A MARKOV MODEL OF CYCLIC STRUCTURED PROGRAMS

by

T. C. Wesselkamper

CS 77002-R

Abstract:- The paper defines a class of flowgraphs which possess neither absorbing nor transient states. It gives a necessary and sufficient condition that any transition matrix associated with such a program be primitive. With a fixed flowgraph is associated an equivalence class of programs and an equivalence class of transition matrices. The paper investigates the hypothesis that the equivalence class of processes generated by the programs is modeled by the behavior of the equivalence class of eigenvectors generated by the transition matrices. The geometric implications are considered as well as the statistical behavior of the model as applied to the first fourteen algorithms of the Communications of the ACM.

This paper has been prepared for presentation at the Second Hungarian Computer Conference, Budapest, Hungary; 27 June - 2 July, 1977.

The research reported herein is supported in part by National Science Foundation Grant DCR 74-18108.

A MARKOV MODEL OF CYCLIC STRUCTURED PROGRAMS

T. C. Wesselkamper
Blacksburg, Virginia, U.S.A.

This paper presents a theoretical background and some practical insights into a technique for the prediction of process behavior. It has always been of interest to be able to analyze the high level source code of a program and from it predict the execution time behavior of the related process. Recent advances in technology have made the need for such prediction more urgent. Currently available processors are increasingly microprogrammable, if not by the user then at least by the vendor. Large scale processors are increasingly composed of a multiplicity of microprocessors. In the environment created by these architectures it is possible to achieve great increases in process execution speed by realizing certain tasks within the process in microcode or by passing each of the tasks to a dedicated microprocessor for execution. For such a downward migration to work most efficiently it is necessary to be able to predict which segments of code will be executed most frequently by the process and which, therefore, will produce the greatest saving if speeded up.

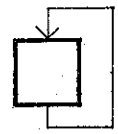
We are first concerned with the flowgraph specification of a program. We model a flowgraph by a finite homogeneous Markov chain and model the execution time behavior of the associated process as the behavior of the principal eigenvector of the transition matrix of the Markov chain. We are interested in determining those blocks of the flowgraph which correspond to the tasks most frequently executed in the process.

Cyclic Structured Flowgraphs

Firstly, we are concerned with flowgraphs which possess neither absorbing states (states which, once entered, are never left) nor transient states (states which, once left, are never reentered).

The research reported herein is supported in part by National Science Foundation Grant DCR 74-18108.

In a positive way we are concerned with flowgraphs which, at the most global level, are represented by the flowgraph:



Secondly, we are concerned with flowgraphs which are developed from the above cyclic construct by the recursive substitution of one of the five constructs in Figure 1 for an action state. (As usual we represent action states by rectangles and represent decision states by rhombi.)

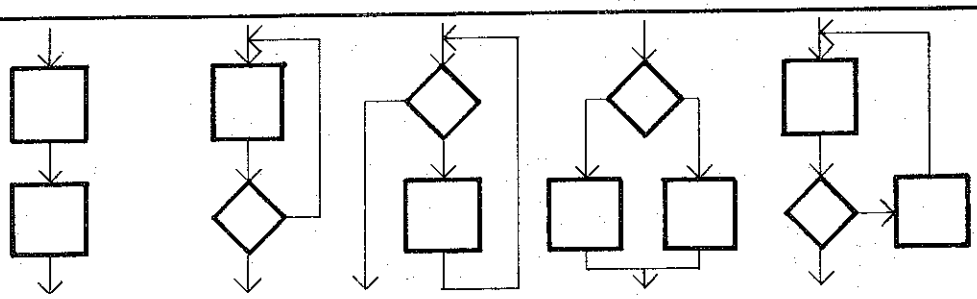


Figure 1

We call a flowgraph produced in this way a cyclic structured flowgraph.

The five constructs of Figure 1 are familiar. All share the characteristic of being one-in-one-out constructs. The first four are frequent features of high level languages: sequencing, repeat ... until, while ... do, if ... then ... else. The fifth construct does not correspond to a control structure in current languages, but occurs frequently in practical situations (in which it is realized by a conditional goto or an exit).

The Model and the RPC Condition

We model a cyclic structured flowgraph by a finite Markov chain. The blocks of the flow graph correspond to the states of a chain. At each action block there is exactly one block which is entered next, that is, the corresponding row of the transition matrix of the chain contains exactly one component equal to one; all other entries in the row are zero. For each decision block in the flowgraph there are exactly two other blocks which are entered with probabilities p (>0) and 1-p respectively.

If A and B are two blocks of a flowgraph we define the directed distance from A to B, d(A,B), to be the minimal number of edges in the set of all paths from A to B, provided that A is different from B. If A = B, then d(A,B) = 0.

We define a cycle for a state S to be a nonempty sequence of directed edges of the flowgraph which sequence originates and terminates at S. The length of a cycle for S is the number of edges in the sequence. We call a cycle of length n an n-cycle. If C is a cycle for a state S we say the C belongs to S or that S possesses C. Consider the example in Figure 2.

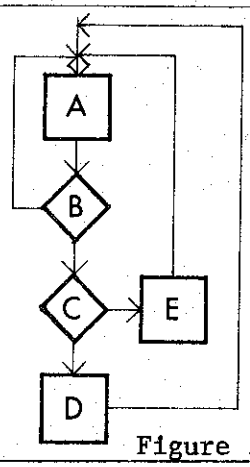


Figure 2

The transition matrix of this flowgraph has the form:

	A	B	C	D	E
A	0	1	0	0	0
B	p	0	1-p	0	0
C	0	0	0	q	1-q
D	1	0	0	0	0
E	1	0	0	0	0

The path AB, BA is a cycle for state A which is of length two. The path AB, BC, CD, DA is a cycle for state A which is of length four.

A stochastic matrix T is called a primitive matrix if there exists a natural number n such that the matrix T^n contains no zero element. In this connection we need to recall that T^n, the nth power of the transition matrix T, represents the probabilities that a process will move from one particular (row) state to another

(column) state in exactly n steps. If an entry in the A^{th} row and B^{th} column of T^n is nonzero, then there is at least one path of length n from A to B.

In the proof of the main results of this section we use a lemma from elementary number theory. [1]

Lemma 1: If m and n are positive integers such that (m,n) = 1, then the Diophantine equation:

$$mx + ny = c$$

has solutions in non-negative integers, x and y, for all c such that $mn - m - n < c$.

Now let us suppose that G is a flowgraph of a cyclic structured program and suppose that S is a state which possesses two cycles of relatively prime length, say m and n.

Lemma 2: There exists a non-negative integer c such that for each p > c, the state S possesses a cycle of length p.

proof: Let $c = mn - m - n$. By Lemma 1 the Diophantine equation $mx + ny = p$ has a solution in non-negative integers since $p > c$. Say that the solution is (x_0, y_0). Beginning at S, traverse the m-cycle x_0 times and then traverse the n-cycle y_0 times. The resulting cycle is of length $mx_0 + ny_0 = p$.

Lemma 3: Each state T of a regular flowgraph G possesses two cycles of relatively prime length.

proof: The regular flowgraph G has a state S which possess two cycles of relatively prime length. By Lemma 2 there exists a positive integer c(S) such that S possesses a cycle of length C' for each $c' \geq c(S)$. Let s' = d(S,T), the directed distance from S to T, and let t' = d(T,S), the directed distance from T to S. Let p' and q' be two relatively prime integers, each greater than s' + t' + c(S).

Now note that $p' - s' - t' > c(S)$ and so S possesses a cycle of length $p' - s' - t'$. Hence the path from T to S followed by the cycle of length $p' - s' - t'$ around S, followed by the path from S to T is a cycle of length p' belonging to T. Similarly there is a cycle of length q' belonging to T.

Corollary: If G is a cyclic structured flowgraph then with each state S of G there is associated a minimal constant c(S) such that for each integer $c' \geq c(S)$, S possesses a cycle of length c'.

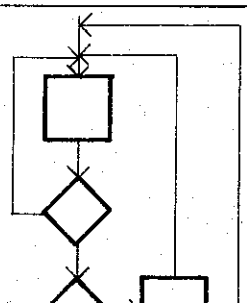
Lemma 4: If S and T are two states of a cyclic structured flowgraph G, then there exists a constant c(S,T) such that for each integer $c' \geq c(S,T)$, there exists a path of length c' from S to T.

proof: Let $c(S,T) = c(S) + d(S,T)$. For each $c' \geq c(S,T)$, cycle around S, $c' - d(S,T)$ times, then go to T.
Hereafter, we let c(S,T) denote the minimum such path.

Theorem: Each cyclic structured flowgraph G is modeled by a Markov process with a primitive transition matrix T if and only if G has a state S which possesses two cycles of relatively prime length.

proof: We need to prove that there exists an integer n such that T^n, the n-th power of T, has no non-zero entries. Let $n = \max \{c(A,B) \; A,B \text{ are states of } G\}$. If P and Q are two states of G then $n \geq c(P,Q)$ and so there is a path of length n from P to Q. Hence the entry in the P-th row and Q-th column of T^n is non-zero. We refer to the condition that a flowgraph have a state which possesses two cycles of relatively prime length as the RPC condition.

Note that for each block in Figure 2 each cycle for the block has a length divisible by 2. However, consider Figure 3.



The associated transition matrix has the form:

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ p & 0 & 1-p & 0 \\ q & 0 & 0 & 1-q \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

The matrix T^6 has no zero entries, that is, between each two blocks there exists a path with length six.

By the Perron-Frobenius theorem [2] a primitive stochastic matrix possesses a (principal) eigenvector with eigenvalue 1; all other eigenvalues of T are strictly less than 1. We have then:

$$\lim_{n \to \infty} T^n = T^* = \begin{bmatrix} \bar{t} \\ \bar{t} \\ \vdots \\ \bar{t} \end{bmatrix}$$ where $\bar{t}$ is the principal eigenvector of T.

In other words, in the limit the probability that a process defined by a cyclic structured flowgraph satisfying the RPC condition will be in a given state is independent of the starting state of the process.

Throughout the rest of this paper we call a cyclic structured flowgraph which satisfies the RPC condition a regular flowgraph.

The Hypothesis of Local Homogeneity

With a particular flowgraph there is associated an equivalence class of programs and an equivalence class of transition matrices. Two programs in the same equivalence class share the same flowgraph at some level of refinement. They may differ in the action to be taken in the action states and in the tests to be performed in the decision states. Transition matrices of the same class have the same fixed pattern of zeros and ones. The values of the entries which are neither zero nor one may differ.

The transition probabilities for the decision states of programs are, in general, not constant since they may be data dependent and hence may vary with time. Herein we adopt the following hypothesis:

Hypothesis: The behavior of the equivalence class of processes generated by the equivalence class of programs with a given regular flowgraph is modeled by the equivalence class of principal eigenvectors of the equivalence class of transition matrices corresponding to the same regular flowgraph.

Viewed from another angle, we recognize that a program is not homogeneous. However, we assume that if viewed locally (with respect to time) the transition probabilities would appear to be nearly constant. Therefore we model the inhomogeneous behavior of the program by an equivalence class homogeneous Markov chains. It is this view which gives rise to the term local homogeneity.

The diagram of Figure 4 suggests these relationships.

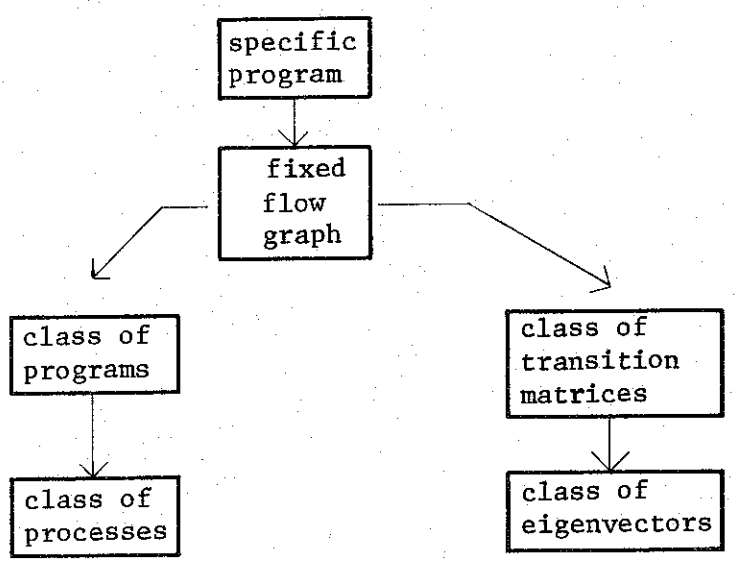


Figure 4

The Geometry of Eigenvectors

Let G be a regular flowgraph with n states, k of which are decision states. Let the n states be labeled by the natural numbers 1, 2, 3, ..., n. There is a subset $j_1, j_2, \ldots, j_k$ of the set 1, 2, ..., n such that for each i, the natural number j_i is the label of a decision state. For each j_i there are two probabilities p_i and $1-p_i$ which are the two nonzero components of the row labeled j_i.

Let T* be the equivalence class of transition matrices corresponding to the flowgraph G.

Firstly, the set $E^* = \{e: e \text{ is an eigenvector of } T \in T^*\}$ determines a subset of an (n-1)-flat in n-space. In particular E* determines a subspace of the portion of the flat $x_1 + x_2 + \ldots + x_n = 1$ which lies in the totally positive unit n-cube. Secondly, if j_a is the label of a decision state of G and if $j_i \neq j_a$ implies that p_i is a constant, then the set of matrices $\{T: 0 < p_a < 1\}$ generates a set of eigenvectors whose intersection with the flat $x_1 + x_2 + \ldots + x_n = 1$ is a straight line. Thirdly, from the above observation it follows that the set E* of eigenvectors determines an open convex set in the flat $x_1 + x_2 + \ldots + x_n = 1$.

Suppose that T is a transition matrix of a regular flowgraph G and that all of the entries of T are fixed except for one decision state, say j_a. The two nonzero entries in the row labeled j_a are p_a and $1-p_a$. Let T_0 denote the matrix T with $p_a = 0$. Let T_1 denote the matrix T with $p_a = 1$. For $0 < p_a < 1$, the matrix T is primitive since the corresponding flowgraph is regular. However the matrices T_0 and T_1 correspond to degeneracies of the flowgraph arising from certain decision states being modified into action states. The degenerate flowgraph may no longer satisfy the RPC condition. Consider, for example, the flowgraph of Figure 5. Suppose that the probability of taking the path 1, 2 is denoted by p and the probability of taking path 1, 4 is denoted by 1-p. For p = 0 and p = 1 the flowgraph of Figure 5 degenerates, respectively into the left and right flowgraphs of Figure 6.

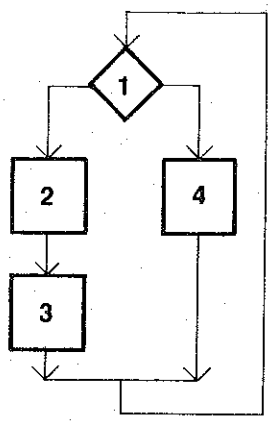


Figure 5

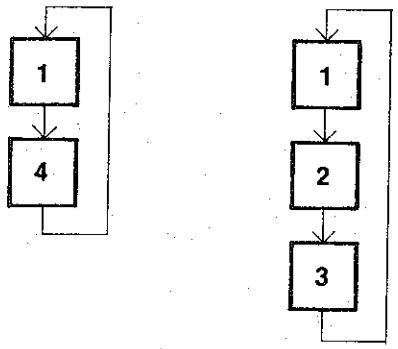


Figure 6

Neither of these degenerate cases satisfies the RPC condition and it is trivially true that

$$\lim_{n \to \infty} T^n$$

does not exist for a transition matrix corresponding to either of these flowgraphs. (The sequences of ascending powers of the two corresponding transition matrices have periods equal to two and three, respectively.)

It does not appear to be easy to take a flowgraph G and the associated equivalence class T* of transition matrices and from T* to determine the measure of the convex set determined by E*. In the absence of such a nice analytic approach we pursue a statistical approach.

The Statistical Behavior of Eigenvectors

As before, let G be a fixed flowgraph with n states labeled 1, 2, ..., n; and with k decision states $j_1, j_2, \ldots, j_k$. For each decision state j_i and let p_i and $1-p_i$ denote the nonzero probabilities in the j_i row of a transition matrix $T \in T^*$. Suppose that we generate an m-sample of transition matrices: $T_1, T_2, \ldots, T_m$, by letting the p_i be values of a uniformly distributed random variate. If $E_1, E_2, \ldots, E_m$ denote the corresponding eigenvectors then the convex hull of $\{E_i\}$ is an estimate of the convex hull of E*. Similarly, the geometric centroid of E* is estimated by the geometric centroid of $\{E_i\}$, that is by:

$$\bar{E} = \sum_i E_i/m,$$

where the summation is performed componentwise.

An intuitive measure of the variability of $\{E_i\}$ (and hence of E*) is the set of cosines $\{c_i\}$, where

$$c_i = \cos(\bar{E}, E_i) = \frac{\bar{E} \cdot E_i}{|\bar{E}||E_i|} .$$

If the set $\{c_i\}$ is very closely clustered about 1, then the centroid $\bar{E}$ is a good approximation of the behavior of the set $\{E_i\}$ and we have a valuable statistical tool for investigating the runtime behavior of the set of processes associated to a set of programs with a common regular flowgraph. Our goal is to develop some insight into the extent to which the behavior of a process is data dependent as over and against the extent to which the behavior of the process is determined by the topology of the flow of its generating program.

Inspired by Halstead [3] we chose to investigate the behavior of the first fourteen algorithms of the Communications of the ACM [4]. It was necessary to modify each of these algorithms so that it was cyclic. Beyond that, modification was not necessary: programmers used the constructs of Figure 1; the RPC condition was always satisfied. Figure 7 provides some information about these fourteen algorithms.

Algorithm	Number of States	Type	Function
1	16	2	n-point quadrature integration
2	6	1	finds root of x = f(x) by secant method
3	46	2	solution of a polynomial by Bairstow-Hitchcock
4	19	1	finds root of f(x) by iterated bisection
5	26	2	Bessel Function I, series expansion
6	5	1	Bessel Function I, asymptotic expansion
7	8	1	Euclidean algorithm
8	12	1	summation of f(x) by Euler transformation
9	25	2	Runge-Kutta integration
10-13	9	1	Evaluation of Chebyshev Polynomials Evaluation of Hermite Polynomials Evaluation of Laguerre Polynomials Evaluation of Legendre Polynomials
14	7	1	Complex exponential integral

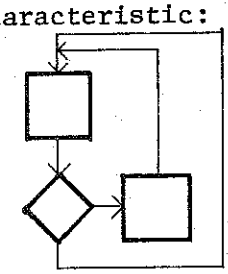
Figure 7

Algorithms 10, 11, 12, and 13 are four distinct algorithms which share the same flowgraph.

For each of the fourteen algorithms one hundred transition matrices were generated. For each transition matrix T_i the corresponding eigenvector E_i was generated by iteratively squaring T_i. The geometric centroid was calculated, as were the set of cosines $\{c_i\}$. A histogram of the set $\{c_i\}$ was plotted. In addition, for each E_i, the indices and values of its three largest components were listed.

There are two very distinctive distribution patterns for the $\{c_i\}$. We call them Type 1 and Type 2. Type 1 behavior is exhibited by the flowgraphs corresponding to Algorithms 2, 4, 6, 7, 8, 10-13, and 14. In these seven cases the $\{c_i\}$ are tightly clustered about the value 1. In each of these cases if one extracts from each sample eigenvector the set of the three largest components of that eigenvector and forms the union of the one hundred 3-sets, that union is, at most, a 4-set.

Algorithms 1, 3, 5, and 9 all exhibit a similar behavior to one another, which is quite different from Type 1 behavior. In Type 2 behavior no values of $\{c_i\}$ are very close to 1. This implies that no element of E_i is very close to $\bar{E}$, although some of the $\{E_i\}$ are very close to one another. The four flowgraphs with Type 2 behavior have a common characteristic: each has as its first level of refinement the form:



One of the two branches from state 2 serves merely as a termination condition. At each iteration a datum is tested which is decremented by state 3 on the other (usually taken) branch. When this datum reaches some base value the process terminates, that is, the cycle begins again with new data.

In this situation it seems unrealistic to model the transition probabilities at state 2 by values of a uniformly distributed random variate on the interval (0, 1). The probability that the termination branch will be taken is, in practice, strictly smaller than the probability that the action branch will be taken. If this modification is made in the model, that is, if the transition probability on the termination branch is taken to be the value of a uniformly distributed random variate on the interval (0, .5) then the behavior called Type 2 behavior disappears. This restriction on the transition probabilities at a decision state corresponds to a reduction in the size of the equivalence class of transition matrices which correspond to a given flowgraph, with a corresponding reduction in the size of the equivalence class of eigenvectors generated by the transition matrices. In using the technique of this paper for the analysis of a real program such partial information about the relative size of probabilities at a decision state can be used to significantly improve the results obtained.

Conclusion: The Validity of the Model

In an effort to test the validity of the hypothesis of this paper the fourteen algorithms were run with randomly generated data uniformly distributed in each case over the appropriate interval. In each of the fourteen cases the number of executions of each task was recorded for each of one hundred sample executions. The centroid of these frequency vectors was calculated and the cosines of the angles between the centroid and the samples was plotted. The correspondence between the model and the runtime behavior was excellent.

Some reservations about the method are in order: 1.) The first fourteen algorithms of CACM are neither large nor typical programs. 2.) It is doubtful that the structures which occur in the first fourteen algorithms is representative of all programs; they are all mathematical functions. 3.) The notion that randomly generated data produces information about typical execution behavior is open to dispute. It may fairly be said that experimental evidence does not now provide any basis for the rejection of the hypothesis.

Several research directions are suggested by the results to date: 1.) Experimentation should be carried out to establish the validity of the model for larger problems and for problems from other applications fields. 2.) Consideration should be given to a model of process behavior in which the Markov chains are inhomogeneous; they will, of course have transition matrices with fixed patterns of zeros and ones. 3.) Consideration should be given to a model of process behavior in which semi-Markov chains are used, thereby taking into account the time which a process spends in a state.

References

1 Alfred Brauer and James E. Shockley, "On A Problem of Frobenius", Journal fur die reine und angewandte Mathematik, B. 211, H. 3-4 (1962) pp. 215-20.

2 Eugene Seneta, Non-Negative Matrices, (London: George Allen and Unwin, 1973)

3 M. H. Halstead, "Natural Laws Controlling Algorithm Structure?", SIGPLAN Notices (Feb. 1972), pp. 19-26.

4 J. H. Wegstein, "Algorithms" CACM 3(1960), pp. 74-5, 174, 240, 311-2, 353, 406.