Behavior Research Methods & Instrumentation
1975, Vol. 7 (2), 105-106

PROJECT_X: A PL/1 program for processing acyclic digraphs

JAMES D. HOLLAN
Clarkson College of Technology, Potsdam, New York 13676

and

ROBERT E. OSTEEN
University of Florida, Gainesville, Florida 32601

A PL/I program for constructing and processing acyclic digraphs is described. As an aid to exposition, an application of the program to the study of epistemological structure in human memory is discussed.

Within cognitive psychology, digraphs are increasingly being used to represent epistemological relationships and to serve as models of semantic memory organization. The recent books of Anderson and Bower (1973), Tulving and Donaldson (1972), and Norman (1970) attest to the burgeoning interest in this area. This paper presents a description of a computer program for constructing acyclic digraphs and identifying many of their structural characteristics. To facilitate the description of the program, a brief discussion of a recent application of the program is given.

Several definitions from graph theory are required for the exposition of the program and its application. A *digraph* or directed graph, D = (V, E), consists of a finite nonempty set, V, of "points," and a set, E, of ordered pairs of points termed "directed lines." The *indegree* of a point is the number of directed lines into the point, and the *outdegree* is the number of directed lines from the point. A *directed walk* from point v_0 to point v_n is a sequence $v_0, v_1, \ldots, v_n$ of points. If no point has multiple occurrences on a walk, the walk is a *directed path* from v_0 to v_n. If $v_0 = v_n$ but no intermediate point of a walk has multiple occurrences, then the walk is a *cycle*. An *acyclic* digraph is one having no cycles.

A DIGRAPH REPRESENTATION OF EPISTEMOLOGICAL STRUCTURE

Inherent in any set of concepts is an epistemological dependency structure. This structure is imposed on the concepts by the relationship of presupposition (conceptual dependency). Any concept is logically dependent upon the concepts which comprise its definition, and these concepts are, in turn, dependent upon the concepts which comprise their definitions, etc. This dependency continues back until the primitive concepts, those given no formal definition (e.g., ostensively defined concepts), are reached.

The epistemological structure imposed upon a set, S, of concepts by the relation of presupposition can be represented by a digraph in which a point of the digraph is associated with each concept of S (i.e., S = V), and a directed edge from a given point, v_i, to another point, v_j, exists just in case the concept associated with v_i is defined in terms of the concept associated with v_j. The existence of a cycle in such a digraph would indicate that the represented system of definitions included a circularity and, thus, the underlying structure of a valid system of definitions is necessarily acyclic.

In order to demonstrate the complexity of the epistemological structure imposed by the relation of presupposition on even a small set of concepts, 54 concepts from an introductory chapter of *Graph Theory* (Harary, 1969, Chap. 2) were chosen, and the epistemological structure imposed by their definitions was constructed. To concretize the complexity of the structure, it is sufficient to consider that there are 1,795 distinct directed paths from the nonprimitive concepts to the primitive (undefined) concepts, ranging in length from 1 to 10 directed lines. It is incontestable that any extensive investigation of the epistemological structure of a nontrivial set of concepts requires the use of a computer.

For a detailed discussion of some experimental investigations of the psychological import of epistemological structure, the reader is referred to Hollan (1974).

THE PROGRAM

PROJECT_X is a PL/1 program which effects the

Requests for reprints should be sent to James D. Hollan, Assistant Professor of Psychology, Clarkson College of Technology, Potsdam, New York 13676. This paper was presented at the National Conference on the Use of On-Line Computers in Psychology, Boston, Massachusetts, November 20, 1974.

construction of acyclic digraphs and generates information concerning certain of their graph-theoretical characteristics. The program represents the digraph by means of BASED structures and OFFSET locator variables within a BASED AREA. If, during the construction of a digraph, the area ceases to be sufficiently large, the program automatically allocates a larger one and continues. At the conclusion of processing, the program preserves the constructed digraph representation by writing the area onto an output device (e.g., a card punch). On a subsequent run, the program first inputs the saved area and then continues the digraph construction from wherever it had left off.

A unit of input to the program consists of the name of a point, the list of the names of the points to which there are directed lines from the given point, and any descriptive data (a character string) associated with the given point. In the application discussed above, an input would consist of the word denoting a concept, a list of the words denoting the key concepts from its definition, and a statement of its definition.

As illustration, let C, X, Y, and Z represent a word being defined and the three key terms from its definition. To facilitate the description, C will be termed a *successor* of each of X, Y, and Z, each of which will be called a *predecessor* of C. The program assimilates this input into the digraph representation in the following form. Each of C, X, Y, and Z has an entry on the alphabetical list of names of points. C has an appearance (in the form of a locator variable pointing to its entry in the "dictionary") on the successor list of each of X, Y, and Z; the predecessor list of C consists of the locators of X, Y, and Z.

It is emphasized that the result is obtained *regardless* of the situation prior to the assimilation of the definition. For example, if X is not found in the dictionary, an entry for it is created with a null predecessor list and a successor list consisting of only C; if X is found in the dictionary, then C is merely added to its (possibly null) successor list. If no dictionary entry is found for C, then one is created with a null successor list and the predecessor list: X, Y, Z. There may, however, already be an entry for C for either of two reasons: C has occurred in the definitions of some other term, or C has been previously defined. In either case, the successor list for C is unchanged. In the latter case, C is removed from the successor list of each element on its predecessor list and the predecessor list for C is nullified. From this point, the processing is just as when there had been no entry for C. Thus, if for any reason one wishes to change a definition of a term already represented, he merely submits an input for the desired new definition; the program detects the situation and performs all the required modifications.

The program automatically preserves acyclicity. As each directed line is considered, e.g., (C, X), a test is made for the existence of a directed path from X to C. If such a path is found, line (C, X) is not accepted. Instead, the program issues a notice of rejection and a list of all the cycles which would have resulted from the acceptance of the rejected line is printed.

The program also provides the user with the option of automatically performing the transitive reduction of the digraph. If this option is chosen, each directed line paralleled by a directed path of length greater than one is removed. For example, if C is defined in terms of X, Y, and Z, and X is defined in terms of Y, then line (C, Y) is deleted.

The printed output from the program, continuing with the terminology of the application, includes the following: (a) all the paths from nonprimitive concepts to primitive concepts; (b) a sequentially numbered alphabetical list of the concepts; this list includes each concept's predecessors and successors, its indegree and outdegree, its definition, and the list of all paths from it to primitive concepts with the length of the paths and their mean; and (c) the paths between every pair of concepts connected by a path with the number of such paths and their lengths.

The program has been compiled and tested using the PL/1 F-level compiler on a 370/165 running under OS/MVT/HASP.

REFERENCES

ANDERSON, J. R., & BOWER, G. H. *Human associative memory*. Washington: Winston, 1973.

HARARY, F. *Graph theory*. Reading, Mass: Addison-Wesley, 1969.

HOLLAN, J. D. Some effects of epistemological structure on memory. *Memory & Cognition*, 1974, **2**, 670-676.

NORMAN, D. A. (Ed.) *Models of human memory*. New York: Academic Press, 1970.

TULVING, E., & DONALDSON, W. (Eds.) *Organization of memory*. New York: Academic Press, 1972.