

ACCESS
A PROGRAM FOR THE CATALOG AND ACCESS OF INFORMATION

BY
J. GERRY PURDY

STAN-CS-71-210
MARCH, 1971

COMPUTER SCIENCE DEPARTMENT

School of Humanities and Sciences

STANFORD UNIVERSITY

ACCESS

A Program for the Catalog and Access of Information

by

J. Gerry Purdy

The preparation of this report has been supported by IBM Corporation.

1. Abstract

ACCESS is a program for the catalog and access of information. The program is primarily designed for and intended to handle a personal library, although larger applications are possible. ACCESS produces a listing of all entries by locator code (so one knows where to find the entry in his library), a listing of entry titles by user-specified category codes, and a keyword-in-context KWIC listing (each keyword specified by the user). ACCESS is presently programmed in FORTRAN and operates on any IBM System/360 under OS (it uses the IBM SORT/MERGE package). It is anticipated a machine language version (soon to be implemented) will greatly decrease the running time of the program.

2. Introduction

Numerous processes have been developed for cataloging and accessing information. The card catalog in any library is a typical example. When these become large and unwieldy, elaborate automated systems have been devised. For one's private library in his office, such automated systems are uneconomical and impractical. As a result, one is usually left to keep his own card catalog, e.g. on 3 x 5 cards in a file.

When one desires to access the information in a library (personal or public) concerning a particular subject, he has two problems:

(1) finding all entries in the library pertaining to that subject and
(2) actually locating the entries of interest. The latter process is the easier to solve. One simply creates a sequential ordering scheme for the file: the next entry to be inserted takes the next sequential assignment code, e.g. the set of integers or some modified code like Uxxxx, where "U" might stand for "unpublished reports" and "xxxx" for "the sequential entry in that file". The choice is arbitrary and can be varied according to the needs of the library.

The other problem -- that of locating all entries in a library pertinent to a particular subject -- quickly becomes quite complex to handle as the number of library entries increases. To create a subject-oriented index means one has to compose a "card" about the entry for each subject it is concerned with.

It becomes evident that there is a need to provide a simple, relatively economical system to provide cataloging and indexing for one's private library. The ACCESS program is an attempt to satisfy that need.

ACCESS takes almost free-form input card images and generates three kinds of reports:

1. ACCESSION LISTING: a listing of each entry in the library by locator (accession) code. This listing provides the master reference showing all input card images for each entry.
2. CATEGORY LISTING: for each category, a listing of each title card belonging to that category.

3. KEYWORD LISTING: a keyword-in-context (KWIC) listing of all keywords giving forward and backward text as additional context.

Examples of typical ACCESS output appear at the end of this report.

3. Input Specification

The ACCESS program may be run in one of two modes of operation: category mode and no-category mode. The category mode gives the full output capability of the program and requires the user to specify his categories. The no-category mode does not list title cards by categories and the codes are not input (see examples in Figures 1 and 2).

The inputs to ACCESS can be given in the following BNF-like notation:

```
*LIB <options>
*CATEGORY NAMES
    <category code> <category images>
*END CATEGORY NAMES
*ACCESSION ENTRIES
    <sets of accession entries>
*END ACCESSION ENTRIES
```

} Not input if
ACCESS run in
no-category mode.

where

<options> contains the options of the program as follows:

NOCAT means that the run is in the no-category mode of operation and category names and codes (above) are not input

NOLIST means the input cards will not be listed on the output stream.

(blank) means category mode and list inputs.

<category code> is the desired category code, specified as any single character available on the 029 keypunch (one of 64 characters) punched in card column 1.

<category images> are the desired category names, up to 79 characters long and punched in card columns 2 through 80.

<sets of accession entries> are the accession entries, each one composed of a title card and one or more continuation cards as necessary; these are described in more detail below.

Keywords

All keywords are specified by the user by placing an @ (8 - 4 punch) immediately before each desired keyword. During listings, all @ are replaced by blanks so that extra blanks between words are not required by the user. Every occurrence of an @ will cause a printed line to be generated in the keyword listing.

Accession Entries

Each accession entry is a description for an actual entry in the user's library, and each entry is composed of one or more 80 character fields. An accession entry is composed of a title card and, if necessary, continuation cards. These are defined as follows:

A. Title Card (First Card of Accession Entry)

1. The Locator Code -- This is the user supplied code which serves as a locator for the entry in the library. It may be up to eight (8) non-blank characters beginning in c.c. 1 (the first blank ends the code). One usually would place a locator character here followed by a sequence number. Some examples might be

U2001	(unpublished report 2001)
L70.029	(1970 letters file, letter no. 29)
20575	(just plain sequence 20,575)
100/555	(sub-report 555 of entry 100)

The user may employ this field for any code he feels will help him locate the actual library entry.

2. The Title -- This is the title of the accession entry and may begin anywhere after the first blank following the locator code. The title may be, itself, more than one card long. If categories are included, the first keyword should be preceded by a \$ (11-3-8 punch) instead of an @ ; this causes the ACCESS program to sort the category listing title cards by \$ entries.

It is also treated as an @ and placed in the keyword listing. (This first keyword is generally the author's name or the originating agency for the report; experience has shown it is preferable to treat author's names as keywords instead of making a separate author list.)

3. The Category Code(s) -- If categories are included, then the user may specify up to three one-character categories for this entry in c.c. 78-80. These one-character codes refer to the code given in the category name cards described above. If categories are not input, then the user may extend his title into columns 78-80.

B. Continuation Cards

1. The Continuation Code -- Each continuation card of the accession entry must begin with a + (12-6-8 punch) in c.c. 1.
2. The Continuation Entry -- The description for the library entry may be continued for up to 255 cards. Card columns 2-80 may be used, and the entry is free form. All @ cause keyword entry listings. These entries should be arranged so that the multiple card images will appear in a pleasing format when printed in the accession listing.

Sample ACCESS Inputs

With the inputs defined, some examples can be given. In Figure 1 at the end of the report an example of the category mode inputs are shown. The category codes appear in the title card for each entry. Note the different methods of locator code assignment.

In Figure 2, a sample of the ACCESS inputs are given where no categories are specified. All 80 columns of the title card may be utilized and a \$

is not necessary. The ACCESS entries correspond to the first few entries of the Computer Science Department Library's unpublished report series.

Sample ACCESS Output

When the inputs to ACCESS are merged with the necessary 360 JCL (see Appendix I), the program may be executed. A sample execution is given in Figure 3 for the full capability of the program, i.e. category mode of operation. Shown in Figure 3 are the accession listing, category listing of title cards, and the keyword listing. This output was generated using the input shown in Figure 1.

A sample run of the non-category mode is given in Figure 4. Here, the category listing is eliminated while the accession and keyword listings are similar to Figure 3. The output here was generated using the input shown in Figure 2.

4. Internal Processing and Organization

In this section we define the data structures employed by the ACCESS program and outline the overall operation of the program.

A. Data Structures

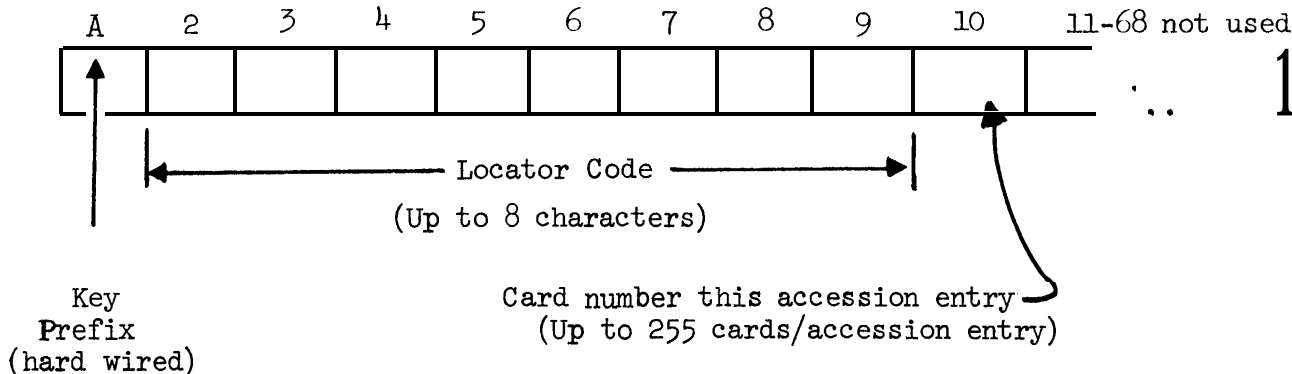
The basic idea behind processing each ACCESS entry is that all print line images for that entry are generated before the next entry is encountered. This is accomplished by the use of the following 200 character data structure, or record:



This structure employs the first 68 characters as a sort key and the remaining 132 characters as the printed line. In the following discussion, it is assumed that the full capability mode is being processed; if it is not, the program simply bypasses the category portion. Only the composition of the sort key is discussed, since the print line structure is easily seen in the output examples (Figures 3 and 4).

1. Accession Listing Sort Key

Of the 68 available characters (8-bit bytes) for the sort key, only the first 10 are utilized. The key takes the following format:



The accession listing is an output of all the ACCESS entries, sorted by locator code. The first character (key prefix) is hard-wired as an " A ".

The category entry _{sort} keys all begin with a " B ", and all keyword sort keys begin with a " C ". As a result, all accession records (key prefix equal to " A ") are sorted to the beginning, as desired.

Each accession entry is characterized by its unique locator code. Therefore, the locator code is placed in the sort key immediately after the key prefix (as shown above). Since there may be more than one card image for each entry, the eighth character in the sort key contains the binary equivalent of the card number. The remainder of the characters are not used in the accession listing sort key.

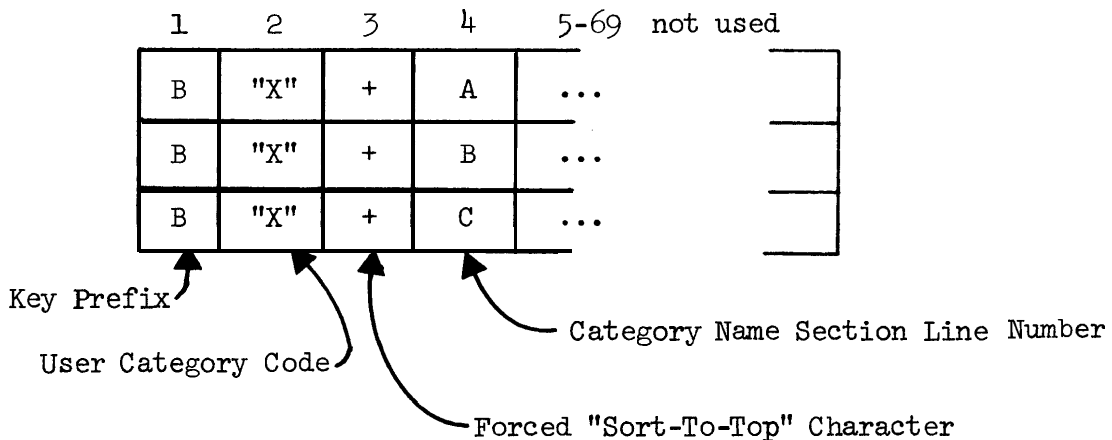
2. Category Listing Sort Key

For the category listing, there are actually two output segments for each category section: the category name segment and the ACCESS entry title card images segment. Each category name is to be set off (with blank lines) and appear at the beginning of each category section. Therefore, the sort key for the category names is different from the sort key for the title card images for that section.

The category name segment of each section is composed of three lines of the form:

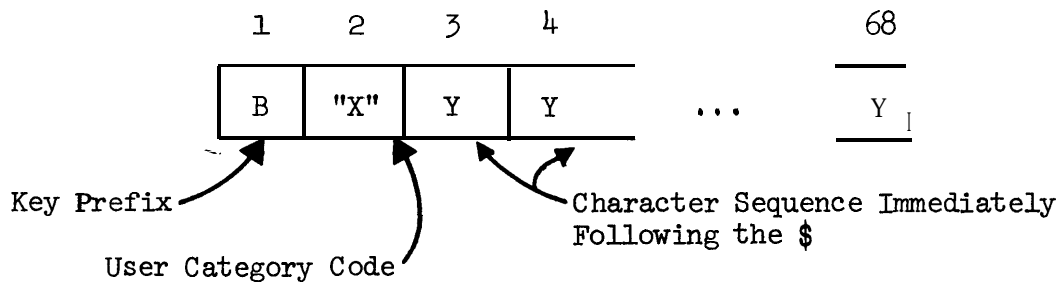
```
(blank)
*****CATEGORY -- X   YYYYYY...Y
(blank)
```

where X is the category code and YYY...Y is the category name. The sort key for these three lines are defined as follows:



Here, the key prefix is a B to indicate the category listing. The "X" is the user category code taken from c.c. 1 of the category names cards. The + (12-6-8 punch) is a character chosen to force the category name segment to the beginning of each category section. If this were not done, then it would be possible for some of the ACCESS entry title cards to sort before the category name. The A , B , C inserted in the fourth character simply serve to uniquely identify the order of the category name segment lines.

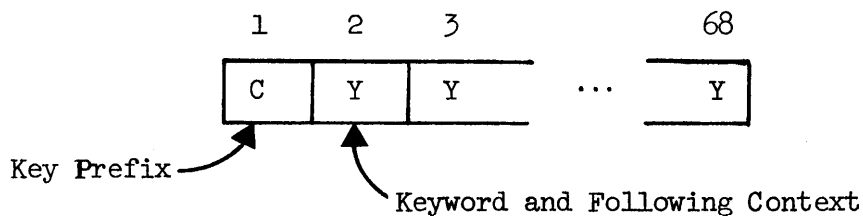
The ACCESS title card image segment contains the first card image in each ACCESS entry that has a category code in columns 78-80 of the first card. Thus, there can be up to three category title images generated for each ACCESS entry. The sort key for each category title image is defined as follows:



As above, the key prefix is a B to indicate a category listing line (or record), and the "X" is the user's category code taken from c.c. 78-80 of the title card in the ACCESS entry. The character sequence YYY...Y is taken from the title card immediately following the \$. If no \$ is present, the character sequence immediately following the locator code is used. The purpose of the YYY...Y sequence is to provide a sort sequence for the title card (usually an author's name, etc.).

3. Keyword Listing Sort Key

The sort key for the keyword listing is quite straightforward and is defined as:



The sort key prefix is a C to indicate the keyword listing. At each occurrence of an @ (a \$ is treated like an @), the character string following the @ is inserted in the sort key until either the sort key is full or the end of the ACCESS entry is reached. Thus, for each keyword (immediately preceded by an @), a keyword listing record is generated.

B. Overall Program Flow

The actual execution of the ACCESS program proceeds in three OS job steps. In the first job step, the input cards are read and a temporary file containing 200 character records (68 character sort key plus 132 character print line) is generated. The second job step (SORT/MERGE) takes the temporary file, sorts it according to the sort keys, and generates another temporary file of sorted 200 character records. The third job step lists the sorted file on the printer.

The processing flow for the first job step (the other two are rather straightforward) begins by reading in the ACCESS *LIB card. The program then decides if there are category names to be read and processed. If there are, they are read. For each category name card, three lines (200 character records) are output. Their sort keys correspond to the category name segment sort key as described above.

After this, the ACCESS entries are processed. The title card for the ACCESS is read, the locator code is stored for later use, and the output record for the accession listing is generated. Next, the category listing entry is composed (unless NOCAT was specified on the *LIB card). Finally, the title card image is placed into the first portion of a 2400 character keyword buffer. A buffer is necessary so that all keyword context may be output for a given keyword; the keyword output records are not generated until either the last card has been sensed or the keyword buffer is full (in which case all keyword records are generated and the buffer is cleared).

All continuation cards are read for the current ACCESS entry and its accession listing output record is made. The continuation card image text is appended to the current end of the keyword buffer. When a new ACCESS entry is read (card image not beginning with a +), the keyword records are composed and output, and the process begins over again.

To demonstrate this process, we can take a sample input deck and show what the keyword portion of the 200 character records look like both before and after the sort. Consider the ACCESS inputs shown in Figure 5. When these have been read in and processed by the first job step, the output records look like those shown in Figure 6. These records are then sorted, and the results are given in Figure 7. One can trace through the above described logic and defined data structures to confirm what is shown.

5. Conclusions

The ACCESS program has been described and sample input and output has been presented. The program is presently coded in FORTRAN IV for the IBM System 360/67 at Stanford. It is anticipated that another version of the program (in machine language) will be developed to decrease its running time.

The ACCESS program provides a free form input which allows one to catalog and subsequently retrieve all information in his private library.

Acknowledgment.

ACCESS is an outgrowth of a very similar, but less versatile, program which had been developed by Prof. D. E. Knuth of Stanford for his personal library.

Appendix I: Operating Instructions, Stanford IBM 360/67

In order to execute the ACCESS program on an IBM 360 under OS, necessary JCL must be specified. A procedure library entry has been made for the complete JCL. As a result, one only needs to generate the following JCL to execute the ACCESS program:

```
//jobname JOB XXXXX
//stepname EXEC ACCESS
//GO.SYSIN DD *
      (ACCESS INPUT CARDS)
/*
```

The JCL as shown above is satisfactory for runs with up to 5,000 ACCESS cards. If more than 5K input cards are anticipated, the user should change his JCL to provide more space for intermediate storage on the disc. See the consultants in the Computation Center for directions concerning how to do this.

Concerning timing estimates, the following sample running times have been recorded on the 360/67:

<u>No. ACCESS Input Cards</u>	<u>No. of Records Generated</u>	<u>Running Time (min.)</u>
52	430	0.76
78	480	0.85
~ 2 K	9.2 K	7.64
-2.5 K	12.5 K	8.50

Appendix II: How to Obtain ACCESS

Stanford Users

For those at Stanford, the ACCESS program has been compiled and placed into a private library and the JCL has been placed in a procedure library (discussed above). The source decks may be obtained from the file &T000.ACCESS1 (source for the first job step) and &T000.ACCESS2 (source for the third job step). These reside on SYS01.

Caution: If you use the source decks, be sure to supply your private library names.

Non-Stanford Users

For those not at Stanford, the program is available for a nominal charge. Those who desire to obtain a copy of the source deck may write:

User Services
Campus Facility
Stanford Computation Center
Stanford University
Stanford, California 94305

```

1. *LIB NCIST
2. *CATEGORY NAMES
3. A A P T I F I C I A L I N T E L L I G E N C E
4. C O M P I L E R S
5. F F W A R E
6. H H A R D W A R E . . . P E R I P H E R A L S
7. J J E P E G G I H C . . . T E S T I N G
8. L C O M P U T E R L A N G U A G E S
9. N N U M E R I C A L A N A L Y S I S
10. *END CATEGORY NAMES
11. *ACCESSION ENTRIES
12. M0001 $SLAGLE, J.R. AND $BURSKY, P. "EXPER. WITH A MULTI-PURPOSE A
13. + $THEOREM PROVING $HEURISTIC PROGRAM", JOUR. ACM, V C L15, N O .1,
14. + PAGES 85-89, 1968. DISCUSSES $MULTIPLE, A DYNAMIC $ORDERING, $BREADTH-FIRST
15. + SEARCHING ALGORITHM WHICH USES $ALPHA-BETA.
16. M0002 $WHITE, C.F., "$TIME-SHARING SERVICES", MODERN DATA, FEB. 1970,
17. + PG. 66. GIVES A TECHNOLOGY SURVEY OF T-S SERVICES.
18. M0003 $GUBENBERGER, FRED, "$PROGRAM $TESTING. AND $VALIDATING", J
19. + DATAMATION, JULY 1968. GIVES A GOOD SURVEY OF THE GOALS OF PROGRAM TESTING.
20. M0004 $SLAGLE, J.R. AND $DIXON, J.K., "$EXPERIMENTS WITH THE $M & $TREE- A
21. + $SEARCHING PROGRAM", CACM, VOL. 13, N O .3, MARCH 1970, P G .147-154.
22. + DISCUSSES A VARIANT $MINI-MAX METHOD WHICH AWARES $CONUSES F C R MULTIPLE
23. + SUCCESSOR MAX.'S AND GIVES A PENALTY FOR MULT. SUCCESSOR MIN.'S. THUS
24. + $NETAKES THE BRANCH WITH THE MOST LIKELIHOOD OF SUCCESS IN DEEPER
25. + PENETRATIONS. THE $ALPHA-BETA PROCEDURE ALSO HAD TO BE MODIFIED.
26. M0005 $FENICHEL, R.F., $WEIFENBAUM, J., AND $VOCHLSON, J.C., "$A PROGRAM FL
27. + TO TEACH $PROGRAMMING", CACM, VOL. 13, N O .3, MARCH 1970, P C .141-146.
28. + GAVE A SURVEY ABOUT THE $TEACH SYSTEM, USED TO A COURSE IN PROGRAMMING.
29. + THE STUDENTS LEARNED THE $UNCL LANGUAGE.
30. M0006 $SCHWAPCZ, R.M., $BURGER, J.F., AND $SIMMONS, R.F., "$ADJECTIVE AI
31. + $QUESTION-ANSWERER FOR $NATURAL LANGUAGE $INFERENCE", CACM, VOL. 13, N O .3,
32. + MARCH 1970, PG. 167-183. DISCUSSES THE $PROTOSYNTHESIS III.
33. M0107 $BRYAN, GLENN L., "$STUDENT-TO-STUDENT $INTERACTION IN $COMPUTER $TIME- E
34. + $SHAQFC SYSTEM", C&A, MARCH, 1970, P. 18.
35. M0008 $MORTON, KENT, "$THE VALUE OF THE $COMPUTER AS $A PUPIL", ARTICLE E
36. + $IN $EDUCATION IN MARCH, 1970, C & A , P .24
37. M0109 $FERGLSON, R.L., "$COMPUTER ASSISTANCE FOR INDIVIDIZED $INSTRUCTION", E
38. + ARTICLE $IN $EDUCATION IN MARCH, 1970, C&A, P .27.
39. M0010 $SCHWETMAN, H.C., "$L'S $E OF $REMOTE $CONSULE SYSTEM ON $UNIVERSITY EH
40. + CAMPS", ARTICLE $IN $EDUCATION IN MARCH, 1970, C&A, P. 30.
41. R0001 $DENDRAL: $BUCHANAN, B., $SUTHERLAND, G., $FEIGENBAUM, E.A., A
42. + "$HEURISTIC DENDRAL: A PROGRAM FOR GENERATING EXPLANATORY
43. + HYPOTHESES IN ORGANIC CHEMISTRY", MACHINE INTELLIGENCE 4
44. + EDINBURGH UNIVERSITY PRESS, 1969.
45. R0002 $GREENBLATT, R.C., ET. AL., "$THE GREENBLATT $CHESS PROGRAM", 4
46. + PROC. F J C C , 1967, PG. 901 TO 810.
47. R0003 $MICHIE, D., $FLEMING, J.G., $OLDFIELD, J.V., "$A COMPARISON OF A
48. + $HEURISTICS, $INTERACTIVE, AND UNAIDED METHODS FOR SOLVING $SHORTEST
49. + ROUTE PROBLEM", MACHINE INTELLIGENCE 3, EDINBURGH UNIV. PRESS, 1968.
50. R0004 $NEWELL, A., "$HEURISTIC $SEARCH: ILL-STRUCTURED PROBLEMS", PROGRESS 4
51. + $IN $OPERATIONS RESEARCH (VOL. 3), WILEY, 1969. DISCUSSES $HEURISTIC METHODS
52. + $IN $PROBLEM SOLVING VERY NICELY - GOOD SUMMARY.
53. R0005 $NILSSON, N.J., "$SEARCHING $PROBLEM SOLVING AND $GAME P L A Y I N $TREES A
54. + $F G R M A T M A L C C S T S C L U T I O N S", PROCEED. IFIP 68 CONGRESS, EDINBURGH, SCOTLAND,
55. + 1968. NOTE - MUCH OF THIS P A P E R IS ALSO I N $NILSSON'S BOOK.
56. + DISCUSSES $NILSSON'S $ARCH METHOD FOR $SEARCHING $AND-OR $TREES.
57. U0053 $SESAR, DENNIS, "$HARDWARE-SOFTWARE $EMULATOR FOR THE $IBM $360", H
58. + ENGR. 225B, SPRING 1968, PROF. G. $ESTRIN. USES $AN $XDS $SIGMA 7 $AS $EMULATOR.
59. U0054 $KNUTH, D.E. AND $FLOYD, R.W. "$NOTES ON AVOIDING $GO TO $STATEMENTS", L
60. + $STANFORD $CSD REPORT CS 144, JAN. 1970. CONSIDERS TRANSFORMING A "$GO TO"
61. + TYPE PROGRAM TO A $RECURSIVE PROCEDURE TYPE USING $ALGOL'S $WHILE, ETC.
62. U0055 $CFEATHAN, T.F. "$THE THEORY AND $CONSTRUCTION OF $COMPILERS", C
63. + $COMPUTER $ASSOCIATES, CA-6606-0111, JUNE 2 , 1966. HAS $GRAMMARS, $SYNTAX
64. + $ANALYSIS, $LEXICAL $ANALYSIS, $DEFINITIONAL $FACILITIES ($EXTENDABILITY) AND
65. + $OPTIMIZATION.
66. U0056 $FORSYTHE, G.F. "$PITFALLS IN $COMPUTATION, OR WHY $MATH $BOCK I $SN'T N
67. + $ENOUGH", $STANFORD $CSD REPORT CS 147, J A N , 1970. $NUMERICAL $ANALYSIS,
68. + $ROUND OFF, $TRUNCATION.
69. U0058 $CFEATHAN, T.F. "$THE THEORY AND $CONSTRUCTION OF $COMPILERS", 1967 C
70. + $R E P O R T F R O M $COMPUTER $ASSOCIATES ON $SYNTAX $ANALYSIS. N C N O . O R I N C E X .
71. *END ACCESSION ENTRIES
72.
73.
74.

```

Figure 1. Sample Category Mode ACCESS Input

*LIB NOCAT
*ACCESSION ENTRIES
000001 *AFUSO, C. *ANALOG COMPUTATION WITH RANDOM PULSE SEQUENCES. *ILLINOIS U.
*COMP. SCI. DEPT. RPT. 255. *1968.
000002 *AMERICAN STANDARDS ASS'N. *WORKING DOCUMENT *COROL
*SPECIFICATIONS *AMERICAN STANDARDS ASS'N. *1965.
000003 *ABRAHAMS, P. *LISP-2 LANGUAGE SPECIFICATIONS. *SYSTEM OF EVFL OPMENT
*CORP. *TM-3417/200/00. *1967.
000004 *APELKANS, M.Y. *DIFFERENCE SCHEMES FOR HYPERBOLIC EQUATIONS WITH
*DISCONTINUOUS INITIAL VALUES. *JIPPSALA U. COMP. SCI. DEPT. *RPT. 5. *1967.
000005 *ANSELONE, P.M. *CORE, R. *AN EXTENSION OF THE NEWTON-KANTOROVICH
*METHOD FOR SOLVING NONLINEAR EQUATIONS. *MICHIGAN STATE U. MATH. RES. CENTER. *MRC
*TECH. SUMMARY RPT. 520. *1965.
000006 *ANDERSON, N.S. *RCSENFELD, A. *SYMENSUN, N.F. *PATTERN-RECOGNITION: I. A
*COMPUTER PROGRAM FOR GENERATING SYNTHETIC PATTERNS. *MARYLAND U. COMP. SCI.
*CENTER. *TR-64-3. *1964.
000007 *APPLIED LOGIC CORP. *SEMI-ANNUAL RPT. OFF: CT RAI DECEMBER-AUTOMATED
*MATHEMATICAL *RANDOM *APPLIED LOGIC CORP. *1967.
000008 *ARBIB, M.A. *SOME THEORETICAL APPROACHES TO NERVOUS-SYSTEM FUNCTION.
*7M JUDY BEONETICS '35T. U.F.F. DEPT. *1967.
000009 *BALZER, R.M. *STUDIES CONCERNING MINIMAL TIME SOLUTIONS TO
*TRIPING-SQUAD SYNCHRONIZATION PROBLEM. *CARNEGIE INST. OF TECHNOLOGY. *
*1966.
000010 *BAECKER, P.M. *PLANAR REPRESENTATIONS OF COMPLEX GRAPHS. *M.I.T.
*LINCOLN LAB. *TN. 1967-1. *1967.
000011 *BARBIB, M.A. *GIVETON, Y. *CONTRIBUTIONS TO THE THEORY OF ALGEBRAS
*CONSIDERED AS AUTOMATA. *ST. U.F.E. DEPT. *1967.
000012 *BALUEY, A.N. *THREE PAPERS ON CHAPLYGIN-METHODS
*AND NONLINEAR EQUATIONS. *MARYLAND U. COMP. SCI. CENTER. *TR-67-41. *1967.
000013 *BARKINS, D.E. *THE THEORY OF IMPLEMENTATION OF SORT DIVISION. *ILLINOIS
*U. COMP. SCI. DEPT. *RPT. 230. *1967.
000014 *ARNOLD, L. *MONT H. *ASYMPTOTIC DISTRIBUTION OF THE EIGENVALUES OF
*RANDOM MATRICES. *MICHIGAN STATE U. MATH. RES. CENTER. *MRC TECH. SUMMARY RPT.
*736. *1967.
000015 *BARBIB, M.A. *KAMN, R.M. *A DEVELOPMENTAL MODEL
*FOR INFORMATION PROCESSING IN THE CHILD. *ST. U.F.E. DEPT. *1967.
000016 *BARBIB, M.A. *KAMN, R.M. *CYBERNETIC APPROACH TO MENTAL DEVELOPMENT. *
*ST. U.F.E. DEPT. *1967.
000017 *BARBIB, M.A. *NOTES ON A PARTIAL SURVEY OF CYBERNETICS IN EUROPE AND
*THE U.S. *R. *P. *E. *C. *RESEARCH ASSOCIATES, INC. *AFOSR-65-1412. *1965.
000018 *BARBIB, M.A. *SELF-REPRODUCING AUTOMATA: SOME IMPLICATIONS FOR
*THEORETICAL BIOLOGY. *ST. U.F.E. DEPT. *1967.
000019 *BARBIB, M.A. *GIVETON, Y. *ALGEBRAIC AUTOMATA II: THE CATEGORICAL
*FRAMEWORK FOR DYNAMIC ANALYSIS. *ST. U.F.E. DEPT. *1967.
000020 *BAYER, P. *ON ENDOMORPHISMS AND CONGRUENCE OF AUTOMATA. *BONNEN
*SCI. RES. LABS. *MATH. NOTE 497. *1967.
000021 *BAYER, R. *THE AUTOMORPHISM GROUP OF A STRONGLY CONNECTED AUTOMATON
*AND ITS QUOTIENT AUTOMATA. *ILLINOIS U. COMP. SCI. DEPT. *RPT. 199. *1965.
000022 *BAYER, R. *AUTOMORPHISM GROUPS AND QUOTIENTS OF STRONGLY
*CONNECTED AUTOMATA AND MONADIC ALGEBRAS. *ILLINOIS U. COMP. SCI. DEPT. *RPT.
*204. *1966.
*END ACCESSION ENTRIES

Figure 2. Sample No Category Mode Access Inputs

ACCESSION CODE	A C C E S S I O N E N T R Y	CATEGORY CODES
M0001	SLAGLE, J.P. AND BURSKY, P. "EXPER. WITH A MULTI-PURPOSE THEORY PROVING HEURISTIC PROGRAM", JCPP, ACM, VOL 19, NO. 1, PAGES 95-99, 1968. DISCUSSES MULTIPLE, A DYNAMIC ORDERING, BREADTH-FIRST SEARCHING ALGORITHM WHICH USES ALPHA-BETA.	A
M0002	WHITE, C.H., "THE SWAPPING SERVICES", MODERN DATA, FEB. 1970, P. 46.	J
M0003	GRUENBERGER, PROF. FRED, "PROGRAM TESTING AND VALIDATING", DATAVIATION, JULY 1968. GIVES A GOOD SURVEY OF THE GOALS OF PROGRAM TESTING.	A
M0004	SLAGLE J.P. AND DIXON, J. K., "EXPERIMENTS WITH THE M&N TREE - DISCUSSES A VARIANT MINI-MAX METHOD WHICH AWARDS BONUSES FOR MULTIPLE SUCCESSOR MAX.'S AND GIVE A PENALTY FOR MULT. SUCCESSOR MIN.'S. THIS GIVE TAKES THE BRANCH WITH THE MOST LIKELIHOOD OF SUCCESS IN DEEPER PENETRATIONS. THEIR ALPHA-BETA PROCEDURE ALS 'J' HAD TO BE MODIFIED.	E L
M0005	FENICHEL, R.R., WEITZENBAUM, J., AND DYCKELSON, J.C., "A PROGRAM TO TEACH PROGRAMMING", CACM, VOL. 13, NO. 3, MARCH 1970, PG. 141-146. GAVE A SURVEY ABOUT THE TEACH SYSTEM, USED TO A COURSE IN PROGRAMMING. THE STUDENTS LEARNED THE UNCL LANGUAGE.	A L
M0006	SCHWAPCZ, R.M., BURGER, J.F., AND SIMMONS, R.F., "4 DEDUCTIVE QUESTIONS ANSWERED BY A NATURAL LANGUAGE INFERENCE", CACM, VOL. 13, NO. 3, MARCH 1970, PG. 167-183. DISCUSSES THEIR PROTO SYNTHESIS II.	E
M0007	RYAN, GLENN L., "STUDENT INTERACTION IN COMPUTER TIME-SHARED SYSTEM", C&A, MARCH 1970, P. 18.	E
M0008	MORTON, KENT, "THE VALUE OF THE COMPUTER AS A PUPIL", ARTICLE ON EDUCATION IN MARCH, 1970, C&A, P. 24	E
M0009	FPPGUSON, P.L., "COMPUTER ASSISTANCE FOR INDIVIDIZED INSTRUCTION", ARTICLE ON EDUCATION IN MARCH, 1970, C&A, P. 27.	E
M0010	SCHWETMAN, H.D., "USE OF A REMOTE CONSOLE SYSTEM ON A UNIVERSITY CAMPUS", ARTICLE ON EDUCATION IN MARCH, 1970, C&A, P. 30.	E H
R0001	CENDRAL: RUCHMAN, B., SUTHERLAND, G., FEIGENBAUM, E.A., "HEURISTIC GENERAL: A PROGRAM FOR GENERATING EXPLANATORY HYPOTHESES IN ORGANIC CHEMISTRY", MACHINE INTELLIGENCE 4, EDINBURGH UNIVERSITY PRESS, 1969.	A
R0002	GREENBLATT, P. D., ET. AL. "THE GREENBLATT CHESS PROGRAM", PROC. FUCC, 1967, PG. 801 TO 810.	A
R0003	MICHIE, D., FLEMING, J.G., OLDFIELD, J.V., "A COMPARISON OF HEURISTICS, INTERACTIVE, AND UNAIDED METHODS FOR SOLVING A SHORTEST	A

Figure 3. Sample Category Mode ACCESS Output (Sheet 1 of 5)

ACCESSION CODE	AC C E S S I O N E N T R Y	CATEGORY CODES
00004	ROUTE PROBLEM", MACHINE INTELLIGENCE 3, EDINBURGH UNIV. PRESS, 1968. NEWELL, A. "HEURISTIC SEARCH: ILL-STRUCTURED PROBLEMS", PROGRESS IN OPERATIONS RESEARCH (VOL. 3), WILEY, 1969. DISCUSSES HEURISTIC METHODS IN PROBLEM SOLVING VERY NICELY - GOOD SUMMARY.	A
00005	NILSSON, N.-J., "SEARCHING PROBLEM SOLVING AND GAMEPLAYING TREES FOR MINIMAL COST SOLUTIONS", PROCEED. IJFP 69 CONGRESS, EDINBURGH, SCOTLAND, 1969. NOTE - MUCH OF THIS PAPER IS ALSO IN NILSSON'S BOOK. DISCUSSES NILSSON'S APPROX. METHOD FOR SEARCHING AND-OF TREES.	A
L0053	SESAR, DENNIS, "HARDWARE-SOFTWARE EMULATOR FOR THE IBM 360", ENGR. 2258, SPRING 1969, PROF. G. F. STRIN. USES AN XOSI GHA 7 AS EMULATOR.	F
L0054	KNUTH, C. E. AND FLOYD, R. W. "NOTES ON AVOIDING GO TO STATEMENTS", STANFORD CS REPORT CS 149, JAN. 1970. CONSIDERS TRANSFORMING A "GOTO" TYPE PROGRAM TO A RECURSIVE PROCEDURE TYPE USING ALGOL'S WHILE, ETC.	L
L0055	CHEATHAM, T. E. "THE THEORY AND CONSTRUCTION OF COMPILERS", COMPUTER ASSOCIATES, CC-5605-0111, JUNE 7, 1966. HAS GRAMMARS, SYNTAX ANALYSIS, LEXICAL ANALYSIS, DEFINITIONAL FACILITIES (EXTENSIBILITY) AND OPTIMIZATION.	C
L0056	FORSYTHE, G. E. "PITFALLS IN COMPUTATION, OR WHY A MATH BOOK ISN'T ENOUGH", STANFORD CS REPORT CS 147, JAN. 1970. NUMERICAL ANALYSIS, ROUND-OFF, TRUNCATION.	N
L0058	CHEATHAM, T. E. "THE THEORY AND CONSTRUCTION OF COMPILERS", 1967 REPORT FROM COMPUTER ASSOCIATES ON SYNTAX ANALYSIS. NCN C. OR IN CEX.	C

Figure 3. Sample Category Mode ACCESS Output (Sheet 2 of 5)

C A T E G O R Y L I S T I N G

ACCESS
CODE

***** CATEGORY -- 4 ...	A R T I F I C I A L I N T E L L I G E N C E	R0001 R0002 R0003 R0004 R0005 R0006 M0001 M0004
	DENDRAL: BUCHANAN, R., SUTHERLAND, G., FEIGENBAUM, E. A., GREENBLATT, R. D., ET AL., "THE GREENBLATT CHESS PROGRAM", WICHIE, D., FLEMING, J. G., CLDFIELD, J. V., "A COMPARISON OF NEWELL, A., "HEURISTIC SEARCH: ILL-STRUCTURED PROBLEMS", PROGRESS NILSSON, N. J., "SEARCHING PROBLEMSOLVING AND GAME PLAYING TREES SCHWARCZ, R. M., RIFGER, J. F., AND SIMMONS, R. F., "A DEDUCTIVE SLAGLE, J. R. AND BURSKEY, P., "EXPER. WITH A MULTI-PURPOSE SLAGLE, J. R. AND DIXON, J. K., "EXPERIMENTS WITH THE M&N TREE-	
***** CATEGORY -- C ...	C O M P I L E R S	U0058 U0058
	CHEATHAM, T. F., "THE THEORY AND CONSTRUCTION OF COMPILERS", CHEATHAM, T. F., "THE THEORY AND CONSTRUCTION OF COMPILERS", 1967	
***** CATEGORY -- E ...	E D U C A T I O N	M0007 M0005 M0009 M0008 M0010
	BRYAN, GLENN L., "STUDENT-TO-STUDENT INTERACTION IN COMPUTER TIME-FENCHEL, R. R., WEIZENBAUM, J., AND YOCHELSON, J. C., "A PROGRAM FERGUSON, R. L., "COMPUTER ASSISTANCE FOR INDIVIDUALIZED INSTRUCTION", MORTON, KENT, "THE VALUE OF THE COMPUTER AS A PUPIL", ARTICLE SCHWETMAN, H. D., "USE OF A REMOTE CONSOLE SYSTEM ON A UNIVERSITY	
***** CATEGORY -- H ...	H A R D W A R E . . . P E R I P H E R A L S	M0010 U0053
	SCHWETMAN, Y. O., "USE OF A REMOTE CONSOLE SYSTEM ON A UNIVERSITY SESAR, DENNIS, "A HARDWARE-SOFTWARE EMULATOR FOR THE IBM 360".	
***** CATEGORY -- J ...	D E B U G G I N G . . . T E S T I N G	M0003
	GRUENBERGER, PROF. FRED, "PROGRAM TESTING AND VALIDATING",	
***** CATEGORY -- L ...	C O M P U T E R L A N G U A G E S	M0005 U0054 M0006
	FENCHEL, R. R., WEIZENBAUM, J., AND YOCHELSON, J. C., "A PROGRAM KNUTH, D. E. AND FLOYD, R. M., "NOTES ON AVOIDING GO TO STATEMENTS", SCHWARCZ, R. M., RIFGER, J. F., AND SIMMONS, R. F., "A DEDUCTIVE	
***** CATEGORY -- h ...	N U M E R I C A L A N A L Y S I S	U0056
	FOR SYTHE, G. E., "PITFALLS IN COMPUTATION, OR WHY A MATH BOOK ISN'T	

Figure 3. Sample Category Mode ACCESS Output (Sheet 3 of 5)

KEYWORD LISTING

U0054	"GOTO" TYPE PROGRAM TO A RECURSIVE PROCEDURE TYPE USING ALGOL'S WHILE-DO	WHILE-DO	U0054
U0004	STYLING-CODES UCCES INCEPER PENETRATIONS. THEIR ALPHA-BETA PROCEDURE ALSO	ALPHA-BETA PROCEDURE ALSO	U0004
U0001	BREADING, BREADTH-FIRST SEARCHING ALGORITHM WHICH USES ALPHA-BETA.	BREADING, BREADTH-FIRST SEARCHING ALGORITHM WHICH USES ALPHA-BETA.	U0001
U0005	*BOOK DISCUSSES NILSSON'S SEARCH METHOD FOR SEARCHING AN-O-TREES.	DISCUSSES NILSSON'S SEARCH METHOD FOR SEARCHING AN-O-TREES.	U0005
U0001	FS 85-39, 1948. DISCUSSES MULTIPLE, A DYNAMIC ORDERING, BREADTH-FIRST SEARCHING ALGORITHM WHICH USES ALPHA-BETA.	FS 85-39, 1948. DISCUSSES MULTIPLE, A DYNAMIC ORDERING, BREADTH-FIRST SEARCHING ALGORITHM WHICH USES ALPHA-BETA.	U0001
U0001	BYAN, GLEN L., "STUDENT-TO-STUDENT INTERACTION IN C++"	BYAN, GLEN L., "STUDENT-TO-STUDENT INTERACTION IN C++"	U0001
U0001	BUCHANAN, R., SUTHERLAND, G., FEIGENBAUM, E.A., "HEURISTIC SEARCHING FOR THE SOLUTION OF THE N-QUEENS PROBLEM"	BUCHANAN, R., SUTHERLAND, G., FEIGENBAUM, E.A., "HEURISTIC SEARCHING FOR THE SOLUTION OF THE N-QUEENS PROBLEM"	U0001
U0001	BURGER, J.F., AND SIMMONS, R.F., "A DEDUCTIVE QUEST"	BURGER, J.F., AND SIMMONS, R.F., "A DEDUCTIVE QUEST"	U0001
U0001	BURSKY, P., "EXPER. WITH A MULTI-PURPOSE THEOREM PROV."	BURSKY, P., "EXPER. WITH A MULTI-PURPOSE THEOREM PROV."	U0001
U0055	CHEATHAM, T.E., "THE THEORY AND CONSTRUCTION OF COMPILER" U0055P	CHEATHAM, T.E., "THE THEORY AND CONSTRUCTION OF COMPILER" U0055P	U0055
U0059	CHES PROGRAM", PROC. FJCC, 1967, P. 801 TO 810.	CHES PROGRAM", PROC. FJCC, 1967, P. 801 TO 810.	U0059
U0002	COMPILES", COMPUTER ASSOCIATES, CA-6606-0111, JUNE 2, 1967.	COMPILES", COMPUTER ASSOCIATES, CA-6606-0111, JUNE 2, 1967.	U0002
U0058	COMPILERS", 1967. QCPRT FROM CCCPUTEQ ASSOCIATES ON S	COMPILERS", 1967. QCPRT FROM CCCPUTEQ ASSOCIATES ON S	U0058
U0056	COMPUTATION, OR WHY A MATH BOOK ISN'T ENOUGH", STANFORD UNIVERSITY	COMPUTATION, OR WHY A MATH BOOK ISN'T ENOUGH", STANFORD UNIVERSITY	U0056
U0010	CONSOLE SYSTEM ON A UNIVERSITY CAMPUS", ART ICL EDINB.	CONSOLE SYSTEM ON A UNIVERSITY CAMPUS", ART ICL EDINB.	U0010
U0006	DEDUCTIVE QUESTION-ANSWERER FOR NATURAL LANGUAGE INF.	DEDUCTIVE QUESTION-ANSWERER FOR NATURAL LANGUAGE INF.	U0006
U0055	DEFINITIONAL FACILITIES (EXTENDABILITY) AND OPTIMIZ.	DEFINITIONAL FACILITIES (EXTENDABILITY) AND OPTIMIZ.	U0055
U0001	DENDRAL: BUCHANAN, R., SUTHERLAND, G., FEIGENBAUM, E.	DENDRAL: BUCHANAN, R., SUTHERLAND, G., FEIGENBAUM, E.	U0001
U0004	DIXON, J.K., "EXPERIMENTS WITH THE M&N TREE" SEACC	DIXON, J.K., "EXPERIMENTS WITH THE M&N TREE" SEACC	U0004
U0006	EDUCATION IN MARCH, 1973, C&A, P. 24	EDUCATION IN MARCH, 1973, C&A, P. 24	U0006
U0009	EDUCATION IN MARCH, 1970, C & A, P. 30.	EDUCATION IN MARCH, 1970, C & A, P. 30.	U0009
U0053	EMULATOR FOR THE IBM 360", ENGR. 2258, SPRING 1968, P.	EMULATOR FOR THE IBM 360", ENGR. 2258, SPRING 1968, P.	U0053
U0055	EMULOR, USES AN XDS SIGMA 7 AS EMULATOR.	EMULOR, USES AN XDS SIGMA 7 AS EMULATOR.	U0055
U0055	EXTENDABILITY) AND OPTIMIZATION.	EXTENDABILITY) AND OPTIMIZATION.	U0055
U0001	FEIGENBAUM, E.A., "HEURISTIC GENERAL: A PROGRAM FOR	FEIGENBAUM, E.A., "HEURISTIC GENERAL: A PROGRAM FOR	U0001
U0005	FENICHEL, O.R., WEIZENBAUM, J., AND YOCHESON, J.C.	FENICHEL, O.R., WEIZENBAUM, J., AND YOCHESON, J.C.	U0005
U0009	FERGUSON, R.L., "COMPUTER ASSISTANCE FOR INDIVIDUAL ZED I	FERGUSON, R.L., "COMPUTER ASSISTANCE FOR INDIVIDUAL ZED I	U0009
U0054	FLOYD, R.W., "VOTES ON AVOIDING GO TO STATEMENTS", STA	FLOYD, R.W., "VOTES ON AVOIDING GO TO STATEMENTS", STA	U0054
U0056	FORSYTH, G.E., "PITFALLS IN COMPUTATION, OR WHY A MATH	FORSYTH, G.E., "PITFALLS IN COMPUTATION, OR WHY A MATH	U0056
U0005	GAME PLAYING TREES FOR MINI-HAL CCST SOLUTIONS", PROCE	GAME PLAYING TREES FOR MINI-HAL CCST SOLUTIONS", PROCE	U0005
U0055	GRAMMARS, SYNTAX ANALYSIS, LEXICAL ANALYSIS, DEFINIT	GRAMMARS, SYNTAX ANALYSIS, LEXICAL ANALYSIS, DEFINIT	U0055
U0054	GREENBLATT, R.D., ET AL., "THE GREENBLATT CHESS PROGR	GREENBLATT, R.D., ET AL., "THE GREENBLATT CHESS PROGR	U0054
U0002	GRUENBERGER, PPOF. FRED., "PROGRAM TESTING AND VALIDATI	GRUENBERGER, PPOF. FRED., "PROGRAM TESTING AND VALIDATI	U0002
U0053	HARDWARE-SOFTWARE EMULATOR FOR THE IBM 360", ENGR. 22	HARDWARE-SOFTWARE EMULATOR FOR THE IBM 360", ENGR. 22	U0053
U0001	HEURISTIC DENDRAL: A PROGRAM FOR GENERATING EXPLANATOR	HEURISTIC DENDRAL: A PROGRAM FOR GENERATING EXPLANATOR	U0001
U0001	HEURISTIC PROGRAM", JCP, ACM, VOL 15, NO 1, PAGE 88	HEURISTIC PROGRAM", JCP, ACM, VOL 15, NO 1, PAGE 88	U0001
U0004	HEURISTIC SEARCH: ILL-STRUCTURED PROBLEMS", PROGRESS	HEURISTIC SEARCH: ILL-STRUCTURED PROBLEMS", PROGRESS	U0004
U0003	HEURISTICS, INTERACTIVE, AND UNAIID VETHOOS FOR SOLVI	HEURISTICS, INTERACTIVE, AND UNAIID VETHOOS FOR SOLVI	U0003
U0053	IBM 360", ENGR. 2258, SPRING 1968, POCF. G. ESTRIN.	IBM 360", ENGR. 2258, SPRING 1968, POCF. G. ESTRIN.	U0053
U0006	INFERENCE, CACM, VOL 13, NO 3, MARCH 1970, PG. 167-	INFERENCE, CACM, VOL 13, NO 3, MARCH 1970, PG. 167-	U0006
U0009	INSTRUCTION", ARTICLE ON EDUCATION IN MARCH, 1970, C&	INSTRUCTION", ARTICLE ON EDUCATION IN MARCH, 1970, C&	U0009
U0003	INTERACTIVE, AND UNAIID METHODS FOR SOLVING A SHORTE	INTERACTIVE, AND UNAIID METHODS FOR SOLVING A SHORTE	U0003
U0007	INTERACTION IN COMPUTER TIME- SHAPREC SYSTEM, C&A, MAR	INTERACTION IN COMPUTER TIME- SHAPREC SYSTEM, C&A, MAR	U0007
U0054	KNUTH, D.E. AND FLOYD, R.W., "NOTES ON AVOIDING GO TO S	KNUTH, D.E. AND FLOYD, R.W., "NOTES ON AVOIDING GO TO S	U0054
U0055	LEXICAL ANALYSIS: DEFINITIONAL FACILITIES (EXTENDABIL	LEXICAL ANALYSIS: DEFINITIONAL FACILITIES (EXTENDABIL	U0055
U0004	"M&N TREE- SEARCHING PROGRAM", CACM, VOL. 13, NO. 3	"M&N TREE- SEARCHING PROGRAM", CACM, VOL. 13, NO. 3	U0004
U0056	MATHBOOK ISN'T ENOUGH", STANFORD CSD FEPOPT CS 147.	MATHBOOK ISN'T ENOUGH", STANFORD CSD FEPOPT CS 147.	U0056

Figure 3. Sample Category Mode ACCESS Output (Sheet 4 of 5)

KEYWORD LISTING

00003 m 6
 00004 1970, 3, "ARCHITECTURE OF THE MULTIPLE SIC", "A COMPACT
 00005 MORTON, KENT, "THE VALUE OF THE COMPUTER AS A PUPIL",
 00006 MULTIPLE, A DYNAMIC ORDERING, BREACT-FIRST SEARCHING,
 00007 NATURAL LANGUAGE INFERENCE", CACM, VOL. 13, NO. 3, MAR
 00008 NEWELL, A., "HEURISTIC SEARCH: ILL-STRUCTURED PROBLEMS
 00009 NILSSON, N. J., "SEARCHING PROBLEMS SOLVING AND GAME PL
 00010 NUMERICAL ANALYSIS, ROUND OFF, TRUNCATION,
 00011 OPTIMIZATION,
 00012 ORDERING, READING-FIRST SEARCHING ALGORITHM WHICH USE
 00013 PROBLEMS SOLVING AND GAME PLAYING TREES FOR MINIMAL CO
 00014 PROGRAMMING", CACM, VOL. 13, NO. 3, MARCH 1970, P. 14
 00015 PUPIL", ARTICLE ON EDUCATION IN MARCH, 1970, C&A, P. 2
 00016 QUESTION-ANSWERER FOR NATURAL LANGUAGE INFERENCE", CAC
 00017 REMOTE CONSOLE SYSTEM ON A UNIVERSITY CAMPUS", ARTIC
 00018 ROUND OFF, TRUNCATION,
 00019 SCHWARTZ, R. M., BURGER, J. F., AND SIMMONS, R. F., "A DE
 00020 SCHWARTZ, R. M., BURGER, J. F., AND SIMMONS, R. F., "A DE
 00021 SCHWARTZ, R. M., BURGER, J. F., AND SIMMONS, R. F., "A DE
 00022 SEARCH: ILL-STRUCTURED PROBLEMS", PROGRESS IN OPERATI
 00023 SEARCHING PROGRAM", CACM, VOL. 13, NO. 3, MARCH 1970,
 00024 SEARCHING PROBLEM SOLVING AND GAME PLAYING TREES FOR
 00025 SESAR, DENNIS, "A HARDWARE-SOFTWARE EMULATOR FOR THE I
 00026 SHORTEST ROUTE PROBLEM", MACHINE INTELLIGENCE 3, EDI N
 00027 SIGMA 7 AS EMULATOR.
 00028 SIMMONS, R. F., "A DEDUCTIVE QUESTION-ANSWERER FOR NA
 00029 SLAGLE, J. R. AND BURSKEY, P., "EXPER. WITH A MULTI-PURP
 00030 SLAGLE, J. R. AND DIXON, J. K., "EXPERIMENTS WITH THE M
 00031 SUTHERLAND, G., FEIGENBAUM, E. A., "HEURISTIC DEMONSTR
 00032 SYNTAX ANALYSIS, NO NC. OR INDEX.
 00033 TEACH SYSTEM USED TO A CCUPSE IN PROGRAMMING. THE ST
 00034 TESTING AND VALIDATING", DATA JOURNAL, JULY 1968, GIVE
 00035 THEOREM PROVING HEURISTIC PROGRAM", JOUR. ACM, VOL. 15,
 00036 TIME-SHARING SERVICES", C&A, VOL. 13, NO. 3, MARC
 00037 TIME-SHARING SERVICES", MODERN DATA, FEB. 1970, PG. 66
 00038 TREES FOR MINIMAL COST SOLUTIONS", PROC. JFIP 68 C O
 00039 TRUNCATION,
 00040 UNCL LANGUAGE.
 00041 UNIVERSITY CAMPUS", ARTICLE ON EDUCATION IN MARCH 19
 00042 VALIDATING", DATA JOURNAL, JULY 1968. GIVES A GOOD SUP
 00043 WEIZENBAUM, J., AND YOCHESON, J. C., "A PROGRAM TO TE
 00044 WHITE, C. H., "TIME-SHARING SERVICES", MODERN DATA, FE
 00045 XDS SIGMA 7 AS EMULATOR.
 00046 YOCHESON, J. C., "A PROGRAM TO TEACH PROGRAMMING", CA
 00047 ENGR. 225B, SPRING 1968, PROF. G. F. STRIN. USES
 00048 XDS SIGMA 7 AS EMULATOR.
 00049 3 IS CUISSE A VARIANT
 00050 PAGE 85-89, 1968. DISCUSSES
 00051 QUESTION-ANSWERER FOR
 00052 STANFORD CS REPORT CS147, JAN. 1970.
 00053 EXTENSIBILITY) AND
 00054 DISCUSSES MULTIPLE, A DYNAMIC
 00055 SEARCHING
 00056 TO TEACH
 00057 DISCUSSES THEIR
 00058 A
 00059 A DEDUCTIVE
 00060 USE OF A
 00061 NUMERICAL ANALYSIS,
 00062
 00063 HEURISTIC
 00064 WITH THE M & N TREE-
 00065
 00066 UNAIRED METHODS FOR SOLVING A
 00067 USES AN XDS
 00068 SCHWARTZ, R. M., BURGER, J. F., AND
 00069
 00070 BRYAN, GLENN L.,
 00071 DENRAL,
 00072 H. A. S. GRAMMAR,
 00073 1967 REPORT FROM COMPUTER ASSOCIATES C
 00074 GAVE A SURVEY ABOUT THE
 00075 GLENNBERGER, PROF. F. FRED, "PROGRAM
 00076 WITH 4 MULTI-PURPOSE
 00077 WHITE, C. H.,
 00078 EXPERIMENTS WITH THE M & N
 00079 SEARCHING PROBLEMS SOLVING AND GAME PLAYING
 00080 DISCUSSES NILSSON'S APPROX METHOD FOR SEARCHING AND-OR
 00081 REPORT CS 147, JAN. 1970. NUMERICAL ANALYSIS, ROUND OFF,
 00082 THE STUDENTS LEARNED THE
 00083 USE C. F. A. REMOTE CONSOLE SYSTEM ON A
 00084 GRUENBERGER, PROF. FRED, "PROGRAM TESTING AND
 00085 FENICHEL, R. R.,
 00086 ENGR. 225A, SPRING 1969, PROF. G. F. STRIN. USES AN
 00087 FENICHEL, R. R., WEIZENBAUM, J., AND N
 00088 "A HARDWARE-SOFTWARE EMULATOR FOR THE IBM 360",
 00089 ENGR. 225B, SPRING 1969, PROF. G. F. STRIN. USES AN XDS SIGMA 7 AS EMULATOR.

Figure 3. Sample Category Mode ACCESS Output (Sheet 5 of 5)

ACCESSION CODE	A C C E S S I O N E N T R Y	CATEGORY CODES
CC0001	APLSC. ANALOG COMPUTATION WITH RANDOM PULSE SEQUENCES.* ILLINOIS U. COMP.SCI.DEPT.*RPT. 255.*1963.	
CC0002	AMERICAN STANDARDS ASS'N.*WORKING DOCUMENT COPOL SPECIFICATIONS.* AMERICAN STANDARDS ASS'N.* 1965.	
CC0003	ABRAMS, P. * LISP-? LANGUAGE SPECIFICATIONS.* SYSTEM DEVELOPMENT CENT. *TR-3417/200/00.*1967.	
CC0004	APFKEMAS, M.V.*ON DIFFERENCE SCHEMES FOR HYPERBOLIC-EQUATIONS WITH CT. CONTINUOUS INITIAL-VALUES.* UPSALA U. COMP.SCI.DEPT.*RPT.5.*1967.	
CC0005	ASSELINE, P.W. MOORE, R.H.*AN EXTENSION OF THE NEWTON-KANTOROVICH METHOD FOR SOLVING NONLINEAR-EQUATIONS.* WISCONSIN U. MATH.RES. CENTER.*MRC TECH. SUMMAR Y RPT. 520.*1965.	
CC0006	ANDERSON, N.S. ROSENFELD, A. SIMENSON, N.F.*PATTERN-RECOGNITION: I. A COMPUTER PROGRAM FOR GENERATING SYNTHETIC PATTERNS.* MARYLAND U. COMP.SCI. CENTER * TR-64-9.*1964.	
CC0007	APPLIED LOGIC CORP.*SEMI-ANNUAL REPORT: CTR-AIDED SEMI-AUTOMATED MATHEMATICS.* SACDOL* APPLIED LOGIC CORP.* 1967.	
CC0008	APPEL, M.A.*SOME THEORETICAL APPROACHES TO NERVOUS-SYSTEM FUNCTION.* NEUROCYBERNETICS* U. E.E.DEPT.*1967.	
CC0009	BALZER, R.M.*STUDIES CONCERNING MINIMAL TIME SOLUTIONS TO THE FIRING-SQUAD SYNCHRONIZATION PROBLEM.* CARNEGIE INST. OF TECHNOLOGY.* 1966.	
CC0010	BARCKER, S.M. *PLANAR-REPRESENTATIONS OF COMPLEX GRAPHS.* M.I.T. LINCOLN LAB.*TR. 1967-1*1967.	
CC0011	BARIN, M.A. GIVON, Y.*CONTRIBUTIONS TO THE THEORY OF ALGEBRAS CONSIDERED AS AUTOMATA.* ST. U. E.E.DEPT.*1967.	
CC0012	BALLET, A.A.*THREE PAPERS ON CHAPLYGIN-METHODS AND NONLINEAR-EQUATIONS. * MARYLAND U. COMP.SCI. CENTER.*TR-67-41.*1967.	
CC0013	BURNS, D.E.*THE THEORY OF IMPLEMENTATION OF SRT DIVISION.* ILLINOIS U. COMP.SCI.DEPT.*RPT. 230.*1967.	
CC0014	BARTELD, L.*ON THE ASYMPTOTIC DISTRIBUTION OF THE EIGENVALUES OF RANDOM MATRICES.* WISCONSIN U. MATH.RES. CENTER.*MRC TECH. SUMMARY RPT. 734.*1967.	
CC0015	BARTELD, L. KAHN, P.W.*A DEVELOPMENTAL MODEL OF THE MATHEMATICS OF PROCESSING IN THE CHILD.* ST. U. E.E.DEPT.* 1967.	
CC0016	BARTELD, L.A. KAHN, P.W.*A CYBERNETIC APPROACH TO MENTAL DEVELOPMENT.*	

Figure 4. Sample No Category Mode ACCESS Output (Sheet 1 of 4)

ACCESSION CODE	ACCESSION ENTRY	CATEGORY CODES
000017	ARRIF, M.A. *MATH. NOTES ON A PARTIAL SURVEY OF CYBERNETICS IN EUROPE AND THE U.S.S.R.* P.E.C. RESEARCH ASSOCIATES, INC.*AFOSM-65-1412.*1965.	
000018	ARRIF, M.A. * SELF-REPRODUCING-AUTOMATA; SOME IMPLICATIONS FOR THEORETICAL LOGIC.* ST. U. E.F.DEPT.* 1967.	
000019	ARRIF, M.A. GIVELSON, Y. * ALGEBRA AUTOMATA II: THE CATEGORICAL FRAMEWORK FOR DYNAMIC ANALYSIS.* ST. U. E.F.DEPT.* 1967.	
000020	BAVER, P.*DIHEDRONS AND CONGRUENCES OF AUTOMATA.* BOEING SCI.RES.LABS.*MATH. NOTE 497.*1967.	
000021	BAVER, P.*THE AUTOMORPHISM GROUP OF A STRONGLY CONNECTED AUTOMATON AND ITS QUOTIENT AUTOMATA.* ILLINOIS U. COMP.SCI.DEPT.*RPT.199.*1966.	
000022	BAVER, P.*AUTOMORPHISM GROUPS AND QUOTIENTS OF STRONGLY CONNECTED AUTOMATA AND MODULAR ALGEBRAS.* ILLINOIS U. COMP.SCI.DEPT.*RPT. 204.*1966.	

Figure 4. Sample No Category Mode ACCESS Output (Sheet 2 of 4)

KEYWORD LISTING

00001 APPAHAMS, P. *LISP-2 LANGUAGE SPECIFICATIONS. *SYSTEM 000003
 00002 AFJSD, C. *PARALOG COMPUTATION WITH RANDOM PULSE SEQUENC 000001
 00003 ALGEBRA AUTOMATA II: THE CATEGORICAL FRAMEWORK FOR D 000019
 00004 ALGEBRAS CONSIDERED AS AUTOMATA. ST. U. E. DEPT. * 000011
 00005 AMERICAN STANDARDS ASSN. *WORKING DOCUMENT COROL SPEC-0 00002
 00006 ANDERSON, N. S. ROSENFELD, A. STIMENSON, N. F. *PATTERN-R 000004
 00007 ANSELD, P. M. *MORE, R. H. *AN EXTENSION OF THE NEWTON 000005
 00008 APELKRANS, M. Y. *ON DIFFERENCE SCHEMES FOR HYPERPOLIC-E 000004
 00009 APPLIED LOGIC CORP. *1967. P 000007
 00010 APPLIED LOGIC CORP. *SEMI-ANNUAL REPORT: CTR-AIDED SEMI-000007
 00011 ARBITR, M. A. *SELF-REPRODUCING-AUTOMATA: SOME IMPLICATI 000018
 00012 ARBITR, M. A. *NOTES ON A PARTIAL SURVEY OF CYBERNETICS 1-000017
 00013 ARBITR, M. A. *SOME THEORETICAL APPROACHES TO MFRVUS-SYS 000008
 00014 ARBITR, M. A. *GIVE'ON, Y. *ALGEBRA AUTOMATA II: THE CAT-000019
 00015 ARBITR, L. A. KAHN, R. M. *A DEVELOPMENTAL MODEL OF INFOR 000015
 00016 ARBITR, M. A. KAHN, R. M. *A CYBERNETIC APPROACH TO METAL 000016
 00017 ATKINS, D. E. *THE THEORY OF IMPLEMENTATION OF FRT DIVIS 000013
 00018 AUTOMATA AND MONADIC-ALGEBRAS. *ILLINOIS U. COMP. SCI. 000022
 00019 AUTOMATA II: THE CATEGORICAL FRAMEWORK FOR DYNAMIC A 000019
 00020 AUTOMATA. *ST. U. E. DEPT. *1967. *MATH. NOTE 497. *1967. 000020
 00021 AUTOMATON ANNOTS QUOTIENT AUTOMATA. *ILLINOIS U. COMP. 000011
 00022 BAFCCKER, R. M. *PLANAR REPRESENTATIONS OF COMPLEX GRAP 000021
 00023 BALUEV, A. N. *THREE PAPERS ON CHAPLYGIN-METHODS AND NO 000010
 00024 BALZER, R. M. *STUDIES CONCERNING MINIMAL TIME SOLUTIONS 000009
 00025 BAYER, R. *AUTOMORPHISM GROUPS AND QUOTIENTS OF STRONGL 000022
 00026 BAYER, R. *ENDOMORPHISMS AND CONGRUENCES OF AUTOMATA 000020
 00027 BAYER, R. *THE AUTOMORPHISM GROUP OF A STRONGLY CONNECT 000021
 00028 BOEING SCI. RES. LABS. *MATH. NOTE 457. *1957. 000020
 00029 CARNESIE INST. OF TECHNOLOGY. *1966. *YARYLAN 000009
 00030 CHAPLYGIN-METHODS AND NONLINEAR-EQUATIONS. * 000012
 00031 CHILD. *ST. U. E. DEPT. *1967. *AMERICAN STANDARDS ASSN. *196 000015
 00032 CYBERNETIC APPROACH TO METAL DEVELOPMENT. ST. U. E. DE. 000016
 00033 CYBERNETICS IN EUROPE AND THE U.S.S.R. *P.E.C. RESEAR 000017
 00034 DIVISION. *ILLINOIS. COMP. SCI. DEPT. *RPT. 230. *1967. 000013
 00035 EIGENVALUES OF RANDOM MATRICES. WISCONSIN U. MATH. RE 000014
 00036 FIRING-SQUAD SYNCHRONIZATION PROBLEM. + CARNegie INST. 000009
 00037 GIVE'ON, Y. *ALGEBRA AUTOMATA II: THE CATEGORICAL FR 000019
 00038 GIVE'ON, Y. *CONTRIBUTIONS TO THE THEORY OF ALGEBRAS C 000011
 00039 GRAPHS. *M.I.T. LINCOLN LAB. *N. 1967-1-1967. 000010
 00040 HYPERPOLIC-EQUATIONS WITH DISCONTINUOUS INITIAL-VALUES. 000004
 00041 ILLINOIS U. COMP. SCI. DEPT. *RPT. 204. *1966. 000022
 00042 ILLINOIS U. COMP. SCI. DEPT. *RPT. 199. *1966. 000021
 00043 ILLINOIS U. COMP. SCI. DEPT. *RPT. 255. *1968. 000011
 00044 ILLINOIS U. COMP. SCI. DEPT. *RPT. 230. *1967. 000013
 00045 INFORMATION-PROCESSING IN THE CHILD. *ST. U. E. DEPT. * 000015

Figure 4. Sample I/O Category Node ACCESS Output (Sheet 3 of 4)


```

1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.
46.
47.
48.
49.
50.
51.
52.
53.
54.
55.
56.
57.
58.
59.

```

*LIE
 *CATEGORY NAMES
 C C C M P I L E S
 D C A T A S T R U C T U R E S
 *END CATEGORY NAMES
 *ACCESSION ENTRIES
 RC1E2 SAISTH, V. AND H. JEWBER, "EULER: A GENERALIZATION OF FRAGEL AND ITS CD
 + FORMAL DEFINITION," GADM, JAN. 1966, PG. 13. PRECEDENCE, PARSING
 *END ACCESSION ENTRIES

Figure 5. Sample ACCESS Input To Show Sort Keys

```

S C R I P T KEY PRINT LINE
-----
BC+A ... (NOT USED) ... ***** CATEGORY -- C ... C L M P I L E Q S
BC+R ... (NOT USED)
BC+C ... (NOT USED)
BD+A ... (NOT USED)
BD+E ... (NOT USED)
BD+C
ARC1E2 1 (ACCESS TITLE)
BCW1R1P, A. AND H. JEWBER ... (1ST CATEGORY)
BCW1R1P, A. AND H. JEWBER ... (2ND CATEGORY)
ARC1E2 2 (CONTINUATION IMAGE FOR ACCESSION LISTING)
CW1R1P, A. AND H. JEWBER ... (1ST KEYWORD IN ACCESSION ENTRY)
CABER, "EULER: A GENF ... (2ND KEYWORD IN ACCESSION ENTRY)
CALGLANDIT S FORMAL D ... (3RD KEYWORD I N ACCESSION ENTRY)
CPRECEDENCE, PARSING ... (4TH KEYWORD I N ACCESSION ENTRY)
CWARSING ... (5TH KEYWORD I N ACCESSION ENTRY)

```

Figure 5. Sort Keys Before Sort

60.	S	R	T	K	E	Y	P	R	I	N	T	L	I	N	E
61.	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
62.	ARC182	1													
63.	ARC1922														
64.	BC+A														
65.	BC+E														
66.	BC+C														
67.	BC+RTF														
68.	BC+A														
69.	BD+B														
70.	BC+C														
71.	BC+RTF														
72.	CALCUL														
73.	CPARSING														
74.	CPRECEDENCE														
75.	CWEBER														
76.	CWIRTH														
77.															
78.															
79.															
80.															
81.															
82.															
83.															

Figure 7. Sort Keys After Sort