1971

Two comparisons of array and list-processing memories for the ALDOUS simulation of personality

Robert Allen Lewis
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Part of the Experimental Analysis of Behavior Commons, and the Psychiatry and Psychology Commons

Recommended Citation
Lewis, Robert Allen, "Two comparisons of array and list-processing memories for the ALDOUS simulation of personality" (1971). Retrospective Theses and Dissertations. 4475.
https://lib.dr.iastate.edu/rtd/4475

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
LEWIS, Robert Allen, 1943-
TWO COMPARISONS OF ARRAY AND LIST-PROCESSING MEMORIES FOR THE ALDOUS SIMULATION OF PERSONALITY.

Iowa State University, Ph.D., 1971
Psychology, experimental

University Microfilms, A XEROX Company, Ann Arbor, Michigan
Two comparisons of array and list-processing memories for the ALDOUS simulation of personality

by

Robert Allen Lewis

A Dissertation Submitted to the Graduate Faculty in Partial Fulfillment of The Requirements for the Degree of DOCTOR OF PHILOSOPHY

Major Subject: Psychology

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1971
PLEASE NOTE:

Some Pages have indistinct print. Filmed as received.

UNIVERSITY MICROFILMS
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>PROGRAM MODIFICATIONS</td>
<td>12</td>
</tr>
<tr>
<td>EXPERIMENT I</td>
<td>16</td>
</tr>
<tr>
<td>EXPERIMENT II</td>
<td>34</td>
</tr>
<tr>
<td>GENERAL DISCUSSION</td>
<td>39</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>43</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>48</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>49</td>
</tr>
<tr>
<td>APPENDIX B</td>
<td>50</td>
</tr>
<tr>
<td>APPENDIX C</td>
<td>63</td>
</tr>
<tr>
<td>APPENDIX D</td>
<td>71</td>
</tr>
<tr>
<td>APPENDIX E</td>
<td>86</td>
</tr>
<tr>
<td>APPENDIX F</td>
<td>102</td>
</tr>
<tr>
<td>APPENDIX G</td>
<td>111</td>
</tr>
</tbody>
</table>
INTRODUCTION

Computer simulation is particularly applicable to information-processing models of behavior, and has become increasingly important in theory development in psychology (Abelson & Carroll, 1965; Abelson & Rosenberg, 1958; Atkinson & Shiffrin, 1968; Baron, 1968; Bellman, Friend & Kurland, 1966; Colby & Gilbert, 1964; Feigenbaum, 1961; Feigenbaum & Simon, 1962; Gullahorn & Gullahorn, 1965a; Loehlin, 1962; Miller, Galanter & Pribram, 1960; Miller, Olds & Norman, 1970; Reitman, 1970; Rumelhart, Norman & Lindsay, 1971). The present study investigates issues of incompatibility among computer models of personality and also attempts to provide solutions for some of the difficulties in using computer models without extensive experience in computer science.

Definition of Simulation and Modeling

Simulation may be defined as the development and use of models to study the dynamics of existing or hypothesized systems. The key word is dynamic, because the nature of human behavior is the dynamic interaction of the organism and its environment (Simulation, 1965). Another term commonly used to describe such research is modeling. "Simulation" refers to a reproduction of a system; "modeling" describes a less perfect copy of the system (Loehlin, 1968). Modeling is presently used to refer to computer programs because this term implies less sophistication and correspondence to human behavior. In other words, no simulation
of personality has been developed. However, computer modeling promises to be a useful tool for studies of behavior.

Benefits of Computer Modeling

What are possible benefits of computer personality models? Loehlin (1968) agreed with Simon and Newell (1964) that the computer model of a process is itself a theory of the process, and he noted several ways in which a model of personality can produce theoretical benefits. The model can stimulate new ideas. A modeler can derive theories by identifying useful features of previous programs much as a clinician examines case notes as an aid to speculation. The model might then serve as a tool in examining the implications and consequences of various assumptions of a personality theory. The model can also serve as a critique of existing verbal theories which must be reduced to explicit sets of relationships to fit within a computer program. In developing such relationships, weak points of verbal theories can be clarified and strengthened. Once a theory is developed, the model serves as documentation of its postulates. Finally, modeling leads to the use of new terminology, which Loehlin maintains can clarify the expression of ideas. A potentially useful byproduct of computer models is that converting a verbal theory into modeling terms discourages hasty or ill-conceived experiments because of the number of supplementary assumptions required.

Loehlin (1968) also listed some practical benefits of com-
puter models. The computer program may serve as an archive of information about the personality being modeled. The model may also be used as a predictor of performance, the degree of accuracy depending upon the model's sophistication. An advanced model might be used as a trainer for counselors, salesmen and teachers, or in the development of testing (Newell & Simon, 1961; Thomas, 1970; Weizenbaum, 1966). It is also possible that interacting with human-like computer programs can serve as a powerful stimulus for attitude change (Dawson, 1962).

Current Problems

The problems of personality modeling appear to fall into three large categories: problems associated primarily with psychology, problems associated primarily with limitations of computers as research tools, and problems which are due to some interaction of psychological theory and computer science.

In psychology, contemporary theories display a remarkable dissimilarity in the use of constructs and operational definitions. In memory research, for example, one problem is the inability to differentiate the effects of encoding, storage and retrieval processes. Certainly there have been several partial theories and models (Atkinson & Shiffrin, 1968; Feigenbaum & Simon, 1962; Laughery, 1969; Reitman, J., 1970; Reitman, W., 1965), but there is no completely quantified theory. The memory models cannot accurately distinguish effects of information encoding from problems of retrieval because of the inability to
study stored information without retrieval. This issue may be approached by computer simulation because information in a model's memory can be dumped at any time and compared with the output (retrieval) the model produces.

The computer environment also imposes restrictions upon modeling. Obviously, computers are not human, so comparisons between computer modeling and human behavior suffer from a generalization problem (Green, 1963). This situation may be considered analogous to generalizing from animal behavior to human behavior. Perhaps the computer can be more useful to psychology than an animal model, because the microscopic operation of the computer is well known, but the internal operation of the research animal is not.

Another serious problem is that present computers are not sufficiently large or complex enough to completely represent human behavior (Loehlin, 1968). To this extent, psychologists are limited by computer hardware.

Computer languages also impose restrictions on modeling (Tocher, 1965). Part of this problem lies in the translation of theory into computer programs, and part of it lies in how the programs are coded. For example, ALDOUS was originally coded (Loehlin, 1963) in Assembler, a relatively flexible but tedious programming language usually used only by computer scientists. With the development of more advanced computers and programming languages, ALDOUS was re-coded into FORTRAN (Loehlin, 1965),
which led to certain restrictions. The memory has unrealistic stimulus combinations, but FORTRAN prohibits their deletion without great cost. Gullahorn and Gullahorn (1965a, 1965b) recognized these problems. Their HOMUNCULUS was coded in IPL-V, allowing a more flexible list memory. All personality models developed to date have had array or list memories. None have included provisions to expand and modify themselves with experience beyond extremely strict guidelines.

A final problem is that research strategies are not consistent on even the most basic level. Models have been developed independently, and each has centered on one specific problem, e.g., impression formation, neurotic behavior and interpersonal attraction. Consequently, it has been difficult to compare the merits of various models and to judge which program ideas, memory organizations for example, might be best suited to future research.

Memory Units in Personality Models

Models of personality can be categorized according to the organization of their memory units. This dimension of analysis is useful because all computer personality models include a memory sub-unit, and the way in which information is encoded and processed affects the entire model (Miller, Olds & Norman, 1970; Rumelhart, Norman & Lindsay, 1971).
The array format is used for memory units by Loehlin (1963, 1965) and Colby (1963, 1965). These memories consist of arrays which are basically cartesian products. Such an organization has several limitations. For example, Loehlin's ALDOUS has one large array memory of approximately 1,300 locations. The stimulus possibilities for ALDOUS are fixed before simulation and cannot change during the program run. In fact, because input stimuli are used as parameters to address the memory, stimulus values outside the dimensions of the array are meaningless. Thus, the memory cannot react to unique situations. Further, the organization in no way represents cognitive processes. The memory is based on computer functions, not theories about human memory. Because this memory contains all possible combinations of all stimuli, there are many situations represented which could never occur in life, e.g., a pregnant man. The program thereby wastes valuable storage and time.

<table>
<thead>
<tr>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2 Unused</td>
</tr>
</tbody>
</table>

Figure 1. Storage of items X(1,1) and X(3,2) in an array memory.
Figure 1 illustrates a 3 x 2 array memory of six locations with data stored only in locations (1,1) and (3,2). The other elements of the array are unused.

Colby (1967) attempted to program a model of a neurotic process using the array memory format. His model manipulates "beliefs" and reacts to stimuli representing the environment. The model's memory is an improvement since it relates various facts to form sets of beliefs. The memory consists of four arrays: belief matrix, dictionary, substitute matrix and reason matrix. The first three are essentially independent memories with equal-length items similar to ALDOUS' memory. The reason matrix, however, is a table of relationships linking items in the other arrays. For example, the reason matrix might link the belief "I must not hate hostile people" with another belief as a reason, "people ought to respect people." This strategy goes beyond simple array storage: the beliefs are related by reasons which themselves are stored in memory. Therefore, the reaction Colby's model makes to a stimulus is not only related to beliefs, but to the relationships among beliefs. Some limitations of the simple array memory are met by the array of interrelationships, but the size and contents of the memory are still fixed within any modeling session.

Ross (1967) outlines a list processing memory, offered as an alternative to the array memory organization. In list processing memories, the space required varies according to item
length, in contrast to the fixed-length memories of ALDOUS and Colby's model. List memories commonly include information about relationships among items so that associations and logical sequences are programmed and stored. And list processing allows the creation of elements without allocating fixed amounts of storage (Amosov, 1967). Figure 2 illustrates a list memory with the same capacity as the array memory in Figure 1. Elements (1,1) and (3,2) occupy adjacent areas, leaving the remaining storage free for other items.

Gullahorn and Gullahorn (1963, 1964, 1965a, 1965b, 1965c) used list processing for HOMUNCULUS, a model of Homans' (1958) theory of social justice. The primary goal of the model is to examine interpersonal interactions. HOMUNCULUS' memory consists of three hierarchically organized lists, two of which have as many as 12 levels of organization. Although the exact content of

Each level is not fixed before running the model, the list structure is predetermined, i.e., the model cannot create unique memory levels. HOMUNCULUS' memory hierarchies can be illus-
treated by the following levels within the "images" list: name; list identification; specification of stimulus as self, environment or group; type of group; group as a whole or some member; acting as a person or for the group; specific action; past rewards for responding; personal cost in time, money and energy; most likely next action of the other person or group; value of this action. HOMUNCULUS uses the postulates of Homans' theory to process incoming situations and select the most favorable response. The model also generalizes from one person to another. Thus HOMUNCULUS' memory can grow in volume with experience along carefully set guidelines (Gullahorn & Gullahorn, 1965c), an ability, however, not characteristic of all list memories.

Abelson's (1959, 1963, 1968) model of an attitude system is concerned with modifying a pre-set attitude system and coping with an environment through defense mechanisms. The model's memory is a series of grammatical elements (sentences and phrases), each of which is associated with a list structure. Three types of information are associated with each element: its own elements and compounds into which it enters; semantic relationships with other elements; and its value, i.e., the affective component of each element. Complex sentences are input to the model which manipulates them through various grammatical rules until they match beliefs existing in the memory. The congruity of beliefs and stimuli is then assessed through a form of Heider's balance theory (Heider, 1958). The model then
modifies its beliefs or applies defense mechanisms, depending upon the affective value of the input statement. As an example of the Abelson list memory, two hierarchical branches of the concept "living organisms" might be organized as follows:

Sparky, terrier, dog, canine, pet, living organism; Sami, Siamese, cat, feline, pet, living organism.

Abelson's model is similar to Colby's model in its use of related material to form beliefs and attitudes; however, the two models approach the problem with different memory organizations. A continuing problem with computer modeling is the inability to control enough variables in two such models to examine the relative merits of list processing or array memories exclusive of program strategies.

The Research

The present research is concerned with a comparison between the two major types of memory organization on the ALDOUS model. ALDOUS was chosen because there has been considerable research on this model. Further, a streamlined version of ALDOUS is available as a benchmark program (Lewis, 1971). Both array and list-processing memories have been used on different computer models of behavior, and each strategy has its advantages. List processing, however, is judged to be the superior design. One aim of this study is to demonstrate the advantages of a list-processing memory by direct comparison with an array memory on the same model of personality. The secondary goal of this re-
search is the development of a strategy for simplifying comparisons among models. The macro-instruction approach was used because of the ease with which persons having no background in computer science can utilize models which employ it.
PROGRAM MODIFICATIONS

The PL/I ALDOUS model was developed as a series of macro-instructions or super-ordinate commands which expand in detail automatically. For example, to study two memory formats in the same model, it should be possible to change a specification from "array" to "list" and rerun the same experiment. The macro-language approach is independently supported by Remy (1970) who found such a method useful in teaching unskilled programmers.

The first step was to examine FORTRAN ALDOUS (Lewis, 1971) which had been made compatible with local programming conventions (Thomas, 1970). A schematic flowchart of ALDOUS is presented in Appendix A. The 624-statement program consisted of a main program (the environment), 4 functions and 10 subroutines. Since all but three of these subroutines were called only once by any other routine, it was clear that substantial savings of computer time could be effected if some subroutines were appropriately inserted in the main program. The following routines were introduced into the main procedure: STIMUL, CONSEQ, and QUESTN. Routines RECOGN, EMOTN, ACTION, REACTN, LEARN and REPORT were inserted into the ALDOUS subroutine. These modifications reduced the program's running time 31 per cent. Timings were made on the IBM System/360 Model 65 at Iowa State University, Ames, Iowa. Several unused statement numbers and labels were also removed and certain data handling inefficiencies were corrected (Lewis, 1971).
ALDOUS was then recoded in PL/I and the macro-language format was employed. The program was coded in sections corresponding to the original subroutine designations (Appendices A and G). The keywords ENV1, ENV2, MEMORY, PASS and LIST were implemented. The detailed function of each keyword and its syntax are outlined in Appendix G, a user's manual for the macro-language. Complete texts and flowcharts of both the array and list programs are presented in Appendices D, E and F. The original FORTRAN program is presented in Appendices B and C.

Operationally, PL/I ALDOUS was run as the last of a three-stage program using PL/I compile-time facilities. Compile-time facilities allow a program to be modified or edited by other steps of the same program. Stage I used the IBM utility IEBGENER (IBM, 1971b) to create a partitioned data set on direct access memory. A partitioned data set is one in which various sections can be retrieved in any order by reference to their names. Inputs to this stage were all the subroutines for both ARRAY and LIST ALDOUS. These subroutines were combined in a later stage of the program. The output of Stage I was the partitioned data set which was passed to Stage III. Stage II was the translator for the macro-language which read the macro-language statements, checked them for syntax, and wrote an output data set. The output of Stage II was a series of compile-time instructions which called and assembled the various sections of ARRAY or LIST ALDOUS. In Stage III, the compile-time
instructions from Stage II were used to assemble the ALDOUS subroutines from the partitioned data set. Parameter values read by Stage II were edited into the program, and the program was executed. To summarize briefly, in Stage I the necessary components for both versions of ALDOUS were stored in the computer's direct access storage. Stage II read the macro-language instructions and wrote directions so that proper components were assembled and executed in Stage III.

A related concern of this research was the determination of which programming language might be the most suitable for such a project. Three possibilities exist: a new programming language developed through a compiler-writing program; a program written in an existing simulation language, or a program written in an existing general-purpose language (Dames, 1965). Regarding the first alternative, such a compiler writer, META PI, has been developed (Reschly, 1970; Theys, 1970). However, its current users indicate META PI has several weaknesses which would make this research primarily a computer science problem. Three simulation languages are presently available (IBM, 1970a; Knuth & McNeley, 1964; Kiviat, Villanueva & Markowitz, 1968), but they tend to be specialized in function and require sophistication in computer science. The General Purpose Simulation System (GPSS) developed by IBM (1970a) and SOL (Knuth & McNeley, 1964), its higher-level relative, are based on queuing theory and operate in essentially the same fashion. For the most part, stimuli are
generated at prescribed time intervals and are considered identical. GPSS and SOL do not allow for highly specialized reactions based upon characteristics of individual stimuli and therefore are not acceptable as personality simulation languages since a personality model's behavior can be highly related to the stimuli it encounters. SIMSCRIPT (Kiviat et al., 1968) is a general-purpose simulation language which allows more flexibility than SOL and GPSS, but cannot support list memories.

For these reasons, then, the first two alternatives were rejected and a solution was sought in PL/I, an existing general-purpose language. PL/I is relatively new and, unlike most languages, it allows both array and list memories. Therefore, PL/I meets the technical requirements for this project, and was used throughout.
EXPERIMENT I

In Experiment I, ALDOUS was used to replicate one of Loehlin's best-known experiments (Loehlin, 1963) in which the model was run through two different environments, hostile and benign. Loehlin found that when the normal model passed from hostile to benign environments, more trials were required to reach an appropriate response in the second environment than when the change was from benign to hostile. The appropriate response to a hostile environment was withdrawal, while the benign environment encouraged approach behavior. A recent replication (Lewis, 1971) with a streamlined version of ALDOUS supported this finding. ALDOUS was set for "normal" parameters: the average values of 26 key program variables which Loehlin described as producing the most stable and logical reaction to a wide variety of environment.

It was hypothesized that when the original hostile and benign stimuli were presented in the mix described by Loehlin, no difference in performance would occur between the array and list memories. Both models had identical inputs which could be entirely stored and utilized by their respective memory units (Figures 1 and 2). However, if the stimulus identification fields were enriched with addresses signifying stimuli outside the range of the array memory, i.e., less or greater than 000 through 999, such enriched information would be lost or useless to ARWAY ALDOUS. That is, the enriched information would func-
tion as filler trials which delayed the effects of useful information. LIST ALDOUS could process enriched information since its memory developed as stimuli were presented. If enriched addresses were not repeated, such information would be stored by LIST ALDOUS, but not used in attitude formation. Non-repeated enriched stimuli would therefore be as useless to LIST ALDOUS as to ARRAY ALDOUS, and so the models would respond the same in this treatment condition. However, if the identification values were repeated, LIST ALDOUS would form attitudes toward these new identities as easily as any other stimuli and its response to the environment change would be more rapid than ARRAY ALDOUS for the same stimuli.

Method

Experimental Stimuli.—Each stimulus consisted of seven numbers: a three-digit identification describing a memory address, and a four-digit segment describing the degree of "press" (Loehlin, 1965) or effect the stimulus could have on ALDOUS. The identification field was limited to three one-digit numbers and therefore could only access the 1,000 addresses 000 to 999. A hostile environment was one which could produce discomfort and frustration, while a benign environment was one in which ALDOUS was free to act without harmful reactions. Figure 3 illustrates typical hostile and benign stimuli. A trial was defined as one stimulus presentation and the corresponding response, if any, and each trial was one unit of time.
Figure 3 illustrates a pair of hostile and benign stimuli for memory location 3 3 3. The press, injury, frustration and satisfaction values are set on a low to high scale of one to nine. In this example, the overall press of the hostile stimulus is seven with greater values of injury and frustration than satisfaction. The benign stimulus, by contrast, has a press of three with satisfaction outweighing injury and frustration.

Enriched stimuli were generated by enlarging the identification field of each stimulus to include two-digit numbers. Three sets of stimuli were developed: 500 hostile and 500 benign non-enriched stimuli; 500 hostile and 500 benign enriched stimuli with non-repeated identifications; and 500 hostile and 500 benign stimuli with enriched repeated identifications. Hostile and benign stimuli were generated by manipulating the affective portions of each stimulus (the four press values). For hostile stimuli, the values of "press," "injury" and "frustration" were set high on the one to nine scale. The hos-
tile environments generated in this study contained the following stimulus patterns: I I I 7 7 5 1, or I I I 7 5 7 1, where I I I represents the identification field of each stimulus. The affective values of injury and frustration were not set equal in any one trial since this condition led to a conflict reaction by ALDOUS. The benign stimuli were characterized by a low press value and low values of injury and frustration coupled with a moderate value for satisfaction. The benign stimuli used in this study had the following forms: I I I 3 1 3 7, and I I I 3 3 1 7. In the non-enriched stimuli, five memory addresses were each repeated 100 times to generate the 500 trials. The identification fields, which are used to address the model's memory, were given the following values: 3 3 3, 3 3 5, 3 5 3, 5 5 3, and 5 5 5. Repetition of these memory addresses built experiences with the affective portion of each stimulus. In the non-repeated condition, 20 per cent of the identifications were replaced with non-repeating three-value numbers in the range 12 through 20, and in the repeated condition, 20 per cent of the identifications were replaced by 100 replications of one enriched identification (12 12 12).

Procedure.—Each of the ALDOUS models, ARRAY and LIST, was run through the three treatment conditions in both the hostile-benign and benign-hostile directions for a total of six runs with each model. After 500 trials in one environment, the models were switched to the other environment without warning.
Parameters were set to "normal" for both models. At the end of each run, the memories were dumped and compared for attitude formation following the environmental change.

Output of the ALDOUS model was coded on a 10-point scale given in Appendix G. This scale did not have a logical affective order, i.e., the indifference value was at one end and paralysis at the other with approach and avoidance near each other in the middle. To make the output more understandable, the data were recoded as follows:

<table>
<thead>
<tr>
<th>Old Value</th>
<th>New Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>9</td>
<td>Strong Approach</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>Weak Approach</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>Strong Attraction</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>Weak Attraction</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>Indifference</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Weak Conflict</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Strong Conflict</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>Weak Avoidance</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>Strong Avoidance</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Paralysis</td>
</tr>
</tbody>
</table>

Results

Figure 4 illustrates the results of non-repeated trials in both hostile-benign and benign-hostile directions for array and list models. Data are presented as means of 10-item blocks beginning 50 trials (5 blocks) before the change in environment, and continuing through the 500 trials (50 blocks) of the second environment. The lines representing responses by the array and list models perfectly overlie each other.

Since the data were generated with a known random error (6
per cent), the standard error of the mean of 10 data points could be expressed as the square root of \((pq/n)\) where this expression is evaluated as square root \((.06*.94)/10\) = 0.0751. Therefore, the data in Figure 4 could be considered statistically correct within ± 3 standard deviations or ± 0.2253 with \(p < .01\). The treatment conditions were separated more than this amount, so no further statistical analyses were conducted to determine the difference between treatments, or the difference between response levels before and after a change in environment. In order to evaluate the prediction that the hostile to benign change will be slower (take more trials) than the benign to hostile change, a criterion for stable adjustment was established as two consecutive blocks (20 trials) with the same mean response.

Figure 5 illustrates the array and list models' responses in the enriched, non-repeated treatment condition. As in Figure 4, data were presented as the means of 10-trial blocks beginning 50 trials before the environment change. Again, the responses of the array and list models overlie each other. Figure 6 presents responses to the enriched repeated treatment condition for both models. The same statistical assumptions and criteria applied to Figure 4 were used to evaluate Figures 5 and 6.
Figure 4. Reactions of the array and list models in the non-enriched environment.
Discussion

Figure 4 provides conclusive evidence of no difference between array and list memory models in the non-enriched condition with stimuli within the range of the array model. Both models display a steady avoidance in the hostile environment preceding the change to a benign environment. The sharp deviation immediately preceding the benign environment was caused by outlying points of strong approach generated by random misrecognition. After two encounters with the hostile environment, the models returned to more predictable avoidance behavior. During the first 20 trials in the benign environment, the models exhibited approach behavior which was abandoned by the end of 100 trials for a stable indifference. The intermittent deviations from indifference were due to random error and to the models' attempts to approach the environment. From time to time, the models deviated from indifference for considerable periods, especially from blocks 32 to 35. Loehlin suggested such periods of attraction to a benign environment were to be expected as the impact of repeated benign trials decreased. One random misrecognition could encourage a series of approach or avoidance trials which soon decayed for lack of reinforcement. Therefore, the model reverted to indifference which became more stable over time.

The benign to hostile models completed their benign environment on a weak approach level and moved immediately toward
Figure 5. Reactions of array and list models to enriched, non-repeated environment.
avoidance after the first hostile presentations. Response in
the hostile environment, while consistent between models, never
reached the flat profile of the benign environment. In general
a steady decay was evident until block 40 when responses finally
stabilized between strong and weak avoidance. Response vari­
bility in the hostile environment could be attributed to the
fact the stimuli had two locations to manipulate hostility
(injury and frustration) and only one location to manipulate
satisfaction. Therefore, when both hostility variables were ma­
nipulated, the models were delayed in accumulating enough exper­
ience with each stimulus value to stabilize the environment.
This observation is supported in the benign stimuli where only
"satisfaction" had significant values, and the acquisition of a
stable response pattern was clear by data block 10.

The criterion of consecutive blocks with equal means showed
that as predicted, the hostile environment reached a plateau
sooner (blocks 7 and 8) than the benign environment (blocks 10
and 11). This finding supports the prediction of faster change
upon entering a hostile environment, but the hostile responses
continued to decline after this first plateau.

As can be seen in Figure 5, the enriched non-repeated
trials led to findings similar to the non-enriched condition.
Both array and list models again produced identical outputs, as
predicted. However, there are some important differences be­
tween the enriched non-repeated and non-enriched responses. As
in the non-enriched case, models entering the benign environment underwent an immediate approach period followed by indifference. In the non-repeated case, the stable indifference found in Figure 4 did not appear, even though the tendency toward indifference was evident. The prediction that a stable response pattern would emerge in the benign environment was not strongly supported. The pattern fluctuated between indifference and a mean composed of indifference and approach probably as a result of three factors. The random misrecognition sometimes started a temporary trend which declined after non-reinforcement. The model had a built-in tendency to approach ambiguous non-repeated situations. That is, the model exhibited "curiosity" and approached 20 per cent of its environment even though the remainder was benign. Finally, the model may have reacted to the new non-repeated environment as one which required its own adjustment. These possible approach strategies would be encouraged by an environment which allows freedom of approach. The final five data blocks in the preceding hostile environment also exhibited greater variability than in Figure 4, further suggesting that non-repeated stimuli encouraged approach responses.

Models entering the hostile environment from the benign followed a brief approach phase, then changed within 200 trials to strong and consistent avoidance responses. A period of variability ensued before the responses settled into a weak
avoidance-strong conflict situation by trial block 50. This pattern, predictable avoidance followed by variability, is again probably due to the nature of the hostile stimuli coupled with ALDOUS’ tendency to approach ambiguous stimuli. Because each stimulus was composed of two hostile values and one friendly value, there was more information to form hostile impressions and make a more rapid adjustment than in the benign condition. When non-repeated trials failed consistently to reinforce avoidance behavior, ALDOUS began exploratory approach behavior. These approach trials were not reinforced, and the model exhibited periodic swings from approach to conflict and avoidance.

The prediction that non-repeated trials would serve to delay a stable environment was not clearly supported. Models in the benign environment reached the criterion only one block later than in the non-enriched case. Models in the hostile environment failed to meet the criterion at all. This finding is due to response variability which was greatly increased by this treatment. Although these results were replicated by FORTRAN ALDOUS, the discussion of latency criteria must be in terms of response trends, rather than a criterion.

In contrast to Figure 4, Figure 5 shows more variability and a tendency toward less extreme responses in either environment. The benign environment produced an almost cyclic rise toward approach while retaining the indifference response as a lower bound. Therefore, this indifference-attraction combina-
tion might be considered a steady attitude state, even though it differs from the clear, stable indifference in the non-enriched condition. The hostile responses, after the first 200 trials, show a similar tendency which may suggest some pattern based on the intricacies of the model.

Predictions about the list and array models' performance in the repeated enriched condition are challenged by the results presented in Figure 6. The benign models show the pattern set forth in Figure 5: initial approach followed by variation from indifference to attraction and approach. This finding supports the prediction that the array model will behave the same in enriched, non-repeated and repeated conditions (Figures 5 and 6). However, the list model also follows the same pattern found in the non-repeated condition, as opposed to a prediction of faster adjustment utilizing more information. These parallel findings may be due to differential effects of the repeated stimuli leading to the same responses. For the array model, enriched stimuli were filler trials which delayed a stable response. The list model recognized a greater number of unique identifications than in the non-enriched situation. In the non-enriched case, each model coped with 5 unique identifications. In the non-repeated condition, the number of unique identifications was 105 (100 non-repeated identifications, i.e., 20 per cent of 500 trials, and the 5 original patterns). In the enriched repeated condition, each model received the 5 original patterns, plus the
repeated pattern. While the array model again ignored enriched trials and reacted as in the non-repeated case, the list memory should have recognized these trials and adjusted faster. It is difficult to believe the effect of 100 filler trials on the array model is identical to adjustment to new repeated stimuli for the list model. Further investigation of stimulus patterns and model responses should be conducted to determine the effects of stimulus repetition. No such data are available from Loehlin.

In the benign to hostile condition, the repeated enriched stimuli produced a difference between list and array responses. The array model behaved more stably than in the non-repeated condition, even though the amount of information available in each condition was theoretically the same. The list model showed greater variability while following the same general trend as the array model. This finding may be explained by the list model's attempt to use a greater amount of available information, thereby exhibiting a pattern similar to the non-repeated condition. This result may also be due to conflicting values of frustration and injury within the stimuli which retard or inhibit a stable adjustment. This explanation is not supported by response data from benign environments which indicate that when only one affective value (satisfaction) is manipulated, the same variability appears in the responses.
Figure 6. Reactions of array and list models to enriched repeated environment.
As a further investigation of response variability in enriched environments, the hostile enriched repeated environment was recreated with 50 and 75 percent enrichment. Results from these trials are presented in Figure 7. In the 50 per cent case, the models followed essentially the same pattern exhibited in the 20 per cent condition except the results are more stable. The two models reached nearly identical adjustments for what must be entirely different reasons if the original research is to believed. The array model has only 250 useful trials, and its responses compare favorably with the 20 per cent enriched condition. This result is not expected since there should be less useful information available to the model in the 50 per cent condition, and therefore more variability in its responses. The list model, on the other hand, using the same stimuli for 500 trials, reached the same level of adjustment. Since the array model could have been expected to exhibit more variable responses based on scantier information, these results are particularly surprising and suggest ALDOUS is not as responsive to changing input stimuli on a trial-to-trial basis as Loehlin reports.

In the 75 per cent case, the models again exhibited nearly identical behavior, but the nature of the responses changed dramatically. At first glance, the output appears to be that in a benign environment since both models center their responses near indifference. This finding is not as unexpected for the 75
Figure 7. Reaction of array and list models to enriched repeated environment with 50 and 75 per cent repetition of stimulus identifications.
per cent condition in the array model since it now had only 125 useful trials, and therefore could be expected to not get reinforcement for approach or avoidance behavior on a regular basis. But the list model was expected to form definite attitudes since 75 per cent of its trials were concentrated on one stimulus identification. The 75 per cent list model should have formed extreme avoidance and conflict reactions if all the information at its disposal were utilized according to previously published literature.

Experiment I supports some of the hypotheses about ALDOUS. The prediction of more rapid adjustment in a hostile environment was supported by a criterion test where the criterion could be established, and by trends in other findings. The identical responses of the array and list models in all except one treatment condition corresponded to the original program's response to the same stimuli. This finding indicates the recoding of ALDOUS into PL/I was successful. The puzzling findings in the enriched repeated stimulus conditions do not support the present hypotheses. They suggest interactions of stimulus structure and model complexity which have not been investigated, but which might be explored with the list model.
EXPERIMENT II

Experiment II involved modifying a "neurotic" ALDOUS to "normal" through counseling (Loehlin, 1965). The neurotic condition involved setting the basic parameters so ALDOUS was over-reactive to threat and generally indecisive. The model was first run through a training session in which the values in its memory locations were modified towards negative reaction tendencies by exposure to 500 stimuli with the following formats: 111997 1, 1119791 and 1119991. The extremely high frustration and injury values were designed to produce negative experience for all stimuli within the range of the experiment and to raise several conflict situations where the injury and frustration values were equal and high. The counseling session was run in a "friendly" environment which made positive responses to every action of the neurotic model. Loehlin found the model could not achieve long-term adjustment to a normal test environment without prolonged exposure to friendly information. Short-term exposure led to only temporary gains.

In Experiment II, the rate of change or convergence toward a normal personality was examined using different memory structures. In the case where stimuli were within the scope of ARRAY ALDOUS, it was predicted that both ARRAY and LIST ALDOUS would react identically to the same normal test environment after the same number of friendly trials. Non-repeated enriched information was expected to retard both models' adjustments as in Ex-
periment I. Repeated enriched information, however, was expected to affect the models differentially. The repeated enriched information was expected merely to delay a normal personality for ARRAY ALDOUS. But because LIST ALDOUS could utilize the repeated information, LIST ALDOUS should make its adjustment more rapidly than ARRAY ALDOUS using enriched repeated identifications.

Method

Experimental Stimuli.--In Experiment II, six stimulus sets were generated. Enriched stimuli were defined as in Experiment I. One thousand friendly counseling trials were generated in each of the conditions: normal; enriched, non-repeated; and enriched, repeated. All friendly stimuli had the format, I I I 7 1 1 9. The identification values were manipulated as described in Experiment I. Three test conditions with a sampling of press values were generated for the three treatment conditions in sets of 500 trials. The test environment was composed of affective values for press, injury, frustration and satisfaction which were equal-sized samples of the hostile, benign and friendly environments described above.

Procedure.--ARRAY and LIST ALDOUS were run through all nine possible combinations of friendly and test environments. The number of trials in the counseling environment ranged in 200-trial steps from 200 to 1,000 while the number of trials in the training and test environments was always 500. Typical stimulus patterns in any treatment condition were as follows:
500 training, 1000 friendly, 500 test; 500 training, 800 friendly, 500 test; 500 training, 600 friendly, 500 test; 500 training, 400 friendly, 500 test; and 500 training, 200 friendly, 500 test. As in Experiment I, the memories were dumped and compared after each run.

Results and Discussion

Experiment II was designed to demonstrate the difference between list and array memories when a "neurotic" ALDOUS was counseled by a friendly environment. Data were collected in the non-enriched, enriched non-repeated, and enriched repeated conditions on responses to 500 normal trials after varying amounts of counseling (200 to 1,000 trials). The results of these runs are summarized in Figure 8.

There were no differences in the test environment regardless of the amount of friendly counseling. This finding is the more extreme in that no trial or block differed for any treatment from the other results within that condition. Moreover, the only responses recorded were the most extreme in each category.

The cyclic response variability is due in part to the fact that the neurotic model was set to make strong reactions to every stimulus. However, different sets of parameters were tried. They were as follows: the most extreme value for each parameter, the average of the extreme and normal values for each parameters and the set of parameters which were used to generate
Figure 8. Example of cyclic response pattern of array and list model to all levels of friendly counseling.
the data in this experiment. These values are the following: .06, 4.00, .90, .90, .90, 1.00, 1.00, .90, .90, .90, .90, .90, .90, .90, .90, .90, .10, .10, .10, .10, .10, .10, .10, .10. An explanation of the parameter values is given in Figure 10. None produced differing responses to any number of friendly trials.

ALDOUS' parameters were set near their most negative values. Even in these settings, approach and attraction accounted for 60 per cent of the models' behavior. The only behaviors not exhibited during this run were indifference and paralysis. Paralysis could not be induced with any stimuli except those of the form IDENT1, IDENT2, IDENT3, 9, 9, 9, 1 (the highest conflicting values of frustration and injury or some combination of these high values).

Loehlin has not published the exact parameter values and methods used to generate the original experiment; it is therefore impossible to determine if the present study precisely replicated the original. The results of Experiment I, in which the original findings were replicated, suggest that the parameters and stimuli were set close to the original specifications, but that the model is generally unresponsive to this type of testing—contrary to Loehlin's reports.
GENERAL DISCUSSION

In assessing the success of the present investigations, we must examine the criteria evaluated and the qualitative issues which the research has raised, particularly the question of validating the model's performance.

Regarding the criterion tests, the project was a qualified success. In Experiment I, the original results were replicated with the PL/I model. The PL/I model may be used instead of the original model in the classic, two-environment experiment. The failure to obtain predicted results in the enriched repeated condition suggests that the interaction of the routines for combining data in impression formations and for accumulating experience do not interact within the computer framework in the straightforward way that the linear model implies. Furthermore, the increased variability in the hostile conditions, especially in the list model's responses to the enriched repeated condition, suggest that the difference between hostile and benign environments is not as simple as has been supposed. Consideration of the internal design which includes two hostile values for every friendly one, suggests that either only one hostile variable should be used at a time, or that the distribution of stimuli should be controlled to equate the incidence of hostile and friendly affect within an environment. These issues should be explored in future research.
The failure to replicate Loehlin's original experiments raises the question of whether the recoded PL/I model is indeed an authentic reproduction of the original program. PL/I ALDOUS must be nearly if not perfectly copied from FORTRAN ALDOUS since their responses to identical stimuli and experiments are the same. This correspondence between models is particularly apparent in the two-environment experiment, where the original results were reproduced by both models.

In retrospect, Loehlin (1963, 1968) diverted the attention of readers from concrete examples of his stimulus materials and the parameters used in his origin research. But this information is needed to judge the model's utility in psychological research. The failure of Experiment II suggests that before more replications or attempts to use ALDOUS in human research are undertaken, Loehlin must be prevailed upon to publish his results in complete form.

The macro-language programming of ALDOUS makes it potentially available to a wider range of users. However, the question of the internal consistency of the model again raises the issue of whether the model is a worthy candidate for further research. The ultimate test of ALDOUS' utility is whether it can be modified to correspond to human behavior. This process of establishing correspondence is the essence of modeling. Loehlin (1968) conceded the model does not possess such correspondence, nor was it based on research findings. In order to
make it useful, such issues as the values and nature of the parameters processed by the model, and the value and range on input stimuli need to be explored. For example, Loehlin assumed a six per cent random misrecognition of stimuli which then modified one affective stimulus. The question of whether this is a realistic value for most human samples needs to be studied. Perhaps existing perception data could provide a better estimate of the actual percentage of misrecognition of stimuli. The nature of the misrecognition and its effects might be examined through the same sources. Perhaps failure to recognize or a tendency to dismiss real dangers is a perceptual defense (a topic which Loehlin did not even consider in his random misrecognition formulation). In other words, misrecognition has nothing to do with ALDOUS' stimuli, even though we believe there is a relationship between misrecognition and stimulus effect in human behavior.

Another topic for ALDOUS research is the quantitative value placed on experience. The model assigns a uniform set of weights to experience in its memory, but these weights are not based on relevant research, even though considerable literature on impression formation is available.

Examples of research topics cited above have centered around changing ALDOUS to reflect research findings in order to use the model as an archive. The model might also assume a less passive role as a predictor of behavior and thus as a heuristic
to stimulate research (Miller et al., 1970). No experiment conducted with ALDOUS to date has suggested the present model can be a useful predictor. Perhaps the optimal strategy is to retire ALDOUS and begin again. Loehlin and other early modelers began with only personal theories. One might instead begin with a definite theory and include only those variables which can be quantified with some certainty. Then unanswered questions and missing relationships among variables will dictate the research required with human Ss to complete the model. In this way, the model can direct research as well as be modified by research findings.
REFERENCES


Gullahorn, J. T., & Gullahorn, J. E. Some computer applications in social science. *American Sociological Review*, 1965, 30, 353-365. (b)

Gullahorn, J. T., & Gullahorn, J. E. Computer simulation of
role conflict resolution. System Development Corporation, Santa Monica, Calif. SP-2261/000/00. November, 1965. (c)


In my opinion. Editor's comment, Simulation, 1965, 5, 381.


International Business Machines Corporation. IBM system/360 operating system: FORTRAN IV language. (9th ed.) White Plains, N.Y.: Author, 1971 (a)


Miller, M., Olds, R., & Norman, D. A. Computer structure for ELINOR. Unpublished manuscript, University of California, San Diego, California, 1970.


ACKNOWLEDGMENTS

I am pleased to acknowledge Dr. James A. Walsh, chairman of
my committee, whose enthusiasm for computer modeling stimulated
my own interests in developing this project.

I would like to express appreciation to my committee mem-
bers, Dr. Wayne H. Bartz, Dr. Harold Dilts, Dr. David C.
Edwards, Dr. Arnold S. Kahn and Dr. Clair G. Maple, for their
interest and cooperation.

Thanks are also due to Dr. Donald A. Norman, who clarified
some particular problems in computer modeling and offered
encouragement in developing my research.

I must thank my wife Mickey for her very helpful
suggestions, and for her love and encouragement.
APPENDIX A

Schematic Flowchart for the ALDOUS Model of Personality
APPENDIX B

Program Text of Streamlined FORTRAN ALDOUS

C MAIN ROUTINE--ENVIRONMENT
C CONTROLS ALDOUS MODEL
C
C INPUT:
C RUN TITLE--ONE CARD, FORMAT A(80)
C CHANGE NOTE--ONE CARD, FORMAT A(80)
C PRINTED AT ENVIRONMENT CHANGE.
C IF NO SECOND ENVIRONMENT, LEAVE BLANK.
C RUN PARAMETERS--ONE CARD, FORMAT 3 I(10)
C M1ST--TRIALS IN FIRST ENVIRONMENT
C M2ND--TRIALS IN SECOND ENVIRONMENT
C LREPT--1=ACTION PRINTED EACH TRIAL
C 2=ACTION PLUS REPORT
C 0=SUMMARY ONLY PRINTED
C ALDOUS PARAMETERS--FORMAT 26 F(3,2), 3 F(3,2)
C FIRST CARD IS PARAMS(26), SECOND IS POWERK(3)
C INPUT STIMULI--FORMAT 35 I(2)

1 FORMAT(80H RUN TITLE
2 FORMAT(80H CHANGE NOTE
3 FORMAT(3I10)
4 FORMAT(26F3.2/3F3.2)
5 FORMAT(/4X,7HCYCLES=,110,4X,10I8)
6 FORMAT(4X,17HCUM. PROPORTIONS=,4X,10F8.2/
7 FORMAT(25X,8H INDIFF,8H W CONFL,8H S CONFL,8H W APPR,
8H S APPR,8H W AVOID,8H S AVOID,8H W ATTK,8H S ATTK,
28H PARALYS)
8 FORMAT(/)
9 FORMAT(4X,50I2)
10 FORMAT(14,7F3.0,2I3,17I3,3I10)
C READ TITLES, WRITE RUN TITLE
READ(5,1)
WRITE(6,1)
READ(5,2)
C READ RUN AND ALDOUS PARAMETERS
READ(5,3) M1ST, M2ND, LREPT
READ(5,4) (PARAMS(I), I=1,26), (POWERK(I), I=1,3)
C INITIALIZE
DO 12 I=1,11
DO 12 J=1,11
DO 12 K=1,11
12 MPERM(I,J,K)=0
DO 13 I=1,7
13 TEMPM(I)=0
IDEGR=0
IACTN=0
NSTOP=0

MSTOP=M1ST
DO 15 I=1,10
NRESP(I)=0
15 NRESP(I)=0
HTALLY=0
C LOAD INPUT
16 CONTINUE
READ (5,11) {(STIMU(I,J),J=1,500),I=1,7)
11 FORMAT (35F2.0)
IF (ITIM.EQ.2) GO TO 1003
1003 CONTINUE
ITIM=2
C RUN 1000 CYCLES OF ALDOUS
DO 25 J=1,1000
TEMPM(I)=0
CALL ALDOUS (MPERM,TEMPM,PARAMS,STIMU(1,J),STIMU(2,J),
1STIMU(3,J),IDEGR,IACTN,IR,NDIM1,NDIM2,NDIM3)
IF (LREPT.LT.I) GO TO 19
IF (IACTN.GE.5) GO TO 17
WRITE (6,8)
WRITE (6,10) J, (STIMU(I,J),I=1,7), IDEGR, IACTN,
1(IR(K),K=1,17),NDIM1,NDIM2,NDIM3
GO TO 18
17 WRITE (6,8)
WRITE (6,10) J, (STIMU(I,J),I=1,7), IDEGR, IACTN,
1(IR(K),K=1,17),NDIM1,NDIM2,NDIM3
18 IF (LREPT.EQ.2) GO TO 19
TEMPM(I)=2
CALL ALDOUS (MPERM,TEMPM,PARAMS,STIMU(1,J),STIMU(2,J),
1STIMU(3,J),IDEGR,IACTN,IR,NDIM1,NDIM2,NDIM3)
C ENCODE ACTION 0-9, AND TALLY
19 IF (IACTN.NE.5) GO TO 20
ICODE=1
GO TO 23
20 IF (IACTN.NE.6) GO TO 21
ICODE=10
GO TO 23
21 IF (IACTN.NE.4) GO TO 22
ICODE=IDEGR+1
GO TO 23
22 ICODE=2*IACTN+IDEGR+1
23 NRESP(ICODE)=NRESP(ICODE)+1
LRESP(J)=ICODE
C DETERMINE IF CONSEQUENCES WILL ENSUE
POWER=STIMU(4,J)
IF (IACTN.EQ.1) GO TO 104
C --IF APPROACH, CONSEQUENCES
IF (IACTN.NE.4 .AND. IACTN.NE.5) GO TO 103
IF (POWER.GE.POWERK(1)) GO TO 104
C INDIFFERENCE OR CONFLICT
103 IF (IACTN.NE.2 .AND. IACTN.NE.3) GO TO 105
IF ((IDEGR.NE.1 .OR. POWER.LT.POWERK(2)) .AND. (IDEGR.NE.
12 .OR. POWER.LT.POWERK(3)) GO TO 105
C AVOIDANCE OR ATTACK
104 NCONSQ=1
GO TO 106
105 NCONSQ=0
106 IF (NCONSQ.NE.1) GO TO 24
C REACTION TO CONSEQUENCES
TEMPM(1)=1
CALL ALDOUS (MPERM,TEMPM,PARAMS,STIMU(5,J),STIMU(6,J),
1STIMU(7,J),IDEGR,IACTN,IR,NDIM1,NDIM2,NDIM3)
C TALLY TRIAL AND MAKE STOP TESTS
24 MTALLY=MTALLY+1
IF (MTALLY.EQ.MSTOP) GO TO 28
25 CONTINUE
C PRINT RESPONSE TALLY
MKEY=1
26 WRITE (6,8)
WRITE (6,9) (LRESP(I),I=1,J)
WRITE (6,8)
WRITE (6,7)
WRITE (6,5) MTALLY,(NRESP(I),I=1,10)
C COMPUTE CUMULATIVE PROPORTIONS
DO 27 I=1,10
NRESP**(I) =NRESP**(I)+NRESP(I)
RESP**(I)=100*NRESP**(I)/MTALLY
RESP**(I)=RESP**(I)/100.
27 NRESP(I)=0
WRITE (6,6) RESP
C QUESTIONNAIRE
C COMPUTES AVERAGE LEVELS OF FEAR, ANGER AND ATTRACTION
C FOR A SAMPLE OF ATTITUDES.
C EVERY 19TH, 7TH, AND 3RD FOR CONCRETE, 1ST AND
C 2ND LEVEL OF ABSTRACTION, RESPECTIVELY.
C IF NO ATTITUDE, IGNORED.
110 FORMAT(4X,4HEXP=',I3,4X,9HAV. FEAR=',F4.2,2X,
110HAV. ANGER=',F4.2,12X,15HAV. ATTRACTION=',F4.2/)
MT=0
MIDEN=1
DO 112 I=1,10
DO 112 J=1,10
DO 112 K=1,10
MT=MT+1
111 CONTINUE
112 CONTINUE

C EVERY 7TH FIRST LEVEL ABSTRACT ATTITUDE
MT=0
MIDEN=2
I=11
DO 114 J=1,10
DO 114 K=1,10
MT=MT+1
IF (MT-7) 114,125,125
113 CONTINUE
114 CONTINUE

J=11
DO 116 I=1,10
DO 116 K=1,10
MT=MT+1
IF (MT-7) 116,125,125
115 CONTINUE
116 CONTINUE

K=11
DO 118 I=1,10
DO 118 J=1,10
MT=MT+1
IF (MT-7) 118,125,125
117 CONTINUE
118 CONTINUE

C EVERY 3RD SECOND LEVEL ABSTRACT ATTITUDE
MT=0
MIDEN=5
I=11
J=11
DO 120 K=1,10
MT=MT+1
IF (MT-3) 120,125,125
119 CONTINUE
120 CONTINUE

MIDEN=6
I=11
K=11
DO 122 J=1,10
MT = MT + 1
IF (MT-3) 122, 125, 125
121 CONTINUE
122 CONTINUE
MIDEN = 7
J = 11
K = 11
DO 124 I = 1, 10
MT = MT + 1
IF (MT-3) 124, 125, 125
123 CONTINUE
124 CONTINUE

\[ C = \text{MTAL} \times 100 \]
\[ Q\text{FEAR} = \text{MFEAR} \]
\[ Q\text{ANGER} = \text{MANGER} \]
\[ Q\text{ATTR} = \text{MATTR} \]

WRITE (6, 110) MTAL, QFEAR, QANGER, QATTR
GO TO 127

C UNPACK AND CUMULATE
125 MT = 0
MM = MPERM(I, J, K)
LFAM = MM / 1000000
IF (LFAM .EQ. 0) GO TO 126
MM = MM - 1000000 * LFAM
LFEAR = MM / 10000
MM = MM - 10000 * LFEAR
MFEAR = MFEAR + LFEAR
LANGER = MM / 100
LATTH = MM - 100 * LANGER
MATTR = MATTR + LATTH
MANGER = MANGER + LANGER
MTAL = MTAL + 1
126 GO TO (111, 113, 115, 117, 119, 121, 123), MIDEN
127 GO TO (16, 14, 30), MKEY
28 IF (NSTOP .NE. 0) GO TO 29

CHANGE ENVIRONMENTS
WRITE (6, 8)
WRITE (6, 8)
WRITE (6, 2)
NSTOP = 1
NSTOP = M2ND
MKEY = 2
GO TO 26

C ENDING
29 MKEY = 3
GO TO 26
30 WRITE (6,31)
31 FORMAT (/2X,23HALDONS PERMANENT MEMORY /)
DO 32 I=1,11
DO 32 J=1,11
32 WRITE (6,33) I,J,(MPERM(I,J,K),K=1,11)
33 FORMAT(2I3,2X,11X10)
STOP
END

SUBROUTINE ALDOUS (MPERM,TEMPP,PARAMS,STIM1,STIM2,STIM3,
                     IDEGR,IACTN,IR,NDIM1,NDIM2,NDIM3)

DIMENSION MPERM (12,12,12), TEMPP(7), PARAMS(26), IR(17),
                NEXPER(7), XFAMIL(7), WTMEM(7)
C TEST IF SET FOR REPORT ON PREVIOUS SITUATION
NDIM1=STIM1
NDIM2=STIM2
NDIM3=STIM3
XINJUR=NDIM1/10.
FRUSTR=NDIM2/10.
SATISF=NDIM3/10.
IF (TEMPP(I) .EQ.2) GO TO 4
IF (TEMPP(I) .EQ. 1) GO TO 5
C ALDOUS RECOGNITION ROUTINE - USES FUNCTION MFD
C INTRODUCE ERROR ON ERRORK PROPORTION OF TRIALS
DATA IRAN/3333/
C RANDOMLY MODIFY ONE PERCEPTUAL DIMENSION
CALL RANDU(IRAN,JRAN,YFL)
IRAN=JRAN
IF (YFL.GT.PARAMS(1)) GO TO 13
M=YFL*3.
IF (M.NE.0) GO TO 11
NDIM1=MFD(NDIM1)
GO TO 13
11 NDIM2=MFD(NDIM2)
GO TO 13
12 NDIM3=MFD(NDIM3)
C RETRIEVE RELEVANT MEMORY LOCATIONS
13 NITEMP(1)=MPERM(NDIM1,NDIM2,NDIM3)
ITEMP(2)=MPERM(NDIM1,NDIM2,11)
ITEMP(3)=MPERM(NDIM1,11,NDIM3)
ITEMP(4)=MPERM(11,NDIM2,NDIM3)
ITEMP(5)=MPERM(NDIM1,11,11)
ITEMP(6)=MPERM(11,NDIM2,11)
ITEMP(7)=MPERM(11,11,NDIM3)
C INCREASE EXPERIENCE TALLY
MPERM(NDIM1,NDIM2,NDIM3)=ITEMP(1)+1000000
MPERM(NDIM1,NDIM2,11) = ITEMP(2) + 1000000
MPERM(NDIM1,NDIM1,NDIM3) = ITEMP(3) + 1000000
MPERM(11,NDIM2,NDIM3) = ITEMP(4) + 1000000
MPERM(NDIM1,NDIM1,11) = ITEMP(5) + 1000000
MPERM(11,NDIM2,11) = ITEMP(6) + 1000000
MPERM(11,11,NDIM3) = ITEMP(7) + 1000000

C

COMPUTE FAMILIARITY

DO 14 1=1,7
14 NEXPER(I) = ITEMP(I)/1000000
X = NEXPER(1) + 1
IF (X.EQ.1.) GO TO 15
TEMPM(7) = ALOG10(X)
GO TO 18
15 DO 16 1=2,4
IF (NEXPER(I).GE.10) GO TO 17
16 CONTINUE
TEMPM(7) = 0.
GO TO 18
17 TEMPM(7) = -1.
18 CONTINUE

C

EMOTION

UNPACK MEMORY LOCATIONS

DO 21 I=1,7
K = ITEMP(I)/1000000
ITEMP(I) = ITEMP(I) - 1000000*K
XFAMIL(I) = K
ITEMP(I) = ITEMP(I)/100000
K = ITEMP(I)/100
ITEMP(I) = ITEMP(I) - 1000*K
XFEAR(I) = K
K = ITEMP(I)/100
XDESIR(I) = ITEMP(I) - 100*K
21 XANGER(I) = K

C

COMPUTE PREVIOUS ATTITUDE - FEAR

FAMILIARITY WEIGHT

DO 22 I=1,7
22 WTMEM(I) = 1. - (1./(1. + PARAMS(6)*XFAMIL(I)))

C

WEIGHT ABSTRACTION LEVEL

DO 23 I=2,4
23 WTMEM(I+3) = WTMEM(I+3)*PARAMS(24)

C

ACCUMULATE

SUMWT=0.
SFEAR=0.
APEAR=0.

DO 24 I=1,7
24 SFEAR = SFEAR + XFEAR(I)*WTMEM(I)/100.
SUMWT = SUMWT + WTMEM(I)
IF (SUMWT.EQ.0.) GO TO 34
APEAR = SFEAR/SUMWT
C COMPUTE PREVIOUS ATTITUDE - ANGER
34 DO 25 I=1,7
25 WTMEM (I) =1.- 1./(1. + PARAMS (7) * XFAMIL (I))
DO 26 I=2,4
WTMEM (I) =WTMEM (I) * PARAMS (25)
26 WTMEM (I+3) =WTMEM (I+3) * PARAMS (25) * PARAMS (25)
SUMWT=0.
SANGER=0.
AANGER=0.
DO 27 I=1,7
SANGER=SANGER+XANGER (I) * WTMEH (I) /100.
27 SUMHT=SUMWT+WTMEM (I)
IF (SUMWT.EQ.0.) GO TO 35
AANGER=SANGER/SUMWT
C COMPUTE PREVIOUS ATTITUDE - DESIRE
35 DO 28 I=1,7
28 WTMEM (I) =1.- 1./(1. + PARAMS (8) * XFAMIL (I))
DO 29 I=2,4
WTMEM (I) =WTMEM (I) * PARAMS (26)
29 WTMEM (I+3) =WTMEM (I+3) * PARAMS (26) * PARAMS (26)
SUMWT=0.
SDESIR=0.
ADESIR=0.
DO 30 I=1,7
SDESIR=SDESIR+XDESIR (I) * WTMEM (I) /100.
30 SUMWT=SUMWT+WTMEM (I)
IF (SUMWT.EQ.0.) GO TO 36
ADESIR=SDESIR/SUMWT
C COMBINE PREVIOUS ATTITUDE AND CURRENT MOOD
36 TEMPM (4) =PARAMS (3) * TEMPM (4) + (1.-PARAMS (3)) * AFEAR
TEMPM (5) =PARAMS (4) * TEMPM (5) + (1.-PARAMS (4)) * AANGER
TEMPM (6) =PARAMS (5) * TEMPM (6) + (1.-PARAMS (5)) * ADESIR
C EMOTIONS INTERACT, THE STRONGEST WEAKENS THE OTHERS
TEMPM (3) =0
IF (AMIN1 ( (TEMPM (4) - TEMPM (5)) , (TEMPM (4) - TEMPM (6))) .LT. 1) GO TO 31
DEGR=TEHPM(4)
TEMPM (5) =TEMPM (5) * PARAMS (9)
TEMPM (6) =TEMPM (6) * PARAMS (10)
GO TO 37
31 IF (AMIN1 ( (TEMPM (5) - TEMPM (4)) , (TEMPM (5) - TEMPM (6))) .LT. 1) GO TO 32
DEGR=TEHPM(5)
TEMPM (4) =TEMPM (4) * PARAMS (11)
TEMPM (6) =TEMPM (6) * PARAMS (12)
GO TO 37
32 IF (AMIN1 ( (TEMPM (6) - TEMPM (4)) , (TEMPM (6) - TEMPM (5))) .LT. 1) GO TO 33
DEGR=TEHPM(6)
TEMPM(4) = TEMPM(4) * PARAMS(13)
TEMPM(5) = TEMPM(5) * PARAMS(14)
GO TO 37

33 TEMPM(3) = 1
DEGR = AMAX1(TEMPM(4), TEMPM(5), TEMPM(6))
37 IF (TEMPM(1) .EQ. 3.0 .OR. TEMPM(1) .EQ. 4) GO TO 1

C ALDOUS ACTION ROUTINE
C SELECT ACTION - SET IDEGR AND IACTN APPROPRIATELY
 IDEGR = 0
IF (DEGR .GE. PARAMS(15)) GO TO 41
IACTN = 5
C INDIFFERENCE
GO TO 1
41 IF (DEGR .LE. PARAMS(17)) GO TO 42
IACTN = 6
C PARALYSIS
GO TO 1
42 IF (DEGR .GE. PARAMS(16)) GO TO 43
IDEGR = 1
C MILD
GO TO 44
43 IDEGR = 2
C STRONG
44 IF (TEMPM(3) .NE. 1) GO TO 45
IACTN = 4
C CONFLICT
GO TO 1
45 IF (DEGR .NE. TEMPM(6)) GO TO 46
IACTN = 1
C APPROACH
GO TO 1
46 IF (DEGR .NE. TEMPM(4)) GO TO 47
IACTN = 2
C AVOIDANCE
GO TO 1
47 IACTN = 3
C ATTACK
C UPDATE IMMEDIATE MEMORY
1 TEMPM(2) = 10000 * NDIM1 + 100 * NDIM2 + NDIM3
C TEST IF SET FOR REPORT
2 IF (TEMPM(1) .GE. 2) GO TO 4
RETURN
4 CONTINUE
C UNPACK TEMPORARY MEMORY
K = TEMPM(2)
NDIM1 = K / 10000
K = K - 10000 * NDIM1
NDIM2 = K / 100
NDIM3 = K - 100 * NDIM2
C
PRINT THE QUESTION
IF (TEMPM(1) .NE. 2) GO TO 67
IR(1) = 1
GO TO 70
67 IF (TEMPM(1) .NE. 3) GO TO 68
IR(1) = 2
GO TO 69
68 IF (TEMPM(1) .NE. 4) GO TO 77
IR(1) = 3
C
SUBJECT IN PRESENT TENSE
69 ISUBJ1 = 6
ISUBJ2 = 4
ISUBJ3 = 5
ISUBJ4 = 4
GO TO 71
C
SUBJECT IN PAST TENSE
70 ISUBJ1 = 3
ISUBJ2 = 1
ISUBJ3 = 2
ISUBJ4 = 1
C
FAMILIARITY
71 IF (TEMPM(7) .EQ. -1.) GO TO 72
IDEGR1 = LIMITD (TEMPM(7) * PARAMS(2) + 1.75)
IADJ1 = 6
GO TO 73
72 IDEGR1 = LIMITD (PARAMS(2) + 2.)
IADJ1 = 5
73 IR(2) = ISUBJ1
IR(3) = IDEGR1
IR(4) = IADJ1
C
—EX., IT IS FAIRLY FAMILIAR
C
EMOTIONS
IDEGR2 = LIMITD (5. * TEMPM(6) + 1. ) * PARAMS(2)
IADJ2 = 1
IDEGR3 = LIMITD (5. * TEMPM(4) + 1. ) * PARAMS(2)
IADJ3 = 2
IDEGR4 = LIMITD (5. * TEMPM(5) + 1. ) * PARAMS(2)
IADJ4 = 3
IR(5) = ISUBJ2
IR(6) = IDEGR2
IR(7) = IADJ2
IR(8) = IDEGR3
IR(9) = IADJ3
IR(10) = IDEGR4
IR(11) = IADJ4
C
CONFlict
IF (TEMPO(3) .NE. 1) GO TO 75
EMOMAX = AMAX1 (TEMPO(4), TEMPO(6), TEMPO(5)) * PARAMS(2)
IF (EMOMAX .LT. -25) GO TO 75
IF(EMOMAX.LE.45) GO TO 74
IDEGR5=2
GO TO 76
74 IDEGR5=1
GO TO 76
75 IDEGR5=3
76 IADJ5=4
C
TENSION
IDEGR6=LIMITD(((TEMPM(4)+TEMPM(5)+TEMPM(6)-.3)/.4)
1*PARAMS(2))
IADJ6=4
IR(12)=ISUBJ3
IR(13)=IDEGR5
IR(14)=IADJ5
IR(15)=ISUBJ4
IR(16)=IDEGR6
IR(17)=IADJ6
77 RETURN
C
REACTION TO CONSEQUENCES
5 TEMPM(4)=XINJUR*PARAMS(18)+TEMPM(4)*(1.-PARAMS(18))
TEHPM(5)=FRUST*PARAMS(19)+TEMPM(5)*(1.-PARAMS(19))
TEHPM(6)=SATISF*PARAMS(20)+TEMPM(6)*(1.-PARAMS(20))
TEHPM(4)=REGUL(TEMPM(4))
TEHPM(5)=REGUL(TEMPM(5))
TEHPM(6)=REGUL(TEMPM(6))
C
RETURN
C
ALDOUS SUBROUTINE FOR MODIFYING PERMANENT MEMORY
C
OBTAIN RELEVANT MEMORY LOCATIONS
ITEMP(1)=MPERM(NDIM1,NDIM2,NDIM3)
ITEMP(2)=MPEHM(NDIM1,NDIM2,11)
ITEMP(3)=MPEHM(NDIM1,11,NDIM3)
ITEMP(4)=MPEHM(11,NDIM2,NDIM3)
ITEMP(5)=MPEHM(NDIM1,11,11)
ITEMP(6)=MPEHM(11,NDIM2,11)
ITEMP(7)=MPEHM(11,11,NDIM3)
C
MODIFY MEMORY
DO 50 I=1,7
C
UNPACK A LOCATION
K=ITEMP(I)/1000000
ITEMP(I)=ITEMP(I)-1000000*K
ZFAMIL=K
K=ITEMP(I)/10000
ITEMP(I)=ITEMP(I)-10000*K
ZFEAR=K
\[ K = \text{ITEMP}(I)/100 \]
\[ Z_{\text{DESI}}R = \text{ITEMP}(I) - 100*K \]
\[ Z_{\text{ANG}}ER = K \]
\[ E = \log_{10}(Z_{\text{FA}}H_{\text{IL}}) \]
\[ W_{\text{FEAR}} = 1/(10.*\text{PARAMS}(21)^{**E}) \]
\[ W_{\text{WANGER}} = 1/(10.*\text{PARAMS}(22)^{**E}) \]
\[ W_{\text{DESIR}} = 1/(10.*\text{PARAMS}(23)^{**E}) \]

C CHANGE MEMORY ELEMENTS
\[ I_{\text{FEAR}} = 100.*(Z_{\text{FEAR}}/100.+W_{\text{FEAR}}*\text{TEMPM}(4))/(1.+W_{\text{FEAR}}) \]
\[ I_{\text{WANGER}} = 100.*(Z_{\text{WANGER}}/100.+W_{\text{WANGER}}*\text{TEMPM}(5))/(1.+W_{\text{WANGER}}) \]
\[ I_{\text{DESIR}} = 100.*(Z_{\text{DESIR}}/100.+W_{\text{DESIR}}*\text{TEMPM}(6))/(1.+W_{\text{DESIR}}) \]

C REPACK
\[ 50 \text{ITEMP}(I) = I_{\text{DESIR}} + 100*I_{\text{WANGER}} + 10000*I_{\text{FEAR}} + 1000000*I_{\text{FAMIL}} \]

C STORE IN PERMANENT MEMORY
\[ \text{MPERM}(NDIM1,NDIM2,NDIM3) = \text{ITEMP}(1) \]
\[ \text{MPERM}(NDIM1,NDIM2,11) = \text{ITEMP}(2) \]
\[ \text{MPERM}(NDIM1,11,NDIM3) = \text{ITEMP}(3) \]
\[ \text{MPERM}(11,NDIM2,NDIM3) = \text{ITEMP}(4) \]
\[ \text{MPERM}(NDIM1,11,11) = \text{ITEMP}(5) \]
\[ \text{MPERM}(11,NDIM2,11) = \text{ITEMP}(6) \]
\[ \text{MPERM}(11,11,NDIM3) = \text{ITEMP}(7) \]

GO TO 2
END

FUNCTION \text{MFD}(NDIMX)
C RANDOMLY MODIFIES INPUTS
\[ K = 1 \]
IF(NDIMX.EQ.1) GO TO 20
IF(NDIMX.EQ.10) GO TO 22
CALL RANDU(IRAN,JEAN,YFL)
IRAN=JEAN
K=1
IF(YFL.LE.-5) GO TO 20
K=-1
20 \text{MFD} = NDIMX + K
RETURN
22 \text{MFD} = NDIMX - 1
RETURN
END

FUNCTION \text{REGUL}(EMOT)
C KEEPS INPUT IN RANGE 0.0 TO 0.99
\[ \text{REGUL} = 0. \]
IF(EMOT.LT.0.) GO TO 21
IF(EMOT.GT.99) GO TO 23
\[ \text{REGUL} = \text{EMOT} \]
RETURN
21 \text{REGUL} = .99
RETURN
RETURN
END

FUNCTION LIMITD(DEGR)
  C  KEEPS INPUT IN RANGE 1 TO 5
  IF(DEGR.LE.5.) GO TO 22
  LIMITD=5
  RETURN
  22 IF(DEGR.GE.1.) GO TO 24
  LIMITD=1
  RETURN
  24 LIMITD=DEGR
  RETURN
END
APPENDIX C

Flowchart of Streamlined FORTRAN ALOUS
APPENDIX C

Flowchart of Streamlined FORTRAN ALDOUS
APPENDIX D

Program Text of PL/I ALDODS with Array Memory

/* MAIN PROGRAM -- ALDODS ENVIRONMENT */
MAIN: PROCEDURE OPTIONS (MAIN);
/* INPUT: */
/* RUN TITLE -- ONE CARD, FORMAT A(80) */
/* CHANGE NOTE -- ONE CARD, FORMAT A(80) */
/* PRINTED AT ENVIRONMENT CHANGE. */
/* IF NO SECOND ENVIRONMENT, LEAVE BLANK. */
/* RUN PARAMETERS -- ONE CARD, FORMAT 3 F(10) */
/* M1ST -- TRIALS IN FIRST ENVIRONMENT */
/* M2ND -- TRIALS IN SECOND ENVIRONMENT */
/* LREPT -- 1=ACTION PRINTED EACH TRIAL */
/* 2=ACTION PLUS REPORT */
/* 0=SUMMARY ONLY PRINTED */
/* ALDOUS PARAMETERS -- FORMAT 26 F(3,2), 3 F(3,2) */
/* FIRST CARD IS MEM (26), SECOND IS SUM (3) */
/* INPUT STIMULI -- FORMAT 35 F(2) */

DCL
(STIMU(7,1000), LRESP(1000), IR(17), NRESP(10)) FIXED BINARY,
(NRESP(10), RESPT(10), MEM (26), SUM (3), SUM (3),
QFEAR, QANGER, QATTR) FIXED BINARY (15,5),
TEMPM(7) FIXED BINARY (31,5),
MEMORY ENTRY (FIXED BINARY, FIXED BINARY, FIXED BINARY,
FIXED BINARY, FIXED BINARY (31,0)),
ADJUST ENTRY (FIXED BINARY, FIXED BINARY (31,0),
FIXED BINARY, FIXED BINARY, FIXED BINARY, FIXED BINARY,
FIXED BINARY, FIXED BINARY, FIXED BINARY),
ALDOUS ENTRY ((7) FIXED BINARY (31,5),
(26) FIXED BINARY (15,5), FIXED BINARY,
FIXED BINARY, FIXED BINARY, FIXED BINARY, FIXED BINARY,
(17) FIXED BINARY, FIXED BINARY, FIXED BINARY, FIXED BINARY),
CARD CHAR (80),
(ICON, MM) FIXED BINARY (31,0);
GET FILE (SYSIN) EDIT (CARD) (A(80));
PUT FILE (SYSPRINT) EDIT (CARD) (SKIP, A(80));
GET FILE (SYSIN) EDIT (CARD) (A(80));
GET FILE (SYSIN) EDIT (M1ST, M2ND, LREPT) (3 F(10));
GET FILE (SYSIN) EDIT
((MEM (I) DO I=1 TO 26), (SUM (J) DO J=1 TO 3))
(SKIP, 26 F(3,2), SKIP, 3 F(3,2));
IK=0;
IAD1, IAD2, IAD3=1;
ICON=0;
CALL MEMORY (IK, IAD1, IAD2, IAD3, ICON);
TEMPM = 0;
IDEGR, IACTN, NSTOP = 0;
MSTOP = NSTOP;

L14:
NRESP, NRESPT = 0;

L16:
MTALLY = 0;
GET FILE (SYSIN) EDIT
((STIMU(I,J) DO I=1 TO 7) DO J=1 TO MSTOP)
(SKIP, 35 F(2));
DO J=1 TO 1000;
TEMPM(1) = 0;
CALL ALDOUS (TEHPM, PARAMS, STIMU(1,J), STIMU(2,J),
STIMU(3,J), IDEGR, IACTN, IR, NDI1, NDI2, NDI3);
IF LREPT<1 THEN GO TO L19;
IF IACTN>=5 THEN GO TO L17;
PUT FILE (SYSPRINT) EDIT (J, (STIMU(I,J) DO I=1 TO 7),
IDEGR, IACTN, (IR(K) DO K=1 TO 17), NDI1, NDI2, NDI3)
(SKIP, F(4), 7 F(3), 2 F(3), 17 F(3), 3 F(10));
GO TO L18;

L17:
PUT FILE (SYSPRINT) EDIT (J, (STIMU(I,J) DO I=1 TO 7),
IDEGR, IACTN, (IR(K) DO K=1 TO 17), NDI1, NDI2, NDI3)
(SKIP, F(4), 7 F(3), 2 F(3), 17 F(3), 3 F(10));

L18:
IF LREPT=2 THEN GO TO L19;
TEMPM(1) = 2;
CALL ALDOUS (TEHPM, PARAMS, STIMU(1,J), STIMU(2,J),
STIMU(3,J), IDEGR, IACTN, IR, NDI1, NDI2, NDI3);

L19:
IF IACTN=5 THEN GO TO L20;
ICODE = 1;
GO TO L23;

L20:
IF IACTN=6 THEN GO TO L21;
ICODE = 10;
GO TO L23;

L21:
IF IACTN=4 THEN GO TO L22;
ICODE = IDEGR+1;
GO TO L23;

L22:
ICODE = 2*IACTN+IDEGR+1;

L23:
NRESP(ICODE) = NRESP(ICODE) + 1;
LRESP(J) = ICODE;
POWER = STIMU(4,J);
/* CONSEQUENCES ROUTINE */
IF IACTN=1 THEN GO TO L104;
/* APPROACH CONSEQUENCES */
IF ((IACTN=4) & (IACTN=5)) THEN GO TO L103;
IF POWER>=POWERK(1) THEN GO TO L104;

/* INDIFFERENCE OR CONFLICT */
L103:
IF ((IACTN=2) & (IACTN=3)) THEN GO TO L105;
IF ((IDEGR=1) | (POWER>=POWERK(2))) THEN GO TO L104;
IF ((IDEGR=2) | (POWER<POWERK(3))) THEN GO TO L105;
L104:
NCONSQ=1;
GO TO L106;
L105:
NCONSQ=0;
L106:
IF NCONSQ=1 THEN GO TO L24;
TEMPM(1)=1;
CALL ALDOUS (TEMPM,PARAMS,STIMU(5,J),STIMU(6,J),
STIMU(7,J),IDEGR,IACTN,IR,NDIM1,NDIM2,NDIM3);
L24:
MTALLY=MTALLY+1;
IF MTALLY=MSTOP THEN GO TO L28;
END;
MKEY=1;
L26:
PUT FILE (SYSPRINT) EDIT ((LRESP(I) DO I=1 TO J))
(SKIP,10(SKIP,X(4),50 P(2)));
PUT FILE (PUNCH) EDIT ((LRESP(I) DO I=1 TO J)) (80 F(1));
PUT FILE (SYSPRINT) EDIT ('CYCLES=',MTALLY,(NRESP(I) DO I=1 TO 10))
(SKIP(2),X(4),A(7),F(10),X(4),10 P(8));
DO I=1 TO 10;
NRESPT(I)=NRESPT(I)+NRESP(I);
RESPT(I)=100*NRESPT(I)/MTALLY;
RESPT(I)=RESPT(I)/100.;
NRESP(I)=0;
END;
PUT FILE (SYSPRINT) EDIT ('CUM. PROPORTIONS=',RESPT)
(SKIP,X(4),A(17),X(4),10 F(8,2));

/* QUESTIONNAIRE ROUTINE */
/* COMPUTES AVERAGE LEVELS OF FEAR, ANGER AND ATTRACTION
FOR A SAMPLE OF ATTITUDES.
EVERY 19TH, 7TH, AND 3RD FOR CONCRETE, 1ST LEVEL
AND 2ND LEVEL OF ABSTRACTION, RESPECTIVELY.
IF NO ATTITUDE, IGNORED. */
MFEAR,MANGER,MATTR,MTAL=0;
/* EVERY 19TH CONCRETE ATTITUDE */

MT=0;
DO I=1 TO 10;
DO J=1 TO 10;
DO K=1 TO 10;
  MT=MT+1;
IF (MT-19) >=0 THEN CALL ADJUST (MT,MM,MTAL,MANGER,MATTR,MFEAR,I,J,K);
END; END; END;

/* EVERY 7TH FIRST LEVEL ABSTRACT ATTITUDE */

MT=0;
I=11;
DO J=1 TO 10;
DO K=1 TO 10;
  MT=MT+1;
IF (MT-7) >=0 THEN CALL ADJUST (MT,MM,MTAL,MANGER,MATTR,MFEAR,I,J,K);
END; END;

J=11;
DO I=1 TO 10;
DO K=1 TO 10;
  MT=MT+1;
IF (MT-7) >=0 THEN CALL ADJUST (MT,MM,MTAL,MANGER,MATTR,MFEAR,I,J,K);
END; END;

K=11;
DO I=1 TO 10;
DO J=1 TO 10;
  MT=MT+1;
IF (MT-7) >=0 THEN CALL ADJUST (MT,MM,MTAL,MANGER,MATTR,MFEAR,I,J,K);
END; END;

/* EVERY 3RD SECOND-LEVEL ABSTRACT ATTITUDE */

MT=0;
I,J=11;
DO K=1 TO 10;
  MT=MT+1;
IF (MT-3) >=0 THEN CALL ADJUST (MT,MM,MTAL,MANGER,MATTR,MFEAR,I,J,K);
END;

I,K=11;
DO J=1 TO 10;
  MT=MT+1;
IF (MT-3) >=0 THEN CALL ADJUST (MT,MM,MTAL,MANGER,MATTR,MFEAR,I,J,K);
END;

J,K=11;
DO I=1 TO 10;
  MT=MT+1;
IF (MT-3)>=0 THEN CALL ADJUST (MT, MM, MTAL, MANGER, MATTR, MFEAR, I, J, K); END;
/* COMPUTE MEANS AND PRINT */
C=MTAL*100;
QFEAR=MFEAR;
QFEAR=QFEAR/C;
QANGER=MANGER;
QANGER=QANGER/C;
QATTR=MATTR;
QATTR=QATTR/C;
PUT FILE (SYSPRINT) EDIT ('EXP=',MTAL,'AT. FEAR=',QFEAR,'AV. ANGER=',QANGER,'ATTRACTION=',QATTR)
(SKIP,X(4),A(4),F(3),X(4),A(9),F(4,2),X(2),A(10),
F(4,2),X(4),A(15),F(4,2));
L27:
IF MKEY=1 THEN GO TO L16;
IF MKEY=2 THEN GO TO L14;
GO TO L30;
L28:
IF NSTOP=0 THEN GO TO L29;
PUT FILE (SYSPRINT) EDIT (CARD) (SKIP(2),A(80));
NSTOP=1;
MSTOP=M2ND;
MKEY=2;
GO TO L26;
L29:
MKEY=3;
GO TO L26;
L30:
IK=3;
CALL MEMORY (IK,IAD1,IAD2,IAD3,ICON);
/* ADJUST */
/* UNPACKS MEMORY LOCATION FOR SAMPLING */
ADJUST: PROCEDURE (MT, MM, MTAL, MANGER, MATTR, MFEAR, I, J, K);
DCL MM FIXED BINARY (31,0);
MT=0;
IK=2;
CALL MEMORY (IK,I,J,K,MM);
LFAM=MM/1000000;
IF LFAM=0 THEN GO TO L126;
MM=MM-1000000*LFAM;
LFEAR=MM/10000;
MM=MM-10000*LFEAR;
MFEAR=MFEAR+LFEAR;
LANGER=MM/100;
LATTR=MM-100*LANGER;
MATTR=MATTR+LATTR;
MANGER=MANGER+LANGER;
MTAL=MTAL+1;
L126:
    END ADJUST;

    /* ALDOUS */
ALDOUS: PROCEDURE (TEMPM,PARAMS,ND1,ND2,ND3,IDEGR,IACTN,
IR,NDIM1,NDIM2,NDIM3);
DCL
(X,NEXPEE (7),XFEAR (7) ,XANGER (7) ,XDESIR (7) ,XFAMIL (7) ,
WTMEM (7) ,XINJUR,FRUSTR,SATISF,YFL,M,SUMWT,SFEAR,
AFEAR,SANGER,AANGER,SDESIR,ADESIR,DEGR,EMOMAX,
ZFAMIL,ZDESIR,ZANGER,E,WFEAR,WANGER,WDDESIR,ZFEAR)
FIXED BINARY (15,5),
TEMPM (7) FIXED BINARY (31,5),
PARAMS (26) FIXED BINARY (15,5),
IR (17) FIXED BINARY,
(ITEMP (7) ,IM) FIXED BINARY (31,0),
MFD ENTRY (FIXED BINARY) RETURNS (FIXED BINARY),
LIMITD ENTRY (FIXED BINARY (15,5))
    RETURNS (FIXED BINARY (15,5)),
REGUL ENTRY (FIXED BINARY (31,5))
    RETURNS (FIXED BINARY (31,5));
NDIM1=ND1;
NDIM2=ND2;
NDIM3=ND3;
XINJUR=NDIM1/10.;
FRUSTR=NDIM2/10.;
SATISF=NDIM3/10.;
IF TEMPM (1) =2 THEN GO TO LA4;
IF TEMPM (1) =1 THEN GO TO LA5;
    /* RECOGNITION ROUTINE */
GET FILE (RAN) EDIT (YFL) (F (2));
YFL=YFL/100.;
IF YFL>PARAMS (1) THEN GO TO LA13;
M=YFL*3.;
IF M==0 THEN GO TO LA11;
NDIM1=MFD(NDIM1);
GO TO LA13;
LA11:
IF M==1 THEN GO TO LA12;
NDIM2=MFD(NDIM2);
GO TO LA13;
LA12:
NDIM3=MFD(NDIM3);
LA13:
    /* RETRIEVE RELEVANT LOCATIONS */
IK=2;
IL=11;
CALL MEMORY (IK,NDIM1,NDIM2,NDIM3,ITEMP(1));
CALL MEMORY (IK,NDIM1,NDIM2,IL,ITEMP(2));
CALL MEMORY (IK,NDIM1,IL,NDIM3,ITEMP(3));
CALL MEMORY (IK,IL,NDIM2,NDIM3,ITEMP(4));
CALL MEMORY (IK,IL,IL,IL,ITEMP(5));
CALL MEMORY (IK,IL,IL,NDIM3,ITEMP(6));
CALL MEMORY (IK,IL,IL,NDIM3,ITEMP(7));
/* STORE EXPERIENCES */
IK=1;
IM=ITEMP(1)+1000000;
CALL MEMORY (IK,NDIM1,NDIM2,NDIM3,IM);
IM=ITEMP(2)+1000000;
CALL MEMORY (IK,NDIM1,NDIM2,IL,IM);
IM=ITEMP(3)+1000000;
CALL MEMORY (IK,NDIM1,IL,NDIM3,IM);
IM=ITEMP(4)+1000000;
CALL MEMORY (IK,IL,NDIM2,NDIM3,IM);
IM=ITEMP(5)+1000000;
CALL MEMORY (IK,IL,IL,IL,IM);
IM=ITEMP(6)+1000000;
CALL MEMORY (IK,IL,IL,NDIM3,IM);
DO I=1 TO 7;
   NEXPER(I)=ITEMP(I)/1000000;
END;
X=NEXPER(1)+1;
IF X=1. THEN GO TO LA15;
TEMPM(7)=LOG10(X);
GO TO LA18;
LA15:
DO I=2 TO 4;
   IF NEXPER(I)>=10 THEN GO TO LA17;
END;
TEMPM(7)=0;
GO TO LA18;
LA17:
TEMPM(7)=-1;
LA18:
/* EMOTION ROUTINE */
/* UNPACK MEMORY LOCATIONS */
DO I=1 TO 7;
   K=ITEMP(I)/1000000;
   ITEMP(I)=ITEMP(I)-1000000*K;
   XFAMIL(I)=K;
   K=ITEMP(I)/10000;
   ITEMP(I)=ITEMP(I)/10000*K;
   XFEAR(I)=K;
   K=ITEMP(I)/100;

XDESIR(I)=ITEMP(I) - 100*K;
XANGER(I)=K;
END;

/* COMPUTE PREVIOUS ATTITUDE—FEAR */
DO I=1 TO 7;
    WTMEM(I)=1.-(1./(1.+PARAMS(6)*XFAMIL(I)));
END;

/* WEIGHT ABSTRACTION LEVEL */
DO I=2 TO 4;
    WTMEM(I)=WTMEM(I)*PARAMS(24);
    WTMEM(I+3)=WTMEM(I+3)*PARAMS(24)*PARAMS(24);
END;

/* ACCUMULATE */
SUMWT,SFEAR,AFEAR=0.;
DO I=1 TO 7;
    SFEAR=SFEAR+XFEAR(I)*WTMEM(I)/100.;
    SUMWT=SUMWT+WTMEM(I);
END;
IF SUMWT=0. THEN GO TO LA34;
    AFEAR=SFEAR/SUMWT;
/* COMPUTE PREVIOUS ATTITUDE—ANGER */
LA34:
DO I=1 TO 7;
    WTMEM(I)=1.-(1./(1.+PARAMS(7)*XFAMIL(I)));
END;
DO I=2 TO 4;
    WTMEM(I)=WTMEM(I)*PARAMS(25);
    WTMEM(I+3)=WTMEM(I+3)*PARAMS(25)*PARAMS(25);
END;
SUMWT,SANGER,AANGER=0.;
DO I=1 TO 7;
    SANGER=SANGER+XANGER(I)*WTMEM(I)/100.;
    SUMWT=SUMWT+WTMEM(I);
END;
IF SUMWT=0. THEN GO TO LA35;
    AANGER=SANGER/SUMWT;
/* COMPUTE PREVIOUS ATTITUDE—DESIRE */
LA35:
DO I=1 TO 7;
    WTMEM(I)=1.-(1./(1.+PARAMS(8)*XFAMIL(I)));
END;
DO I=2 TO 4;
    WTMEM(I)=WTMEM(I)*PARAMS(26);
    WTMEM(I+3)=WTMEM(I+3)*PARAMS(26)*PARAMS(26);
END;
SUMWT,SDESIR,ADESIR=0.;
DO I=1 TO 7;
    SDESIR=SDESIR+XDESIR(I)*WTMEM(I)/100.;
    SUMWT=SUMWT+WTMEM(I);
END;
IF SUMWT=0. THEN GO TO LA36;
ADESIR=SDESIR/SUMWT;
/* COMBINE PREVIOUS ATTITUDE AND CURRENT MOOD */
LA36:
  TEMPM(4)=PARAMS(3)*TEPM(4)+(1.-PARAMS(3))*APEAR;
  TEMPM(5)=PARAMS(4)*TEPM(5)+(1.-PARAMS(4))*AANGER;
  TEMPM(6)=PARAMS(5)*TEPM(6)+(1.-PARAMS(5))*ADESIR;
/* EMOTIONS INTERACT, THE STRONGEST WEAKENS THE OTHERS */
  TEMPM(3)=0;
  IF (MIN((TEPM(4)-TEPM(5)),(TEPM(4)-TEPM(6)))<0.1)
    THEN GO TO LA31;
  DEGR=TEPM(4);
  TEMPM(5)=TEPM(5)*PARAMS(9);
  TEMPM(6)=TEPM(6)*PARAMS(10);
  GO TO LA37;
LA31:
  IF (MIN((TEPM(5)-TEPM(4)),(TEPM(5)-TEPM(6)))<0.1)
    THEN GO TO LA32;
  DEGR=TEPM(5);
  TEMPM(4)=TEPM(4)*PARAMS(11);
  TEMPM(6)=TEPM(6)*PARAMS(12);
  GO TO LA37;
LA32:
  IF (MIN((TEPM(6)-TEPM(4)),(TEPM(6)-TEPM(5)))<0.1)
    THEN GO TO LA33;
  DEGR=TEPM(6);
  TEMPM(4)=TEPM(4)*PARAMS(13);
  TEMPM(5)=TEPM(5)*PARAMS(14);
  GO TO LA37;
LA33:
  TEMPM(3)=1;
  DEGR=MAX(TEPM(4),TEPM(5),TEPM(6));
LA37:
  IF (TEPM(1)=3) | (TEPM(1)=4) THEN GO TO LA1;
/* ALDOUS ACTION ROUTINE */
    IDEGR=0;
    IF DEGR>PARAMS(15) THEN GO TO LA41;
    IACTN=5;
    GO TO LA1;
/* INDIFFERENCE */
LA41:
    IF DEGR<=PARAMS(17) THEN GO TO LA42;
    IACTN=6;
    GO TO LA1;
/* PARALYSIS */
LA42:
    IF DEGR>PARAMS(16) THEN GO TO LA43;
    IDEGR=1;
GO TO LA44; /* MILD */

LA43:
    IDEGR=2; /* STRONG */

LA44:
    IF TEMPM(3)=1 THEN GO TO LA45;
    IACTN=4;
    GO TO LA1; /* CONFLICT */

LA45:
    IF DEGR=TEMPM(6) THEN GO TO LA46;
    IACTN=1;
    GO TO LA1; /* APPROACH */

LA46:
    IF DEGR=TEMPM(4) THEN GO TO LA47;
    IACTN=2;
    GO TO LA1; /* AVOIDANCE */

LA47:
    IACTN=3;
    /* MAIN ROUTINE */

LA1:
    /* UPDATE IMMEDIATE MEMORY */
    TEMPM(2)=10000*NDIM1+100*NDIM2+NDIM3;

LA2:
    /* TEST IF SET FOR REPORT */
    IF TEMPM(1)>=2 THEN GO TO LA4;
    RETURN;

LA4:
    /* ALDOUS VERBAL REPORT ROUTINE, USES LIMITED */
    /* UNPACK TEMPORARY MEMORY */
    K=TEMPM(2);
    NDIM1=K/10000;
    K=K-10000*NDIM1;
    NDIM2=K/100;
    NDIM3=K-100*NDIM2;
    /* PRINT THE QUESTION */
    IF TEMPM(1)=2 THEN GO TO LA67;
    IR(1)=1;
    GO TO LA70;

LA67:
    IF TEMPM(1)=3 THEN GO TO LA68;
    IR(1)=2;
    GO TO LA69;

LA68:
    IF TEMPM(1)=4 THEN GO TO LA77;
    IR(1)=3;
/* SUBJECT IN PRESENT TENSE */

LA69:
IR(2)=6;
IR(5)=4;
IR(12)=5;
IR(15)=4;
GO TO LA71;

/* SUBJECT IN PAST TENSE */

LA70:
IR(2)=3;
IR(5)=1;
IR(12)=2;
IR(15)=1;

/* FAMILIARITY */

LA71:
IF TEMPM(7)=-1 THEN GO TO LA72;
IR(3)=LIMITD(TEMPM(7)*PARAMS(2)+1.75);
IR(4)=6;
GO TO LA73;

LA72:
IR(3)=LIMITD(PARAMS(2)+2.);
IR(4)=5;

LA73:

/* EMOTIONS */

IR(6)=LIMITD((5.*TEMPM(6)+1.)*PARAMS(2));
IR(7)=1;
IR(8)=LIMITD((5.*TEMPM(4)+1.)*PARAMS(2));
IR(9)=2;
IR(10)=LIMITD((5.*TEMPM(5)+1.)*PARAMS(2));
IR(11)=3;

/* CONFLICT */

IF TEMPM(3)=-1 THEN GO TO LA75;
ENOMAX=MAX(TEMPM(4),TEMPM(6),TEMPM(5))*PARAMS(2);
IF ENOMAX<0.25 THEN GO TO LA75;
IF ENOMAX<=0.45 THEN GO TO LA74;
IR(13)=2;
GO TO LA76;

LA74:
IR(13)=1;
GO TO LA76;

LA75:
IR(13)=3;

LA76:
IR(14)=4;

/* TENSION */

IR(16)=LIMITD(((TEMPM(4)+TEMPM(5)+TEMPM(6)
-.3)/.4)*PARAMS(2));
IR(17)=4;

LA77:
RETURN; /* REACTION ROUTINE */

LA5:

TEMPM(4) = XINJUR * PARAMS(18) + TEMPM(4) * (1 - PARAMS(18));
TEMPM(5) = FRUSTR * PARAMS(19) + TEMPM(5) * (1 - PARAMS(19));
TEMPM(6) = SATISF * PARAMS(20) + TEMPM(6) * (1 - PARAMS(20));
TEMPM(4) = REGUL(TEMPM(4));
TEMPM(5) = REGUL(TEMPM(5));
TEMPM(6) = REGUL(TEMPM(6));
NDIM1 = TEMPM(2) / 10000;
K = TEMPM(2) - NDIM1 * 10000;
NDIM2 = K / 100;
NDIM3 = K - NDIM2 * 100;

/* LEARN ROUTINE */

IK = 2;
IL = 11;
CALL MEMORY (IK, NDIM1, NDIM2, NDIM3, ITEMP(1));
CALL MEMORY (IK, NDIM1, NDIM2, IL, ITEMP(2));
CALL MEMORY (IK, NDIM1, IL, NDIM3, ITEMP(3));
CALL MEMORY (IK, IL, NDIM2, NDIM3, ITEMP(4));
CALL MEMORY (IK, NDIM1, IL, IL, ITEMP(5));
CALL MEMORY (IK, IL, NDIM2, IL, ITEMP(6));
CALL MEMORY (IK, IL, IL, NDIM3, ITEMP(7));
DO I = 1 TO 7;
   K = ITEMP(I) / 1000000;
   ITEMP(I) = ITEMP(I) - 1000000 * K;
   ZFAMIL = K;
   K = ITEMP(I) / 10000;
   ITEMP(I) = ITEMP(I) - 10000 * K;
   ZFEAR = K;
   K = ITEMP(I) / 100;
   ZDESIR = ITEMP(I) - 100 * K;
   ZANGER = K;
   E = LOG10(ZFAMIL);
   WFEAR = 1 / ((10 * PARAMS(21)) ** E);
   WANGER = 1 / ((10 * PARAMS(22)) ** E);
   WDESIR = 1 / ((10 * PARAMS(23)) ** E);
   IPEAR = 100 * (ZPEAR/100 + WFEAR * TEMPM(4) / (1 + WFEAR));
   IANGER = 100 * (ZANGER / 100 + WANGER + TEMPM(5)) / (1 + WANGER);
   IDESIR = 100 * (ZDESIR / 100 + WDESIR * TEMPM(6)) / (1 + WDESIR);
   IFAMIL = ZFAMIL;
   ITEMP(I) = IDESIR + 100 * IANGER + 10000 * IPEAR + 1000000 * IFAMIL;
END;

IK = 1;
CALL MEMORY (IK, NDIM1, NDIM2, NDIM3, ITEMP(1));
CALL MEMORY (IK, NDIM1, NDIM2, IL, ITEMP(2));
CALL MEMORY (IK,NDIM1,IL,NDIM3,ITEMP(3));
CALL MEMORY (IK,IL,NDIM2,NDIM3,ITEMP(4));
CALL MEMORY (IK,NDIM1,IL,IL,ITEMP(5));
CALL MEMORY (IK,IL,NDIM2,IL,ITEMP(6));
CALL MEMORY (IK,IL,IL,NDIM3,ITEMP(7));
GO TO LA2;

/* MFD */
MFD: PROCEDURE (NDIMX) RETURNS (FIXED BINARY);
DCL NDIMX FIXED BINARY;
K=1;
IF NDIMX=1 THEN GO TO LM20;
IF NDIMX=10 THEN GO TO LM22;
GET FILE (RAN) EDIT (IFL) (F (2));
K=1;
IF IFL<=50 THEN GO TO LM20;
K=-1;
LM20:
NDIMX=NDIMX+K;
RETURN (NDIMX);
LM22:
NDIMX=NDIMX-1;
RETURN (NDIMX);
END MFD;

/* REGUL */
REGUL: PROCEDURE (EMOT) RETURNS (FIXED BINARY (31,5));
DCL EMOT FIXED BINARY (31,5);
IF EMOT<0.0 THEN GO TO LR21;
IF EMOT>0.99 THEN GO TO LR23;
RETURN (EMOT);
LR21:
EMOT=0.0;
RETURN (EMOT);
LR23:
EMOT=0.99;
RETURN (EMOT);
END REGUL;

/* LIMITD */
LIMITD: PROCEDURE (DEGR) RETURNS (FIXED BINARY (15,5));
DCL DEGR FIXED BINARY (15,5);
IF DEGR<=5.0 THEN GO TO LL22;
DEGR=5.0;
RETURN (DEGR);
LL22:
IF DEGR>=1. THEN GO TO LL24;
DEGR=1.;
RETURN (DEGR);

LL24:
RETURN (DEGR);
END LIMITD;
END AI DOUS;
MEMORY: PROCEDURE (IK,IAD1,IAD2,IAD3,ICON);
/* UNIQUE TEXT FOR ARRAY VERSION */
DCL MPERM (20,20,20) FIXED BINARY (31,0) STATIC,
ICON FIXED BINARY (31,0),
KOUNT FIXED BINARY (15,5);
/* ZERO MEMORY */
IF IK=0 THEN DO;
MPERM=0;
GO TO DN;
END;
/* STORE IN MEMORY */
IF IK=1 THEN DO;
MPERM(IAD1,IAD2,IAD3)=ICON;
GO TO DN;
END;
/* RETRIEVE FROM MEMORY */
IF IK=2 THEN DO;
ICON=MPERM(IAD1,IAD2,IAD3);
GO TO DN;
END;
/* DUMP MEMORY */
IF IK=3 THEN DO;
KOUNT=0;
PUT FILE (SYSPRINT) EDIT
('ALDOUS PERMANENT MEMORY')
(SKIP(2),X(55),A(23));
DO I=1 TO 11;
DO J=1 TO 11;
DO K=1 TO 11;
IF MPERM(I,J,K)=0 THEN KOUNT=KOUNT+1;
END;
PUT FILE (SYSPRINT) EDIT
(I,J,MPERM(I,J,1),MPERM(I,J,2),MPERM(I,J,3),
MPERM(I,J,4),
MPERM(I,J,5),MPERM(I,J,6),MPERM(I,J,7),MPERM(I,J,8),
MPERM(I,J,9),MPERM(I,J,10),MPERM(I,J,11))
(SKIP,2 F(2),11 F(10));
END;
PUT FILE (SYSPRINT) EDIT
(KOUNT,' OR ',((KOUNT/1331)*100),' PER CENT EMPTY CELLS')
(SKIP(2),F(4),A(4),F(6,2),A(21));
GO TO DN;
END;
PUT FILE (SYSPRINT) EDIT
  ('IK VALUE ',IK,' OUTSIDE RANGE 1 TO 3')
  (SKIP,A(9),F(2),A(21));
DN: RETURN;
   END MEMORY;
   END MAIN;
APPENDIX E

Program Text of PL/I ALDOUS with List-Processing Memory

/* MAIN PROGRAM--ALDOUS ENVIRONMENT */

MAIN: PROCEDURE OPTIONS (MAIN);

/* INPUT:
   RUN TITLE--ONE CARD, FORMAT A(80)
   CHANGE NOTE--ONE CARD, FORMAT A(80)
   PRINTED AT ENVIRONMENT CHANGE.
   IF NO SECOND ENVIRONMENT, LEAVE BLANK.
   RUN PARAMETERS--ONE CARD, FORMAT 3 F(10)
   M1ST--TRIALS IN FIRST ENVIRONMENT
   M2ND--TRIALS IN SECOND ENVIRONMENT
   LREPT--1=ACTION PRINTED EACH TRIAL
   2=ACTION PLUS REPORT
   0=SUMMARY ONLY PRINTED
   ALDOUS PARAMETERS--FORMAT 26 F (3,2), 3 F(3,2)
   FIRST CARD IS PARAMS(26), SECOND IS POWERK(3)
   LOCATION STIMULI--FORM 35 F(2) */

DCL
  (STIMU(7,1000),LRESP(1000),IR(17),NRESP(10)) FIXED BINARY,
  (NRESP(10),RESPT(10),PARAMS(26),POWERK(3),POWER,C,
  QFEAR,CANGER,QATTR) FIXED BINARY (15,5),
  TEMP(7) FIXED BINARY (31,5),
  MEMORY ENTRY (FIXED BINARY,FIXED BINARY,FIXED BINARY,
  FIXED BINARY,FIXED BINARY(31,0)),
  ADJUST ENTRY (FIXED BINARY,FIXED BINARY(31,0),
  FIXED BINARY,FIXED BINARY,FIXED BINARY,FIXED BINARY,
  FIXED BINARY,FIXED BINARY,FIXED BINARY),
  ALDOUS ENTRY ((7) FIXED BINARY (31,5),
  (26) FIXED BINARY (15,5), FIXED BINARY,
  FIXED BINARY,FIXED BINARY,FIXED BINARY,FIXED BINARY,
  FIXED BINARY,FIXED BINARY,FIXED BINARY,FIXED BINARY,
  FIXED BINARY),
  CARD CHAR (80),
  (ICON,MM) FIXED BINARY (31,0);

GET FILE (SYSIN) EDIT (CARD) (A(80));
PUT FILE (SYSPRINT) EDIT (CARD) (A(80));
GET FILE (SYSIN) EDIT (CARD) (A(80));
GET FILE (SYSIN) EDIT (MIST,M2ND,LREPT) (3 F(10));
GET FILE (SYSIN) EDIT ((PARAMS(I) DO I=1 TO 26),
  (POWERK(J) DO J=1 TO 3))
  (SKIP,26 F(3,2),SKIP,3 F(3,2));

IK=0;
IAD1,IAD2,IAD3=1;
ICON=0;
CALL MEMORY (IK,IAD1,IAD2,IAD3,ICON);
TEMPH=0;
IDEGR,IACTN,NSTOP=0;
MSTOP=M1ST;
L14:
NRESP,NRESPT=0;
L16:
MTALLY=0;
GET FILE (SYSIN) EDIT ((STIMU(I,J) DO I=1 TO 7)
   DO J=1 TO MSTOP))
   (SKIP,35 F(2));
DO J=1 TO 1000;
TEMPM(1)=0;
CALL ALDOUS (TEMPM,PARAMS,STIMU(1,J),STIMU(2,J),
   STIMU(3,J),IDEGR,IACTN,IR,NDIM1,NDIM2,NDIM3);
IF LREPT<1 THEN GO TO L19;
IF IACTN>=5 THEN GO TO L17;
PUT FILE (SYSPRINT) EDIT (J,(STIMU(I,J) DO I=1 TO 7),
   IDEGR,IACTN,(IR(K) DO K=1 TO 17),NDIM1,NDIM2,NDIM3)
   (SKIP,F(4),7 F(3),2 F(3),17 F(3),3 F(10));
GO TO L18;
L17:
PUT FILE (SYSPRINT) EDIT (J,(STIMU(I,J) DO I=1 TO 7),
   IDEGR,IACTN,(IR(K) DO K=1 TO 17),NDIM1,NDIM2,NDIM3)
   (SKIP,F(4),7 F(3),2 F(3),17 F(3),3 F(10));
GO TO L18;
L14:
PUT FILE (SYSPRINT) EDIT (J,(STIMU(I,J) DO I=1 TO 7),
   IDEGR,IACTN,(IR(K) DO K=1 TO 17),NDIM1,NDIM2,NDIM3)
   (SKIP,F(4),7 F(3),2 F(3),17 F(3),3 F(10));
L19:
IF IACTN=5 THEN GO TO L20;
ICODE=1;
GO TO L23;
L20:
IF IACTN=6 THEN GO TO L21;
ICODE=10;
GO TO L23;
L21:
IF IACTN=4 THEN GO TO L22;
ICODE=IDEGR+1;
GO TO L23;
L22:
ICODE=2*IACTN+IDEGR+1;
L23:
NRESP(ICODE)=NRESP(ICODE)+1;
LRESP(J)=ICODE;
POWER=STIMU(4,J);
/* CONSEQUENCES ROUTINE */
IF IACTN=1 THEN GO TO L104;
/* APPROACH CONSEQUENCES */
IF ((IACTN="=4) & (IACTN="=5)) THEN GO TO L103;
IF POWER>=POWERK(1) THEN GO TO L104;
   /* INDIFFERENCE OR CONFLICT */
L103:
   IF ((IACTN="=2) & (IACTN="=3)) THEN GO TO L105;
   IF ((IDEGR=1) | (POWER>=POWERK(2))) THEN GO TO L104;
   IF ((IDEGR="=2) | (POWER<POWERK(3))) THEN GO TO L105;
L104:
   NCONSQ=1;
   GO TO L106;
L105:
   NCONSQ=0;
L106:
   IF NCONSQ="=1 THEN GO TO L24;
   TEMPM=(1)="=1;
   CALL ALDODS (TEMPM,PARAMS,STIM0(5,J),STIM0(6,J),
                   STIM0(7,J),IDEGR,IACTN,IR,NDII11,NDIi32,NDIM3);
L24:
   MTALLY=MTALLY+1;
   IF MTALLY=MSTOP THEN GO TO L28;
   END;
   MKEY=1;
L26:
   PUT FILE (SYSPRINT) EDIT ((LRESP(I) DO I=1 TO J))
                   (SKIP,10(SKIP,X(4),50 F(2)));
   PUT FILE (PUNCH) EDIT ((LRESP(I) DO I=1 TO J)) (80 F(1));
   PUT FILE (SYSPRINT) EDIT ('" INDIFF"," W CONFL"," S CONFL"," W APPR"," S APPR"," W AVOID"," S AVOID"," W ATTR"," S ATTR"," PARALYS")
                   (SKIP(2),X(25),10 A(8));
   PUT FILE (SYSPRINT) EDIT
                   ("CYCLES=",MTALLY,(NRESP(I) DO I=1 TO 10))
                   (SKIP(2),X(4),A(17),X(4),10 F(6));
   DO I=1 TO 10;
      NRESPT(I)=NRESPT(I)+NRESP(I);
      RESPT(I)=100*NRESPT(I)/MTALLY;
      RESPT(I)=RESPT(I)/100.;
      NRESP(I)=0;
   END;
   PUT FILE (SYSPRINT) EDIT ('CUM. PROPORTIONS=" RESPT)
                   (SKIP,X(4),A(17),X(4),10 F(8,2));
   /* QUESTIONNAIRE ROUTINE */
/* COMPUTES AVERAGE LEVELS OF FEAR, ANGER AND ATTRACTION
   FOR A SAMPLE OF ATTITUDES.
   EVERY 1ST, 7TH AND 3RD FOR CONCRETE, 1ST AND
   2ND LEVEL OF ABSTRACTION, RESPECTIVELY.
   IF NO ATTITUDE, IGNORED. */
MFEAR,MANGER,MATTR,MTAL=0;
   /* EVERY 19TH CONCRETE ATTITUDE */
MT=0;
DO I=1 TO 10,12 TO 20;
DO J=1 TO 10,12 TO 20;
DO K=1 TO 10,12 TO 20;
    MT=MT+1;
    IF (MT-19) >= 0
    THEN CALL ADJUST (MT, MM, MTAL, MANGER, MATTR, MFEAR, I, J, K);
      END; END; END;
/* EVERY 7TH FIRST LEVEL ABSTRACT ATTITUDE */
MT=0;
I=11;
DO J=1 TO 10,12 TO 20;
DO K=1 TO 10,12 TO 20;
    MT=MT+1;
    IF (MT-7) >= 0
    THEN CALL ADJUST (MT, MM, MTAL, MANGER, MATTR, MFEAR, I, J, K);
      END; END;
J=11;
DO I=1 TO 10,12 TO 20;
DO K=1 TO 10,12 TO 20;
    MT=MT+1;
    IF (MT-7) >= 0
    THEN CALL ADJUST (MT, MM, MTAL, MANGER, MATTR, MFEAR, I, J, K);
      END; END;
K=11;
DO I=1 TO 10,12 TO 20;
DO J=1 TO 10,12 TO 20;
    MT=MT+1;
    IF (MT-7) >= 0
    THEN CALL ADJUST (MT, MM, MTAL, MANGER, MATTR, MFEAR, I, J, K);
      END; END;
/* EVERY 3RD SECOND-LEVEL ABSTRACT ATTITUDE */
MT=0;
I,J=11;
DO K=1 TO 10,12 TO 20;
    MT=MT+1;
    IF (MT-3) >= 0
    THEN CALL ADJUST (MT, MM, MTAL, MANGER, MATTR, MFEAR, I, J, K);
      END; END;
I,K=11;
DO J=1 TO 10,12 TO 20;
    MT=MT+1;
    IF (MT-3) >= 0
    THEN CALL ADJUST (MT, MM, MTAL, MANGER, MATTR, MFEAR, I, J, K);
      END; END;
J,K=11;
DO I=1 TO 10,12 TO 20;
    MT=MT+1;
    IF (MT-3) >= 0
    THEN CALL ADJUST (MT, MM, MTAL, MANGER, MATTR, MFEAR, I, J, K);
      END; END;
THEN CALL ADJUST (MT, MM, MTAL, MANGER, MATTR, MFEAR, I, J, K);
END;

/* COMPUTE MEANS AND PRINT */
C = MTAL * 100;
QFEAR = MFEAR;
QFEAR = QFEAR / C;
QANGER = MANGER;
QANGER = QANGER / C;
QATTR = MATTR;
QATTR = QATTR / C;
PUT FILE (SYSPRINT) EDIT ('EXP=", MTAL, 'AV. FEAR=", QFEAR,
"AV. ANGER=", QANGER, 'AV. ATTRACTION=", QATTR)
(SKIP, X(4), A(4), F(3), X(4), A(9), F(4,2), X(2), A(10),
F(4,2), X(4), A(15), F(4,2));
L27:
IF MKEY = 1 THEN GO TO L16;
IF MKEY = 2 THEN GO TO L14;
GO TO L30;
L28:
IF NSTOP = 0 THEN GO TO L29;
PUT FILE (SYSPRINT) EDIT (CARD) (SKIP(2), A(80));
NSTOP = 1;
NSTOP = M2ND;
MKEY = 2;
GO TO L26;
L29:
MKEY = 3;
GO TO L26;
L30:
IK = 3;
CALL MEMORY (IK, IAD1, IAD2, IAD3, ICON);

/ * ADJUST */

/ * UNPACKS MEMORY LOCATION FOR SAMPLING */
ADJUST: PROCEDURE (MT, MM, MTAL, MANGER, MATTR, MFEAR, I, J, K);
DCL MM FIXED BINARY (31, 0);
MT = 0;
IK = 2;
CALL MEMORY (IK, I, J, K, MM);
LFAM = MM / 1000000;
IF LFAM = 0 THEN GO TO L126;
MM = MM - 1000000 * LFAM;
LFEAR = MM / 10000;
MM = MM - 10000 * LFEAR;
MFEAR = MFEAR * LFEAR;
LANGER = MM / 100;
LATTR = MM - 100 * LANGER;
MATTR = MATTR + LATTR;
MANGER = MANGER + LANGER;
MTAL=MTAL+1;

L126:  
END ADJUST;

/* ALDOUS */

ALDOUS: PROCEDURE (TEMPM,PARAMS,ND1,ND2,ND3,IDEGR,IACTN,
IR,NDIM1,NDIM2,NDIM3);

DCL
(X,ZPER (7) ,XFEAR(7) ,XANGER (7) ,XDESIR(7) ,XFAMIL(7) ,
WMEN(7),XINJUR,FRUSTR,SATISF,YFL,EM,SUMWT,SWFAR,
AFEAR,SANGER,AANGER,SDESIR,ADESIR,DEGR,EMOMAX,
ZFAM1,2DESIR,ZANGER,E,WFEAR,WANGER,WDESIR,WFEAR)
FIXED BINARY (15,5),
TEMPM(7) FIXED BINARY (31,5),
PARAMS(26) FIXED BINARY (15,5),
IR(17) FIXED BINARY,
(TOTM(7),IM) FIXED BINARY (31,0),
MFD ENTRY (FIXED BINARY) RETURNS (FIXED BINARY),
LIMIT ENTRY (FIXED BINARY (15,5))
RETURNS (FIXED BINARY (15,5)),
REGUL ENTRY (FIXED BINARY (31,5))
RETURNS (FIXED BINARY (31,5));

NDIM1=ND1;
NDIM2=ND2;
NDIM3=ND3;
XINJUR=NDIM1/10.;
FRUSTR=NDIM2/10.;
SATISF=NDIM3/10.;
IF TEMPM(1)=2 THEN GO TO LA4;
IF TEMPM(1)=1 THEN GO TO LA5;

/* RECOGNITION ROUTINE */
GET FILE {RAW} EDIT {YFL} {F(2)};
YFL=YFL/100.;
IF YFL>PARAMS(1) THEN GO TO LA13;
M=YFL*3.;
IF M=.=0 THEN GO TO LA11;
NDIM1=MFD(NDIM1);
GO TO LA13;
LA11:
IF M=.=1 THEN GO TO LA12;
NDIM2=MFD(NDIM2);
GO TO LA13;
LA12:
NDIM3=MFD(NDIM3);
LA13:
/* RETRIEVE RELEVANT MEMORY LOCATIONS */
IK=2;
IL=11;
CALL MEMORY (IK,NDIM1,NDIM2,NDIM3,ITEMP(1));
CALL MEMORY (IK,NDIM1,NDIM2,IL,ITEMP(2));
CALL MEMORY (IK,NDIM1,IL,NDIM3,ITEMP(3));
CALL MEMORY (IK,IL,NDIM2,NDIM3,ITEMP(4));
CALL MEMORY (IK,NDIM1,IL,IL,ITEMP(5));
CALL MEMORY (IK,IL,NDIM2,IL,ITEMP(6));
CALL MEMORY (IK,IL,IL,NDIM3,ITEMP(7));
/* STORE EXPERIENCES */
IK=1;
IM=ITEMP(1)+1000000;
CALL MEMORY (IK,NDIM1,NDIM2,NDIM3,IM);
IM=ITEMP(2)+1000000;
CALL MEMORY (IK,NDIM1,NDIM2,IL,IM);
IM=ITEMP(3)+1000000;
CALL MEMORY (IK,NDIM1,IL,NDIM3,IM);
IM=ITEMP(4)+1000000;
CALL MEMORY (IK,IL,NDIM2,NDIM3,IM);
IM=ITEMP(5)+1000000;
CALL MEMORY (IK,IL,IL,IL,IM);
IM=ITEMP(6)+1000000;
CALL MEMORY (IK,IL,IL,NDIM3,IM);
IM=ITEMP(7)+1000000;
CALL MEMORY (IK,IL,IL,IL,IM);
DO I=1 TO 7;
   NEXPER(I)=ITEMP(I)/1000000;
END;
X=NEXPER(1)+1;
IF X=1. THEN GO TO LA15;
TEMPM(7)=LOG10(X);
GO TO LA18;
LA15:
DO I=2 TO 4;
   IF NEXPER(I)>=10 THEN GO TO LA17;
END;
TEMPM(7)=0;
GO TO LA18;
LA17:
TEMPM(7)=-1;
LA18:
   /* EMOTION ROUTINE */
   /* UNPACK MEMORY LOCATIONS */
DO I=1 TO 7;
   K=ITEMP(I)/1000000;
   ITEMP(I)=ITEMP(I)-1000000*K;
   XFAMIL(I)=K;
   K=ITEMP(I)/10000;
   ITEMP(I)=ITEMP(I)/10000*K;
   XFEAR(I)=K;
   K=ITEMP(I)/100;
   XDESR(B)=ITEMP(I)-100*K;
XANGER(I) = K;
END;

/* COMPUTE PREVIOUS ATTITUDE -- FEAR */
DO I = 1 TO 7;
  WTMEM(I) = 1. - (1. / (1. + PARAMS(6) * XFAMIL(I)));
END;

/* WEIGHT ABSTRACTION LEVEL */
DO I = 2 TO 4;
  WTMEM(I) = WTMEM(I) * PARAMS(24);
  WTMEM(I+3) = WTMEM(I+3) * PARAMS(24) * PARAMS(24);
END;

/* ACCUMULATE */
SUMWT, SFEAR, AFEAR = 0.;
DO I = 1 TO 7;
  SFEAR = SFEAR + XFEAR(I) * WTMEM(I) / 100.;
  SUMWT = SUMWT + WTMEM(I);
END;
IF SUMWT = 0. THEN GO TO LA34;
AFEAR = SFEAR / SUMWT;

/* COMPUTE PREVIOUS ATTITUDE -- ANGER */
LA34:
DO I = 1 TO 7;
  WTMEM(I) = 1. - (1. / (1. + PARAMS(7) * XFAMIL(I)));
END;
DO I = 2 TO 4;
  WTMEM(I) = WTMEM(I) * PARAMS(25);
  WTMEM(I+3) = WTMEM(I+3) * PARAMS(25) * PARAMS(25);
END;
SUMWT, SANGER, AANGER = 0.;
DO I = 1 TO 7;
  SANGER = SANGER + XANGER(I) * WTMEM(I) / 100.;
  SUMWT = SUMWT + WTMEM(I);
END;
IF SUMWT = 0. THEN GO TO LA35;
AANGER = SANGER / SUMWT;

/* COMPUTE PREVIOUS ATTITUDE -- DESIRE */
LA35:
DO I = 1 TO 7;
  WTMEM(I) = 1. - (1. / (1. + PARAMS(8) * XFAMIL(I)));
END;
DO I = 2 TO 4;
  WTMEM(I) = WTMEM(I) * PARAMS(26);
  WTMEM(I+3) = WTMEM(I+3) * PARAMS(26) * PARAMS(26);
END;
SUMWT, SDESIR, ADESIR = 0.;
DO I = 1 TO 7;
  SDESIR = SDESIR + XDESIR(I) * WTMEM(I) / 100.;
  SUMWT = SUMWT + WTMEM(I);
END;
IF SUMWT=0. THEN GO TO LA36;
ADESIR=SDESIR/SUMWT;
/* COMBINE PREVIOUS ATTITUDE AND CURRENT MOOD */

LA36:
TEMPM (4)=PARAMS (3)*TEMPM (4)+(1.-PARAMS (3))*APEAR;
TEMPM (5)=PARAMS (4)*TEMPM (5)+(1.-PARAMS (4))*AANGER;
TEMPM (6)=PARAMS (5)*TEMPM (6)+(1.-PARAMS (5))*ADESIR;
/* EMOTIONS INTERACT, THE STRONGEST WEAKENS THE OTHERS */
TEMPM (3)=0;
IF (MIN((TEMPM(4)-TEMPM(5)),(TEMPM(4)-TEMPM(6)))<0.1)
    THEN GO TO LA31;
DEGR=TEMPM(4);
TEMPM(5)=TEMPM(5)*PARAMS(9);
TEMPM(6)=TEMPM(6)*PARAMS(10);
GO TO LA37;

LA31:
IF (MIN((TEMPM(5)-TEMPM(4)),(TEMPM(5)-TEMPM(6)))<0.1)
    THEN GO TO LA32;
DEGR=TEMPM(5);
TEMPM(4)=TEMPM(4)*PARAMS(11);
TEMPM(6)=TEMPM(6)*PARAMS(12);
GO TO LA37;

LA32:
IF (MIN((TEMPM(6)-TEMPM(4)),(TEMPM(6)-TEMPM(5)))<0.1)
    THEN GO TO LA33;
DEGR=TEMPM(6);
TEMPM(4)=TEMPM(4)*PARAMS(13);
TEMPM(5)=TEMPM(5)*PARAMS(14);
GO TO LA37;

LA33:
TEMPM(3)=1;
DEGR=MAX(TEMPM(4),TEMPM(5),TEMPM(6));

LA37:
IF (TEMPM(1)=3) I (TEMPM(1)=4) THEN GO TO LA1;
    /* ALDOUS ACTION ROUTINE */
IDEGR=0;
IF DEGR>=PARAMS(15) THEN GO TO LA41;
IACTN=5;
GO TO LA1;
    /* INDIFFERENCE */

LA41:
IF DEGR<=PARAMS(17) THEN GO TO LA42;
IACTN=6;
GO TO LA1;
    /* PARALYSIS */

LA42:
IF DEGR>=PARAMS(16) THEN GO TO LA43;
IDEGR=1;
GO TO LA44;
LA43:
IDEGR=2;

/* MILD */

LA44:
IF TEMPM(3)=1 THEN GO TO LA45;
IACNT=4;
GO TO LA1;

/* STRONG */

LA45:
IF DEGR=TEMPM(6) THEN GO TO LA46;
IACNT=1;
GO TO LA1;

/* CONFLICT */

LA46:
IF DEGR=TEMPM(4) THEN GO TO LA47;
IACNT=2;
GO TO LA1;

/* APPROACH */

LA47:
IACNT=3;

/* AVOIDANCE */

/* MAIN ROUTINE */

/* UPDATE IMMEDIATE MEMORY */

LA1:
TEMPM(2)=10000*NDIM1+100*NDIM2+NDIM3;

/* TEST IF SET FOR REPORT */

LA2:
IF TEMPM(1)>2 THEN GO TO LA4;
RETURN;

LA4:
/* ALDOUS VERBAL REPORT ROUTINE, USES LIMITD */
/* UNPACK TEMPORARY MEMORY */

K=TEMPM(2);
NDIM1=K/10000;
K=K-10000*NDIM1;
NDIM2=K/100;
NDIM3=K-100*NDIM2;

/* PRINT THE QUESTION */

IF TEMPM(1)=2 THEN GO TO LA67;
IR(1)=1;
GO TO LA70;

LA67:
IF TEMPM(1)=3 THEN GO TO LA68;
IR(1)=2;
GO TO LA69;

LA68:
IF TEMPM(1)=4 THEN GO TO LA77;
IR(1)=3;

/* SUBJECT IN PRESENT TENSE */
LA69:
IR(2)=6;
IR(5)=4;
IR(12)=5;
IR(15)=4;
GO TO LA71;

LA70:
IR(2)=3;
IR(5)=1;
IR(12)=2;
IR(15)=1;

LA71:
IF TEMP(M(7)=-1 THEN GO TO LA72;
IR(3)=LIMITD(TEMP(M(7)*PARAMS(2)+1.75);
IR(4)=6;
GO TO LA73;

LA72:
IR(3)=LIMITD(PARAMS(2)+2.);
IR(4)=5;

LA73:
IF TEMP(M(3)=-1 THEN GO TO LA75;
EMOMAX=MAX(TEMP(M(4),TEMP(M(6),TEMP(M(5)))*PARAMS(2);
IF EMOMAX<0.25 THEN GO TO LA75;
IF EMOMAX<=0.45 THEN GO TO LA74;
IR(13)=2;
GO TO LA76;

LA74:
IR(13)=1;
GO TO LA76;

LA75:
IR(13)=3;

LA76:
IR(14)=4;

LA77:
RETURN;
/* REACTION ROUTINE */

LA5:

```c
TEMPM(4) = XINJR * PARAMS(18) + TEMPM(4) * (1. - PARAMS(18));
TEMPM(5) = FucerR * PARAMS(19) + TEMPM(5) * (1. - PARAMS(19));
TEMPM(6) = SATISF * PARAMS(20) + TEMPM(6) * (1. - PARAMS(20));
TEMPM(4) = REGUL(TEMPM(4));
TEMPM(5) = REGUL(TEMPM(5));
TEMPM(6) = REGUL(TEMPM(6));
NDIM1 = TEMPM(2) / 10000;
K = TEMPM(2) - NDIM1 * 10000;
NDIM2 = K / 100;
NDIM3 = K - NDIM2 * 100;

/* LEARN ROUTINE */

IK = 2;
IL = 11;
CALL MEMORY (IK, NDIM1, NDIM2, NDIM3, ITEMP(1));
CALL MEMORY (IK, NDIM1, NDIM2, IL, ITEMP(2));
CALL MEMORY (IK, NDIM1, IL, NDIM3, ITEMP(3));
CALL MEMORY (IK, IL, NDIM2, NDIM3, ITEMP(4));
CALL MEMORY (IK, NDIM1, IL, IL, ITEMP(5));
CALL MEMORY (IK, IL, NDIM2, IL, ITEMP(6));
CALL MEMORY (IK, IL, IL, NDIM3, ITEMP(7));
DO I = 1 TO 7;
  K = ITEMP(I) / 1000000;
  ITEMP(I) = ITEMP(I) - 1000000 * K;
  ZFAMIL = K;
  K = ITEMP(I) / 10000;
  ITEMP(I) = ITEMP(I) - 10000 * K;
  ZFEAR = K;
  K = ITEMP(I) / 100;
  ZDESIR = ITEMP(I) - 100 * K;
  ZANGER = K;
  E = LOG10(ZFAMIL);
  WFEAR = 1. / ((10. * PARAMS(21)) ** E);
  WANGER = 1. / ((10. * PARAMS(22)) ** E);
  WDESIR = 1. / ((10. * PARAMS(23)) ** E);
  IFAHIL = IFESR = 100. * (ZFEAR / 100. + WFEAR * TEMPM(4)) / (1. + WFEAR);
  IANGER = 100. * (ZANGER / 100. + WANGER * TEMPM(5)) / (1. + WANGER);
  IDESIR = 100. * (ZDESIR / 100. + WDESIR * TEMPM(6)) / (1. + WDESIR);
  IFAHIL = ZFAMIL;
  ITEMP(I) = IDESIR + 100 * IANGER + 10000 * IFAHIL + 1000000 * IFAHIL;
END;
IK = 1;
CALL MEMORY (IK, NDIM1, NDIM2, NDIM3, ITEMP(1));
CALL MEMORY (IK, NDIM1, NDIM2, IL, ITEMP(2));
CALL MEMORY (IK, NDIM1, IL, NDIM3, ITEMP(3));
```

CALL MEMORY (IK, IL, NDIM2, NDIM3, ITEMP(4));
CALL MEMORY (IK, NDIM1, IL, IL, ITEMP(5));
CALL MEMORY (IK, IL, NDIM2, IL, ITEMP(6));
CALL MEMORY (IK, IL, IL, NDIM3, ITEMP(7));
GO TO LA2;

/* MFD */
/* RANDOMLY MODIFIES STIMULUS */
MFD: PROCEDURE (NDIMX) RETURNS (FIXED BINARY);
DCL NDIMX FIXED BINARY;
K=1;
IF NDIMX=1 THEN GO TO LM20;
IF NDIMX=10 THEN GO TO LM22;
GET FILE (RAN) EDIT (IFL) (F(2));
K=1;
IF IFL<=50 THEN GO TO LM20;
K=-1;
LM20:
NDIMX=NDIMX+K;
RETURN (NDIMX);
LM22:
NDIMX=NDIMX-1;
RETURN (NDIMX);
END MFD;

/* REGUL */
/* KEEPS INPUT IN RANGE 0.0 TO 0.99 */
REGUL: PROCEDURE (EMOT) RETURNS (FIXED BINARY (31,5));
DCL EMOT FIXED BINARY (31,5);
IF EMOT<0.0 THEN GO TO LR21;
IF EMOT>0.99 THEN GO TO LR23;
RETURN (EMOT);
LR21:
EMOT=0.;
RETURN (EMOT);
LR23:
EMOT=0.99;
RETURN (EMOT);
END REGUL;

/* LIMITD */
/* KEEPS INPUT IN RANGE 1 TO 5 */
LIMITD: PROCEDURE (DEGR) RETURNS (FIXED BINARY (15,5));
DCL DEGR FIXED BINARY (15,5);
IF DEGR<=5. THEN GO TO LL22;
DEGR=5.;
RETURN (DEGR);
LL22:
IF DEGR>=1. THEN GO TO LL24;
DEGR=1.;
RETURN (DEGR);
LL24:
RETURN (DEGR);
END LIMITD;
END ALDOUS;

/* MEMORY */
MEMORY: PROCEDURE (IK,AD1,AD2,AD3,ICON);
/* LIST-PROCESSING SECTION */
DCL (IDEE(0:500) FIXED BINARY STATIC,
MPERM(0:500) FIXED BINARY (31,0) STATIC,
POINT(0:500) FIXED BINARY STATIC,
PTR FIXED BINARY STATIC INIT (47),
KOUNT FIXED BINARY (15,5),
ICON FIXED BINARY (31,0),
(AD1,AD2,AD3) FIXED BINARY);
/* ZERO MEMORY */
IF IK=0 THEN DO;
MPERM=0;
IDEE=0;
POINT=-1;
GO TO DN;
END;
/* STORE IN MEMORY */
IF IK=1 THEN DO;
IDEN=10000*AD1+100*AD2+AD3;
IMOD=MOD(IDEN,47);
IF POINT(IMOD)=-1 THEN DO;
POINT(IMOD)=0;
MPERM(IMOD)=ICON;
IDEE(IMOD)=IDEN;
GO TO DN;
END;
IF IDEE(IMOD)=IDEN THEN DO;
MPERM(IMOD)=ICON;
GO TO DN;
END;
IF POINT(IMOD)=0 THEN DO;
MPERM(PTR)=ICON;
POINT(PTR)=0;
IDEE(PTR)=IDEN;
POINT(IMOD)=PTR;
PTR=PTR+1;
GO TO DN;
END;
N1=POINT(IMOD);
UP:
IF IDEE(N1)=IDEN THEN DO;
MPERM(N1) = ICON;
GO TO DN;
END;
IF POINT(N1) = 0 THEN DO;
MPERM(PTR) = ICON;
POINT(PTR) = 0;
IDEE(PTR) = IDEN;
POINT(N1) = PTR;
PTR = PTR + 1;
GO TO DN;
END;
N1 = POINT(N1);
GO TO UP;
END;
/* RETRIEVE FROM MEMORY */
IF IK = 2 THEN DO;
IDEN = 10000 * AD1 + 100 * AD2 + AD3;
IMOD = MOD(IDEN, 47);
IF POINT(IMOD) = -1 THEN DO;
ICON = 0;
GO TO DN;
END;
N1 = IMOD;
UP1:
IF IDEE(N1) = IDEN THEN DO;
ICON = MPERM(N1);
GO TO DN;
END;
IF POINT(N1) = 0 THEN DO;
ICON = 0;
GO TO DN;
END;
N1 = POINT(N1);
GO TO UP1;
END;
KOUNT = 0;
/* DUMP MEMORY */
IF IK = 3 THEN DO;
PUT FILE (SYSPRINT) EDIT ('ID', 'CONTENTS', 'POINTER')
(SKIP(2), X(4), A(2), X(5), A(8), X(3), A(7));
DO I = 0 TO 46;
IF POINT(I) = -1 THEN GO TO EMPTY;
KOUNT = KOUNT + 1;
PUT FILE (SYSPRINT) EDIT (IDEE(I), MPERM(I), POINT(I))
(SKIP, X(2), F(6), X(2), F(10), X(8), F(2));
IF POINT(I) = 0 THEN GO TO EMPTY;
N1 = POINT(I);
LAGO:
KOUNT = KOUNT + 1;
PUT FILE (SYSPRINT) EDIT (IDEE(N1), MPERM(N1),
    POINT(N1))
    (SKIP, X(2), F(6), X(2), F(10), X(8), F(2));
IF POINT(N1)=0 THEN GO TO EMPTY;
N1=POINT(N1);
GO TO LAGO;
EMPTY:
    END;
PUT FILE (SYSPRINT) EDIT
    (KOUNT,' OR ',
    ((KOUNT/500)*100), ' PER CENT UTILIZATION')
    (SKIP(2), F(4), A(4), F(6,2), A(21));
GO TO DN;
    END;
DN:
    RETURN;
    END MEMORY;
    END MAIN;
APPENDIX F

Flowchart of PL/I ALDOUS with Array and List-Processing Memories
APPENDIX G

User's Manual for Macro-Language Version of ALDOUS

ALDOUS (Loehlin, 1968) is a weighted additive model of impression formation which adheres to no specific theory. The program's primary utility is in testing specific impression formation hypotheses for which the parameters of attitude change are well defined. The computer program itself consists of three main sections: a monitor or environment, an impression formation routine and a memory routine.

Inputs are 7-number groups which correspond to stimuli for each trial. The seven numbers are considered in two groups: a three-number field which identifies each trial and addresses the program's memory, and a four-number section which describes the effect the stimulus or encounter can have on the model. In the original model, the identification numbers were limited to values 1 through 10 for a total of 1,000 (10 x 10 x 10) possible stimulus combinations. One option of the present program allows the identifications to have any positive integer value. The four remaining numbers of each stimulus indicate the influence of the encounter. The first specifies the affective power or "press" of the stimulus on a low to high scale of 1 to 9. The remaining numbers specify levels of injury, frustration and satisfaction on the same one to nine scale. Therefore, each stimulus has the following fixed format: IDENT1, IDENT2, IDENT3, POWER, INJURY, FRUSTRATION, and SATISFACTION. Zero values are
not allowed for any variable.

If the addressing scheme of the original model is retained, it is clear that even two-digit identification numbers could generate a maximum memory of $99 \times 99 \times 99$ or 970,299 possible combinations, a value beyond the capacity of most computers. However, impression formation variables with two-digit values are not an unreasonable assumption in most experiments; therefore, a more efficient, list-processing memory has been added to ALDOUS. This memory only generates storage locations for the stimuli actually presented. If 50 unique stimuli are presented, only 50 cells are required, regardless of their identification values.

Original ALDOUS can run two environments through its monitor in each program cycle. This capability allows for studies of attitude formation during sudden changes in environment and for the manipulation of varying exposure in two different environments, e.g., 400 trials in one type of stimuli and 100 trials in another. This capability has been retained in the present ALDOUS with the additional ability to study several two-environment experiments within the same computer run.
Control Cards

The model may be altered for each user and requires the following control cards:

1. Option card(s) which specify the type of ALDOUS to be executed. Only one set of option cards is required no matter how many environments are generated. The format and location of the option cards will be discussed later.

2. Data cards must be provided for each cycle of ALDOUS. A cycle is one run on the model where one or two environments share the same memory, parameters and input data. Data cards are placed as follows:

```c
//GO.SYSIN DD *
Title One
Title Two
Parameter Card
Powerk Card
Data cards with format (35 F(2)) or format specified in FORMAT option.
/*
```

The placement of all control cards in the program deck is illustrated in Figure 9.

Title One and Title Two are identifications printed before the output of each environment. They act as labels for each environmental output. Even if only one environment is specified, two title cards must be used. These cards may be blank or contain any combination of IBM System/360 characters in any position on the card.
Figure 9. Sample input deck for macro-language ALDOUS

The parameter card specifies values for 26 key variables punched in fixed format (26 P(3,2)). The card provided contains "normal" values for these variables. The range of possible values and their interpretations are explained in Figure 10.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>&quot;Normal&quot; Value</th>
<th>Extreme Low</th>
<th>High</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.06</td>
<td>.00</td>
<td>1.00</td>
<td>Per cent of misrecognition</td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>.20</td>
<td>5.00</td>
<td>Scale factor for extremity of response</td>
</tr>
<tr>
<td>3</td>
<td>.20</td>
<td>.00</td>
<td>4.00</td>
<td>Weight of previous attitude relative to mood (GT.1 means negative weight of previous attitude)</td>
</tr>
<tr>
<td>4</td>
<td>.20</td>
<td>.00</td>
<td>4.00</td>
<td>Same as 3</td>
</tr>
<tr>
<td>5</td>
<td>.20</td>
<td>.00</td>
<td>4.00</td>
<td>Same as 3</td>
</tr>
<tr>
<td>6</td>
<td>1.00</td>
<td>.00</td>
<td>1000</td>
<td>Weight of N in denominator of familiarity factor $1-1/(1-PAMK*PAMIL)$</td>
</tr>
<tr>
<td>7</td>
<td>1.00</td>
<td>.00</td>
<td>1000</td>
<td>Same as 6</td>
</tr>
<tr>
<td>8</td>
<td>1.00</td>
<td>.00</td>
<td>1000</td>
<td>Same as 6</td>
</tr>
<tr>
<td>9</td>
<td>.70</td>
<td>.00</td>
<td>1.00</td>
<td>Multiplier of second-named emotion in emotional interaction. (1=second emotion unaffected)</td>
</tr>
<tr>
<td>10</td>
<td>.30</td>
<td>.00</td>
<td>1.00</td>
<td>Same as 9</td>
</tr>
<tr>
<td>11</td>
<td>.70</td>
<td>.00</td>
<td>1.00</td>
<td>Same as 9</td>
</tr>
<tr>
<td>12</td>
<td>.30</td>
<td>.00</td>
<td>1.00</td>
<td>Same as 9</td>
</tr>
<tr>
<td>13</td>
<td>.30</td>
<td>.00</td>
<td>1.00</td>
<td>Same as 9</td>
</tr>
<tr>
<td>14</td>
<td>.30</td>
<td>.00</td>
<td>1.00</td>
<td>Same as 9</td>
</tr>
<tr>
<td>15</td>
<td>.25</td>
<td>.00</td>
<td>1.00</td>
<td>Level of shift from none to mild action</td>
</tr>
<tr>
<td>16</td>
<td>.55</td>
<td>.00</td>
<td>1.00</td>
<td>Level of shift from mild to strong action</td>
</tr>
<tr>
<td>17</td>
<td>.85</td>
<td>.00</td>
<td>1.00</td>
<td>Level of shift from strong action to paralysis</td>
</tr>
<tr>
<td>18</td>
<td>.50</td>
<td>.00</td>
<td>4.00</td>
<td>Weight of original experience relative to reaction (GT.1=negative weight of previous exposure)</td>
</tr>
<tr>
<td>19</td>
<td>.50</td>
<td>.00</td>
<td>4.00</td>
<td>Same as 18</td>
</tr>
<tr>
<td>20</td>
<td>.50</td>
<td>.00</td>
<td>4.00</td>
<td>Same as 18</td>
</tr>
<tr>
<td>21</td>
<td>.20</td>
<td>.10</td>
<td>10.00</td>
<td>Base of power in denominator $WT=1/10*STABLK)^{**EXPER}$</td>
</tr>
</tbody>
</table>

Figure 10. Values and descriptions of parameter card variables.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>&quot;Normal&quot; Value</th>
<th>Extreme Low</th>
<th>Extreme High</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>.20</td>
<td>.10</td>
<td>10.00</td>
<td>Same as 21</td>
</tr>
<tr>
<td>23</td>
<td>.20</td>
<td>.10</td>
<td>10.00</td>
<td>Same as 21</td>
</tr>
<tr>
<td>24</td>
<td>.10</td>
<td>.00</td>
<td>10.00</td>
<td>Multiplier for higher levels of abstraction</td>
</tr>
<tr>
<td>25</td>
<td>.10</td>
<td>.00</td>
<td>10.00</td>
<td>Same as 24</td>
</tr>
<tr>
<td>26</td>
<td>.10</td>
<td>.00</td>
<td>10.00</td>
<td>Same as 24</td>
</tr>
<tr>
<td>POWERK(1)</td>
<td>.25</td>
<td>.00</td>
<td>1.00</td>
<td>Level of power for defining consequences</td>
</tr>
<tr>
<td>POWERK(2)</td>
<td>.25</td>
<td>.00</td>
<td>1.00</td>
<td>Same as 24</td>
</tr>
<tr>
<td>POWERK(3)</td>
<td>.25</td>
<td>.00</td>
<td>1.00</td>
<td>Same as 24</td>
</tr>
</tbody>
</table>

Figure 10. (Continued)

The Powerk card contains three numbers punched in fixed format (3 F(3,2)). They specify the emotional levels necessary to switch the model from indifference to approach, approach to withdrawal, and withdrawal to paralysis respectively. Normal values for these cutoff points are given in Figure 10.

Data cards may be punched in format (35 F(2)) or in any other format specified by the FORMAT option. Since this option is repeated for each modeling cycle, data of varying formats can be run through different parameters in succeeding cycles. It should be noted that an incorrect FORMAT specification may lead to inaccurate results without an error message, particularly if blank (zero) columns are read as data. Most commonly, reading blank columns will lead to a SUBSCRIPTRANGE error in the ARRAY memory, but will pass without error message in the LIST memory.
Options

The exact nature of the ALDOUS model depends upon the values specified on the option card(s). These cards control the number of modeling cycles, the type of memory used in each cycle, the data format and other variables. They must be placed in the deck as follows:

//GO.OPTION DD *

Option cards

/*

Each option must be punched in the following format: OPTION= 'value' where the option name is followed by an equated value in quotation marks. Each option in a set is separated from the next by commas except the last which is followed by a semicolon. Since more than one version of ALDOUS may be constructed on any one modeling run, it is important to follow each set of options by one and only one semicolon. Any misspelled options, incorrect options or comments will be ignored if they do not contain semicolons. There is no limit on the number of cards used to describe any ALDOUS model.

The options and their values are described below. Any option which is not explicitly stated is assumed to have the underlined default value. Therefore, it is not necessary to specify any options. However a card with a semicolon must be used when all default options are acceptable.

ENV1='100'   ENV1 specifies the number of trials in the first
ENV2='0' ENV2 specifies the number of trials in the second environment using the same memory and parameters associated with the corresponding ENV1. If no second environment is required, ENV2 must be set to zero (the default). If ENV2 is set greater than 1,000 an error message is generated and the cycle is terminated.

ENV1 and ENV2 describe two environments sharing the same memory and parameter values. If the user wishes to compare two environments with different memory organizations, he should specify the PASS='NEW' option.

MEMORY='LIST','ARRAY' This option specifies a list-processing memory which creates unique memory locations for the stimuli presented. This option is therefore most efficient in terms of storage space. Any positive integer identification may be used provided it is accurately described in the FORMAT option, i.e., 10,000 cannot be read with an F(2) format. The ARRAY memory has the original 10 x 10 x 10 format. This option therefore limits the
range of inputs and can greatly increase the memory space required to execute the program. Identification values greater than 10 with the ARRAY option will raise the SUBSCRIPTRANGE condition and terminate the entire run. Therefore, when comparing ARRAY and LIST memories, no identification can exceed the limits imposed by the ARRAY option.

**PASS='OLD','NEW'** The PASS option allows the user to generate more than one version of ALDOUS in the same program run. When PASS='OLD', the current cycle is the last, regardless of subsequent option statements. If the NEW option is specified, an entirely new ALDOUS is created using new options. No limits are placed on the number of times the PASS='NEW' option may be used. Each invocation, however, requires a new set of input data, and each OPTION is reset to the default value unless respecified. The last cycle in any modeling run must specify PASS='OLD' (the default).

**REPORT='NO','YES'** If the REPORT option is activated (REPORT='YES'), a written report of each trial is printed. The interpretation of these reports is explained under verbal Reports. Except for special purposes, the YES option is not recommended
since a large amount of printed output can be
generated.

\texttt{FORMAT='(SKIP,35 F(2))'} The user may specify any data format.
The default is 35 two-digit numbers, or five
stimuli sets per card with the last ten columns
free for sequence numbers or other identifica-
tions. This option may be useful with a \texttt{LIST}
memory where identification fields of more than
digits are possible.

The options for any \texttt{ALDOUS} cycle may be specified in any
order, and in any location on any number of cards. The only re-
quirement is that for every modeling cycle, there be one semico-
lon to separate sets of options. For example, if a user wished
to compare 400 trials of the \texttt{ARRAY} and \texttt{LIST} versions of \texttt{ALDOUS},
the following control cards could be used.

\begin{verbatim}
ENV1='400', ENV2='0', MEMORY='LIST', PASS='NEW', REPORT='NO',
FORMAT='(SKIP,35 F(2))';
ENV1='400', ENV2='0', MEMORY='ARRAY', PASS='OLD', REPORT='NO',
FORMAT='(SKIP,35 F(2))';
\end{verbatim}

These options are exactly equivalent to the following:

\begin{verbatim}
ENV1='400', PASS='NEW'; ENV1='400', MEMORY='ARRAY'; where
default values are assumed.
\end{verbatim}

To run 500 trials in two environments and then 500 trials
in two more environments, each with default options, the control
cards could read:
To run 100 trials in an ARRAY memory, and then 100 trials in a LIST memory: MEMORY='ARRAY', PASS='NEW';

**Program Output**

The ALDOUS output consists of three parts: the optional report for each trial; the summary of each environment; and dump of the model's memory which is made after each modeling cycle. The report has the following format: the trial number, variable IDEGR, variable IACTN, 17-digit report, and three address numbers. The trial number is the count of the current trial. Variable IDEGR indicates the level of consequences of the current trial where a value of one is mild consequences, two is strong consequences and three is no consequences. Variable IACTN indicates the current action ALDOUS will take. IACTN has the following values: one for approach, two for avoidance, three for attack, four for conflict, five for indifference and six for paralysis. The 17 report digits represent a short form of a verbal report ALDOUS makes about its past, present or future impressions. The numbers correspond to questions for ALDOUS, and five word or phrase groups which are combined to construct the answers. The possible outputs are as follows: questions (QUES) 1=How do you feel about the last situation?, 2=How do you feel about the current situation?, 3=How do you feel about situation (IDENT1, IDENT2, IDENT3)?; subjects (SUBJ) 1=I was, 2=I felt, 3=It was, 4=I am, 5=I feel, 6=It is; adjectives (ADJ1)
122

1=not, 2=slightly, 3=fairly, 4=very, 5=extremely; adjectives
(ADJ2) 1=attracted, 2=afraid, 3=angry, 4=tense, 5=novel,
6=familiar; adjectives (ADJ3) 1=mild, 2=strong, 3=no; actions
(ACTN) 1=approach, 2=avoidance, 3=attack, 4=conflict,
5=indifference, 6=paralysis.

The 17 variables are divided among the outputs as follows:
QUES=variable 1; SUBJ=2, 5, 12, and 15; ADJ1=3, 6, 8, 10, and
16; ADJ2=4, 7, 9, 11, and 17; ADJ3=13; and ACTN=14. A typical
report and its interpretation might be as follows: 1 3 3 6 2 2
2 3 3 1 1 2 1 2 1 3 4

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Value</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>QUES</td>
<td>1</td>
<td>How do you feel about the last situation?</td>
</tr>
<tr>
<td>2</td>
<td>SUBJ</td>
<td>3</td>
<td>It was</td>
</tr>
<tr>
<td>3</td>
<td>ADJ1</td>
<td>3</td>
<td>fairly</td>
</tr>
<tr>
<td>4</td>
<td>ADJ2</td>
<td>6</td>
<td>familiar</td>
</tr>
<tr>
<td>5</td>
<td>SUBJ</td>
<td>2</td>
<td>I felt</td>
</tr>
<tr>
<td>6</td>
<td>ADJ1</td>
<td>2</td>
<td>slightly</td>
</tr>
<tr>
<td>7</td>
<td>ADJ2</td>
<td>2</td>
<td>afraid</td>
</tr>
<tr>
<td>8</td>
<td>ADJ1</td>
<td>3</td>
<td>fairly</td>
</tr>
<tr>
<td>9</td>
<td>ADJ2</td>
<td>3</td>
<td>angry, and</td>
</tr>
<tr>
<td>10</td>
<td>ADJ1</td>
<td>1</td>
<td>not</td>
</tr>
<tr>
<td>11</td>
<td>ADJ2</td>
<td>1</td>
<td>attracted</td>
</tr>
<tr>
<td>12</td>
<td>SUBJ</td>
<td>2</td>
<td>I felt</td>
</tr>
<tr>
<td>13</td>
<td>ADJ3</td>
<td>1</td>
<td>mild</td>
</tr>
<tr>
<td>14</td>
<td>ACTN</td>
<td>2</td>
<td>avoidance</td>
</tr>
<tr>
<td>15</td>
<td>SUBJ</td>
<td>1</td>
<td>I was</td>
</tr>
<tr>
<td>16</td>
<td>ADJ1</td>
<td>3</td>
<td>fairly</td>
</tr>
<tr>
<td>17</td>
<td>ADJ2</td>
<td>4</td>
<td>tense</td>
</tr>
</tbody>
</table>

Interpretation: How do you feel about the last situation? It was fairly familiar. I felt slightly afraid, fairly angry and not attracted. I felt mild avoidance. I was fairly tense. The current memory addresses apply only to question three: How do
The summary of each environment includes a one-number report of the action taken in each trial, the proportion of trials for each action, and the average fear, anger and attraction in the environment. Coding for the summary is as follows:
1 = indifference, 2 = weak conflict, 3 = strong conflict, 4 = weak approach, 5 = strong approach, 6 = weak avoidance, 7 = strong avoidance, 8 = weak attraction, 9 = strong attraction, 0 = paralysis.

The dump of ALDOUS' memory takes two forms. The LIST memory produces the contents of all unique cells and a count of the memory locations generated. The ARRAY memory dumps all 1,000 locations including the empty or unused cells. The ARRAY report also includes a count of the memory locations used and the percentage of total memory capacity utilized.