of its vertex).

Theorem 9: The expected time for interception of any q-state mouse on a n-vertex graph with edge labels drawn from $\{1,2,\cdots,d\}$ is $\leq nq + (d+2)n(n-1) + (d+2)n^3q^2$.

Proof: Define a *total state* for the mouse as a pair $<v,q>$, where v is a vertex of G and q is an internal state. The next total state is completely determined by the present total state. Thus the mouse eventually enters, and repeats forever, a cycle of distinct total states.

The time required for the cat to intercept the mouse is the sum of three terms:

(i) t_1, the time until the mouse enters his cycle of total states;

(ii) t_2, the time thereafter until the cat first visits a vertex which occurs in some total state in the cycle;

(iii) t_3, the time thereafter until the cat intercepts the mouse.

Clearly, $t_1 \leq nq$. To estimate t_2, let j be a vertex occurring in a total state of the cycle. Given that the cat is at vertex i at time t_1, the expected value of t_2 is $\leq T^H(i,j)$.† Hence,

† $T^H(i,j)$ denotes the expected number of steps for a random walk on H to reach j, starting at i .

$E[t_2] \le \max_i T^H(i,j)$. Since H has n vertices and at most $(\frac{d}{2}+1)n$ edges, Lemma 3 shows that $\max_i T^H(i,j) \le (d+2)n(n-1)$, so $E[t_2] \le (d+2)n(n-1)$.

Finally, we bound the expected value of t_3. Let the sequence of total states on the mouse's cycle be $C = c_0, c_1, \cdots, c_{m-1}$. Let H have vertex set $\{1,2,\cdots,n\}$. Then, once the mouse is in his cycle, the joint motion of the cat and the mouse can be described as a random walk on a digraph K with vertex set $\{1,2,\cdots,n\} \times C$. If H has an edge from v to w, then K has an edge from $<v,c_i>$ to $<w,c_{i+1 \bmod m}>$, for $i = 0,1,\cdots,m-1$; a loop from v to v generates two such edges for each i. It is easy to see that K is Eulerian.

Let x be the vertex occupied by the cat at time $t_1 + t_2$. Let c_i be any total state in the cycle C having x as its first component. Then, the conditional expectation of t_3, given x and c_j, is $\le T^K(<x,c_i>,<x,c_j>)$. The digraph K has $m(d+2)n$ edges, and $\Delta^K(<x,c_j>,<x,c_i>) \le (i-j) \bmod m$, the length of the sequence of transitions from $<x,c_j>$ to $<x,c_i>$ in which the cat remains at x while the mouse progresses around C from c_j to c_i. By Lemma 7 the conditional expectation of t_3 given x is $\le m(d+2)\cdot n\cdot(i-j) \bmod n \le m^2(d+2)n$. Hence, $E[t_3] \le m^2(d+2)n \le n^3q^2(d+2)$. □

A Uniformly Good Interception Policy

We next present an interception policy for the cat which works well, on the average, against any evasion policy for the mouse. By an *evasion policy* for the mouse we mean an infinite sequence $\{v(t)\}$ of vertices; $v(t)$ is interpreted as the vertex where the mouse resides at time t. We do not require that this sequence correspond to a walk along the edges of G, or that it even be computable.

The idea behind the cat's interception policy is to determine by a stochastic experiment when to switch between periods of waiting and periods of executing a random walk on G. The stochastic experiment is so designed that the cat's position quickly becomes quite unpredictable.

The experiment is based on a fact first exploited by Gill [1] in his proof that probabilistic machines determine the same space complexity classes as non-deterministic ones: a probabilistic machine with space s can "count" not just to 2^s but to 2^{2^s}. (In order to count this high just wait until 2^s 1's occur in a row in a random stream of 0's and 1's.)

Let $F = \lceil \log_2(3d(n-1)) \rceil$. In our first version of the interception policy we assume that the cat is given the integer F. At each time step the cat flips a fair coin, and he keeps track of the number of heads since the last tail. As soon as this number becomes equal to F he switches from a waiting period to a random walk period, or vice versa. Implementation of the policy requires $\lceil \log_2 F \rceil$ bits of auxiliary storage to keep track of the count.

Theorem 10: For every evasion policy, the expected time until interception is $\le 576e^2(n-1) = O(e^2n)$.

The proof requires three simple lemmas which we state without proof.

Lemma 11: In a random walk on G starting at vertex i, the expected number of visits to vertex j in the time period $[0,1,\cdots,x-1]$ is $\ge \frac{d(j)}{2e}(x - T(i,j))$.

Lemma 12: For any integer t such that $0 \le t \le 2^F - 1$, the probability that, at time $2^F + F - 1$, the cat is in his first random walk period and has executed t random walk steps (so that his first waiting period is of length $2^F + F - 1 - t$) is $\ge 2^{-(F+2)}$.

Lemma 13: The expected number of flips of a fair coin until F consecutive heads occur is $\le 2^{F+2}$.

Proof of Theorem 10: Let $\{v(t)\}$ be an evasion policy. Let $v^* = v(2^F + F - 1)$, and let P_t be the probability that, after t steps of a random walk on G starting from w_0, vertex v^* is occupied. By Lemma 11,

$$\sum_{t=0}^{2^F-1} P_t \ge \frac{d(v^*)}{2e}(2^F - T(w_0,v^*)) \ge \frac{1}{2e}(2^F - 2e(n-1)).$$

By Lemma 12, the probability that the cat intercepts the mouse at time $2^F + F - 1$, and at that time is executing his first random walk, is

$$\ge 2^{-(F+2)} \sum_{t=0}^{2^F-1} P_t \ge \frac{1}{2e}\,\frac{2^F - 2e(n-1)}{2^{F+2}}.$$

By Lemma 13, the expected time between the end of the Kth random walk period and the end of the $(K+1)$th random walk period is $\le 2\cdot 2^{F+2} = 2^{F+3}$. Hence the expected time until interception is

$$\le \frac{2^{F+3}}{\frac{1}{2e}\,\frac{2^F - 2e(n-1)}{2^{F+2}}} \le 576e^2(n-1).$$

□

If F is not known to the cat, he can simply use the successive trial values $F = 1,2,3,\cdots$. To determine how long to stick with a given trial value F, he flips a fair coin after every alternation between waiting and executing a random walk, and increments F whenever F consecutive heads are observed. Against any evasion policy, the expected time until interception is $O(e^2 n)$, and, with probability tending to 1, the storage requirement can be made $O(\log\log n)$.

6. CONCLUSIONS

There are a number of open questions that remain. First, there is the question of just what is $\mathrm{RSPACE}^{\mathrm{poly}}(\log n)$. Is this class equal to either $\mathrm{DSPACE}(\log n)$ or $\mathrm{NSPACE}(\log n)$? A proof that $\mathrm{RSPACE}^{\mathrm{poly}}(\log n) = \mathrm{NSPACE}(\log n)$ would of course yield a polynomial time logspace probabilistic algorithm for reachability in all directed graphs; on the other hand, a proof that $\mathrm{RSPACE}^{\mathrm{poly}}(\log n) = \mathrm{DSPACE}(\log n)$ would yield a new deterministic logspace algorithm for the reachability problem for undirected graphs. Either result would be very interesting. As usual since

$$\mathrm{DSPACE}(\log n) \subseteq \mathrm{RSPACE}^{\mathrm{poly}}(\log n) \subseteq \mathrm{NSPACE}(\log n)$$

any separation result would be very strong for it would solve the long standing question: is

$$\mathrm{DSPACE}(\log n) = \mathrm{NSPACE}(\log n)\ ?$$

The reachability problem for undirected graphs can be solved in logspace and $O(ne)$ time by a probabilistic algorithm that simulates a random walk, or in linear time and space by a conventional deterministic graph traversal algorithm. Is there a spectrum of time-space trade-offs between these extremes? Also, is there a logspace algorithm for the undirected reachability problem that is never incorrect and, uniformly for all graphs, runs in polynomial-bounded expected time?

A third class of open problems concerns universal sequences. Can the upper bound of $O(n^3 \log n)$ on universal sequences be improved? Currently we know of no lower bound that grows faster than cn. Also of interest is the question of a constructive upper bound: can a constructive sequence yield a polynomial upper bound?

Another aspect of universal sequences is the failure of directed graphs to have polynomial length universal sequences. However, there still remains the whole question of non-uniform space complexity of the directed reachability problem. There are many ways to define this notion; one attractive way is based on the size complexity of finite automata. Suppose L is a language. We say that L has *non-uniform space complexity* $S(n)$ if, for every n, there is a two-way deterministic finite automaton with at most $2^{S(n)}$ states which correctly accepts strings from L and rejects those from $\bar{L}$ when restricted to inputs of length n. Our results here on universal sequences can be viewed as showing that the set of threadable undirected mazes ([6]) has non-uniform complexity $\log n$. Our construction yields a very non-adaptive automaton. Perhaps a more adaptive one can always in polynomial time traverse any directed maze.

A final question concerns the role of symmetry in the complexity of computation. Is there a fundamental principle involved in the fact that a symmetric random walk has a polynomial expected traversal time while a directed one need not? Or is this merely chance?

8. REFERENCES

[1] J. Gill, "Complexity of probabilistic Turing machines," *SIAM J. of Computing*, Vol. 6, No. 4 (1977), 675-695.

[2] F. Göbel and A. A. Jagers, "Random walks on graphs," *J of Stochastic Processes and their Applications*, 2 (1974), 311-336.

[3] J. Hartmanis, N, Immerman, S. Mahaney, "One-way log-tape reductions," *19th Annual Symposium on Foundations of Computer Science* (1978), 65-71.

[4] N. D. Jones, "Space-bounded reducibility among combinatorial problems," *J. of Computer System Sciences*, Vol. 11 (1965), 68-85.

[5] N. D. Jones, Y. E. Lien, W. T. Laaser, "New problems complete for logspace," *MST*, Vol. 10 (1976), 1-17.

[6] W. J. Savitch, "Relations between nondeterministic and deterministic tape complexities," *J. of Computer System Sciences*, Vol. 4 (1970), 122-129.