INFORMATION AND CONTROL 12, 21–22 (1968)

A Note on Segmentation of Computer Programs

A. T. BERZTISS*

Department of Computer Science, University of Pittsburgh, Pittsburgh, Pennsylvania 15213

It has been shown that the cycle picture of the directed graph of a computer program can be used to solve the segmentation problem. A simple algorithm for finding the cycle picture is given.

At times a computer program cannot be accommodated in its entirety in the main store of the computer, in particular if the computer is operated in a time-sharing mode. The program has to be segmented, and there arises the problem of selecting cutting points which minimize the number of segment interchanges between different levels of storage. This is the segmentation problem. One approach to the solution of the problem is to select points spanned by the least number of program loops. Schurmann (1964) has developed a rather complicated algorithm for determining the number of loops which span a point. The purpose of this note is to give a simpler algorithm.

The flow of control in a computer program, i.e., the structure of the program, can be represented by a directed graph, the flowchart graph of the program. Let the statements in the program be numbered consecutively $1, 2, \cdots, n$. Nodes in the flowchart graph correspond to these statements and are also numbered $1, 2, \cdots, n$. Schurmann defines the cycle picture of the flowchart graph as the partial graph comprising only arcs which lie on cycles. His algorithm finds the adjacency matrix C of the cycle picture. If a cut is made between statements i and $i + 1$ in the program, the cut produces two segments, comprising, respectively, statements $1, 2, \cdots, i$ and statements $i + 1, \cdots, n$. Then, in terms of elements of the matrix, the number of program loops spanning the cut is

$$L_i = \sum_{j=1}^{i} \sum_{k=i+1}^{n} c_{kj} .$$

* On leave from the Department of Computation, University of Melbourne, Australia.

Schurmann's complicated method for finding matrix C is based on the theory of gathering nodes. Our algorithm finds C from the path matrix B of the flowchart graph, defined as follows:

$b_{ij} = 1$, if there exists a path from node i to node j;

$b_{ij} = 0$, if there exists no such path.

The algorithm is extremely simple:

1. Find path matrix B.
2. Define C: $c_{ij} = a_{ij} \times b_{ji}$, $i, j = 1, 2, \cdots, n$,

where the a_{ij} are elements of the adjacency matrix A of the flowchart graph.

Since there exists a very efficient algorithm for finding the path matrix from the adjacency matrix (Warshall, 1962; Ingerman, 1962), the segmentation problem may be considered solved for programs which can be cut into segments of required size with no loops spanning the cuts. If, however, it is necessary to use cutting points which are spanned by loops, a knowledge of the iteration counts of the loops is required for optimum segmentation. These counts may not be available. Therefore it seems unlikely that a general solution of the segmentation problem can be found.

The new algorithm gives precise diagnostic information if applied to the tracing of loops in a network. The increase of running time with an increase in the number of nodes is slightly greater than for the algorithm which checks for loops by reduction of the adjacency matrix (Marimont, 1959). But if a network is found inconsistent, Marimont's algorithm cannot distinguish between an arc which lies on a loop and an arc which lies between two loops without being itself part of a loop.

RECEIVED: December 11, 1967

REFERENCES

INGERMAN, P. Z. (1962), Algorithm 141-path matrix. *Commun. Assoc. Comp. Mach.* **5**, 556.

MARIMONT, R. B. (1959), A new method of checking the consistency of precedence matrices. *J. Assoc. Comp. Mach.* **6**, 164–171.

SCHURMANN, A. (1964), The application of graphs to the analysis of distribution of loops in a program. *Inform. Control* **7**, 275–282.

WARSHALL, S. (1962), A theorem on Boolean matrices. *J. Assoc. Comp. Mach.* **9**, 11–12.