IX. ADAPTING SMART TO THE M.I.T. COMPUTIBLE TIME-SHARING SYSTEM

Mark Cane

1. Introduction

Conventional document retrieval methods are characterized by the facility with which the user may modify his search procedure on the basis of the "feedback" he receives during the search operation. An operational automatic information retrieval system should ideally be designed so as to provide a similar facility to the user. The present section deals with an information retrieval system useful in the investigation of man-machine interaction in general, and in the determination of particular types of feedback which might be provided. A thorough investigation of man-machine systems is particularly crucial in view of the fact that systems which provide for on-line communication between man and machine tend to incorporate input-output methods which are far more time consuming (and therefore more costly) than the off-line input-output method used in conjunction with the usual large computer installation. For this reason, it is highly desirable to hold communication between the user and the machine to a minimum.

In an earlier report, three facilities were suggested as being necessary for such a system:

(1) A time-sharing system to effect on-line man-machine communication at a reasonable cost;

(2) a flexible set of document retrieval programs which are capable of incorporating information received from the user and of providing alternative retrieval operations; and

(3) a communication-oriented executive routine designed to provide the man-program interaction and to control the scheduling of the various processing options in a manner compatible with the time-sharing system.

The present report deals with (2) and (3); that is, the modification of the SMART system for use with CTSS, and the design of executive programs with the characteristics outlined in (3).

2. Modifications of the SMART System

The SMART system is programmed for the IBM 7094 in FORTRAN and FAP. Both of these languages are available on CTSS, but FORTRAN programs are first translated into the MAD language, and it is the MAD compiler which produces the object program. The differences between MAD and FORTRAN (e.g. MAD uses full word integers, and a different common allocation as well as different types of subroutines and function linkages) necessitate extensive modification of FAP programs which interact with FORTRAN programs. Moreover, all input-output statements had to be completely reformulated.

Because interaction time and cost are functions of the amount of core storage taken up by program and data (the scheduling algorithm used in the M.I.T. CTSS produces a swap time which depends on the amount of memory space covered), it was felt that operations requiring manipulation of the document-concept matrix would not be added to CTSS SMART for the present. This restriction eliminates the possibility of performing document-document correlations and concept-concept correlations; obviously, the opportunity of generating document clusters or concept clusters is also lost.

It is apparent, moreover, that inclusion of the syntactic processing as part of CTSS operations would be very expensive, both because of the extensive reprogramming which would be required, and because of the machine cost of running the syntactic analyzer on CTSS (see *Information Storage and Retrieval*, Report No. ISR-7, Sec. XII, p. 6).[†] The syntax is therefore not considered in the initial CTSS conversion.

The CTSS SMART system does not provide for the processing of original document texts. Instead, the vectors which result from the Phase II processing of texts in the regular SMART system (see ISR-7, Sec. II) are loaded onto a disk file off-line. The SMART system has thus been recast for CTSS along the lines of a request-answering system.

Furthermore, the CTSS SMART system makes no provision for setting up its own library files. However, thesaurus and suffix trees, as well as a dictionary of statistical phrases and the hierarchy are required by the CTSS SMART system, in addition to the

[†] Hereafter referred to as ISR-7.

file of document vectors referred to above. These dictionaries are therefore produced by the regular SMART setup programs and punched onto cards to be loaded off-line onto the CTSS disks. The only programs needed on CTSS to accomplish the loading operations are relatively trivial editing programs to do format conversions (i.e. BCD to binary) and to pack the files more tightly than is possible merely by loading off-line from cards.

In addition to the regular thesaurus, the SMART system provides for the use of a null thesaurus (see ISR-7, Sec. III, pp. 2,15).

The programming changes needed to incorporate this feature into the CTSS SMART system are minimal. However, the addition of such an option would require that the two dictionaries be maintained on the disk files simultaneously. Furthermore, two files of document vectors would then have to be maintained, consisting of a file for the document vectors looked up in each dictionary. Since insufficient track space is presently available, only one dictionary is included on CTSS SMART. Alteration of this dictionary would imply that a new dictionary, as well as a new set of document vectors compatible with this new dictionary, be loaded onto the disk off-line.

No description is presented here of the subroutines in the CTSS SMART system. They are all modifications of, or analogous to, the programs of the same name described in ISR-7. The executive programs are described in some detail in Part 3 which follows.

3. Implementation – Executive Program Description

The user begins by typing the English text of a search request to be answered into a file which must be named "QUEST TEXT." The normal SMART punctuation conventions (described in ISR-7, Sec. II, pp. 22,29), are followed. The text is typed in columns 1-72; anything past column 72 is ignored by the CTSS system. The usual SMART end-of-text conventions are followed, with end of file considered as an additional end-of-text sentinel. (If an end of file is encountered before the other sentinels, a message to this effect is printed along with the line number of the last valid line read by SEGMNT. Processing continues however in the normal way.)

After the user is satisfied that his request is entered correctly, he calls in the SMART system by issuing the sequence of commands

```
NLØAD     SMART
START.
```

The user makes no further interchange with the CTSS system at this point. He is now in the first phase of CTSS SMART; this phase converts the request text into a vector.

The first thing done by the SMART executive program is to call SEGMNT to read in and segment the text (see Flowchart 1).[†] This routine exits on two error conditions: a text of more than 1,000 words and a text of no words. If there is no file named QUEST TEXT, the CTSS system will disconnect the SMART system, and print the message "FILE QUEST TEXT

[†] The flowcharts appear in the Appendix to this section.

NOT FOUND." If any of these error conditions occur, the user must modify his request and reload SMART.

Upon a successful return from SEGMNT, the executive routine extends the memory bounds to allow enough room to store the thesaurus. The request text is then looked up in the thesaurus and the memory bound is reset. If all the words in the request were found in the thesaurus, processing continues. Otherwise, any words not found are printed out on the user's console, and he is given the option of allowing processing to continue, or of allowing SMART to exit so that he may modify his request (for example, if an important word of the request was not found in the thesaurus, he might want to find a synonym which is in the thesaurus).

If processing continues beyond the lookup phase, the executive program now calls KCOUNT to form the request vector. This vector consists of two arrays called MAPKLS and KLSOCC. The address field of each word of MAPKLS contains a concept number (originally found in the thesaurus) that has occurred in the request text. Corresponding to each entry in MAPKLS there exists a word in KLSOCC, whose address field contains the weight of the corresponding concept in the given text exclusive of the title (the first sentence of the text), and whose decrement field contains the weight of the concept computed from the title only. The request vector is now written onto a file called "QUEST VECTOR."

A last facility provided as part of Phase I processing is the statistical phrase finder (see ISR-7, Sec. IX). The user is queried to determine whether he will want to use statistical phrase data at any future point in the processing. If so, the phrase searcher is called, and a file of phrases found is produced. If the user has not requested the statistical phrase option, a file with a word of zeroes is created to indicate to the Phase II program that no statistical phrases were found. Since the phrase finder requires the original English text, and since that text is erased at this point, no possibility exists for performing a phrase search at some future time.

Phase I now ends by calling subroutine CHAIN 1 which sets up the commands to load and start the Phase II programs.

All of Phase II (with the exception of a brief initialization phase) is set up so as to allow the user to re-enter the program as often as he chooses (see Sec. N in Flowchart 2). After an initialization routine, the executive program sets up the document concept matrix. The user is asked to supply the various parameters needed, that is, a weight for the titles; an indication of whether concepts should be weighted logically, and so on (see ISR-7 for a complete discussion of these and all other options referred to below). If the statistical phrase searcher was requested in Phase I, the user now is given the option of having the statistical phrases augment his original request vector. This can be done either by having the statistical phrase concepts <u>added</u> to the original vector, or by having the phrase concepts <u>replace</u> the original ones (see Sec. S of Flowchart 2).

The user also has the option of expanding his request through use of the hierarchy. (See Sec. H of Flowchart 2; Sec. V of ISR-7.) All hierarchy options that exist in the standard SMART system may be used on CTSS. During all passes through Phase II other than the first, the request vector being manipulated may differ from the original request vector (for example, a hierarchical expansion may have been performed during the last previous pass).

The user then has the choice of applying a hierarchical expansion either to the original vector (i.e., the vector as it existed at the end of Phase I), or the modified vector (i.e., the vector as it existed at the end of the last pass). If, however, the user has requested further modifications of the request vector, that is by changing the weight of the titles or of the concepts, or by specifying a different action for statistical phrases, the use of the vector from the previous pass would have the effect of nullifying such specifications. In such a case, hierarchical expansion can only be applied to the vector resulting from the changes made on this pass to the original vector. (See the example of a console session in Part 4 of this section.) Provision is made for expansions of the modified vector so as to permit multiple hierarchical expansions, such as an expansion to brothers followed by an expansion by cross-references.

After completion of each hierarchical expansion, the request vector is added to the document concept matrix (Sec. K of Flowchart 2). The executive program now calls the subroutines needed to perform the correlations of the request with the documents. The user is queried to determine what correlation mode and what cut-off value in the correlation coefficient are desired. The number of answers are then typed out on the user's console, and he is asked whether he wishes to see the correlations as well as the document identifications which constitute the answers. If so, these are also printed out.

At this point, the user is asked whether he is satisfied with the answers received, or whether the search request should be reprocessed with suitable modifications. If no reprocessing is requested, CTSS SMART deletes any files which were created (that is, the request vector and statistical phrase file) and exits. If the user requests modifications, control returns to the beginning of Phase II (see Sec. N of Flowchart II).

During any pass through Phase II other than the first, the user may type "SAME" in answer to any question that SMART puts to him. This is both a convenient shorthand for the user and a device to avoid unnecessary reprocessing. For example, assume the only modification the user wishes to make is a hierarchical expansion. There is no need then to set up the entire document concept matrix again; only the request vector need be modified. If the user responds to queries asking for the parameters needed to set up the matrix (that is, the title and concept weights) by typing "SAME," no such unnecessary reprocessing will be performed.

4. Progress Report

With the exceptions of the hierarchy and statistical processing, all described modifications to the regular SMART system are completed. The coding of the executive programs has also been completed. However, much of this coding has yet to be debugged.

It may prove essential to provide the user with further information during the course of processing. For example, information about the contents of the request vector and about the phrases found statistically might make it possible to come to more rational decisions concerning the use of the statistical phases. Such information might also aid the user in choosing the proper hierarchical expansion.

Some typical processing sequences are given below, together with the respective typeout sequences.

SIMULATION OF A TYPICAL CONSOLE SESSION

[The user begins by logging-in. Once logging-in is completed, he proceeds to set up his request text.]

```
INPUT

00010   TELL ME ABOUT PATTERN RECOGNITION.  HOW ARE TWO DIMENSIONAL
00020   PATTERNS SCANNED AND ENCODED FOR A COMPUTER.  QUE.  WHAT
00030   PROCEDURES ARE USED TO DETECT AND IDENTIFY GEOMETRICAL SHAPES,
00040   AND TO RECOGNIZE ALPHABETIC CHARACTERS.  QUE.
```

```
00050   *
00660
 MAN.   FILE QUEST TEXT
```

[A file called "QUEST TEXT" containing the request text has been set up. The user may now issue the PRINTF command and inspect the printed text to ensure that it is correct. If it is not satisfactory, he may use the CTSS EDIT command to alter it. Once he is satisfied that the text is correct, he calls the SMART system into play.]

```
NLØAD   SMART
START
```

Execution

[The SMART system is now in control. It prints out the words in the text which were not found in the thesaurus and gives the user the option of: (1) continuing the processing, or (2) if a crucial word has been missed, processing may be stopped and the text altered, for example, by using a synonym which will be found.]

```
  WORD NOT FOUND      KIND     LOC    NUM    SENTENCE AND WORD NUMBERS

  TELL                SUFFIX    2      1              1,1
  TWØ-DIMENSIØNAL     SUFFIX    2      1              2,3
  IDENTIFY            SUFFIX    2      1              3,8
  GEØMETRICAL         SUFFIX    1      1              3,9
  ALPHABETIC          SUFFIX    3      1              3,15
```

IF YOU WOULD LIKE PROCESSING TO CONTINUE PLEASE TYPE GO ON
IF NOT PLEASE TYPE STOP AND GO BACK TO START.

GØ ØN

[The user wishes to continue. See ISR-7, Sec. IV-8 for an explanation of the words not found in print-out.]

WILL YOU WANT TO USE STATISTICAL PHRASES AT ANY TIME, NO

[The user has indicated that he will not want to use statistical phrases. SMART now forms the request vector and puts this program out onto a disk file. SMART then chains into the second link of the SMART system.]

Execution

[1] YOU ARE NOW IN THE SECOND PHRASE OF SMART. PLEASE SPECIFY PARAMETERS AS THEY ARE CALLED FOR.

[2] HOW MUCH WOULD YOU LIKE TITLES TO BE WEIGHTED, 100

[3] WOULD YOU LIKE ALL CONCEPTS TO BE WEIGHTED LOGICALLY, NØ

[SMART now has sufficient information to set up the document-concept matrix with the exception of the request vector. It sets up the matrix and calls for more information about the request vector.]

[4] DO YOU WISH TO EXERCISE ANY HIERARCHY OPTIONS, NO

[Since no hierarchy is wanted, the request vector is added to the matrix immediately.]

```
[5]     PLEASE SPECIFY THE CORRELATION METHOD YOU PREFER, CØSINE

[6]     WHAT CUTOFF VALUE WOULD YOU LIKE, 3100
```

[SMART can now perform the correlation and find answers.]

```
[7]     SMART IS PLEASED TO ANNOUNCE THAT YOU HAVE 6 ANSWERS.  WOULD
        YOU LIKE TO SEE THE CORRELATIONS AS WELL AS THE ANSWER, YES

[8]                   ANSWER            CORRELATION

                 351A  SYSTEM              0.6481
                  353  THE DESIG           0.4207
                  350  PATTERN R           0.4027
                 163A  COMPUTE             0.3607
                   82  ARTIFICIAL          0.3527
                   1A  COMPUTER            0.3161
```

[The full title of the document is not printed out now because there is insufficient disk space to permit keeping full titles on the disk. The full titles of the above documents follow:]

(a) A System for the Automatic Recognition of Patterns,

(b) The Design and Operation of the Mechanical Speech Recognizer at the University College, London,

(c) Pattern Recognition by Means of Automatic Analogue Apparatus,

(d) A Computer Simulation Chain for Research on Picture Coding,

(e) Artificial Auditory Recognition in Telephony,

(f) A Computer Oriented Toward Spatial Problems.

[9] WOULD YOU CARE TO RESPECIFY ANY PARAMETERS OR TRY ANY HIERARCHY EXPANSIONS, YES

[10] O.K. THEN, HERE WE GO AGAIN. PLEASE SPECIFY PARAMETERS AS THEY ARE CALLED FØR.

IF YOU WISH YOU MAY LEAVE SPECIFICATIONS UNCHANGED BY TYPING SAME. GOOD LUCK.

[11] HOW MUCH WOULD YOU LIKE TITLES TO BE WEIGHTED, SAME

[Titles will be weighted the same as for the last pass.]

[12] WOULD YOU LIKE CONCEPTS TO BE WEIGHTED LOGICALLY, SANE

[The user has typed SANE instead of the intended SAME.]

[13] THE ABOVE PARAMETER SPECIFICATION IS INCORRECT. PLEASE RESPECIFY, SAME

[Since neither of the above parameters has changed, the document matrix, possibly excepting the request vector, is unchanged. SMART goes on to process the request vector.]

[14] [Same as line 4 except response is now "YES"]

[15] WHAT TYPE OF EXPANSION WOULD YOU LIKE, PARENTS

[16] DO YOU WANT TO ADD TO THE VECTOR OR REPLACE IT WITH HIERARCHY CONCEPTS, ADD

[SMART now calls the hierarchy routines and forms the new vector. This is then added to the document-concept matrix.]

[17] (Same as line 5)

[18] (Same as line 6 except response is now 3700)

[17] (Same as line 7 except 3 answers)

[18]
```
              ANSWER          CORRELATION

        351A  SYSTEM            0.4661

          60  INTELLIGEN        0.4002

          1A  COMPUTER          0.3773
```

(The full title of document 60 is "Intelligent Behavior in Problem-Solving Machines.)

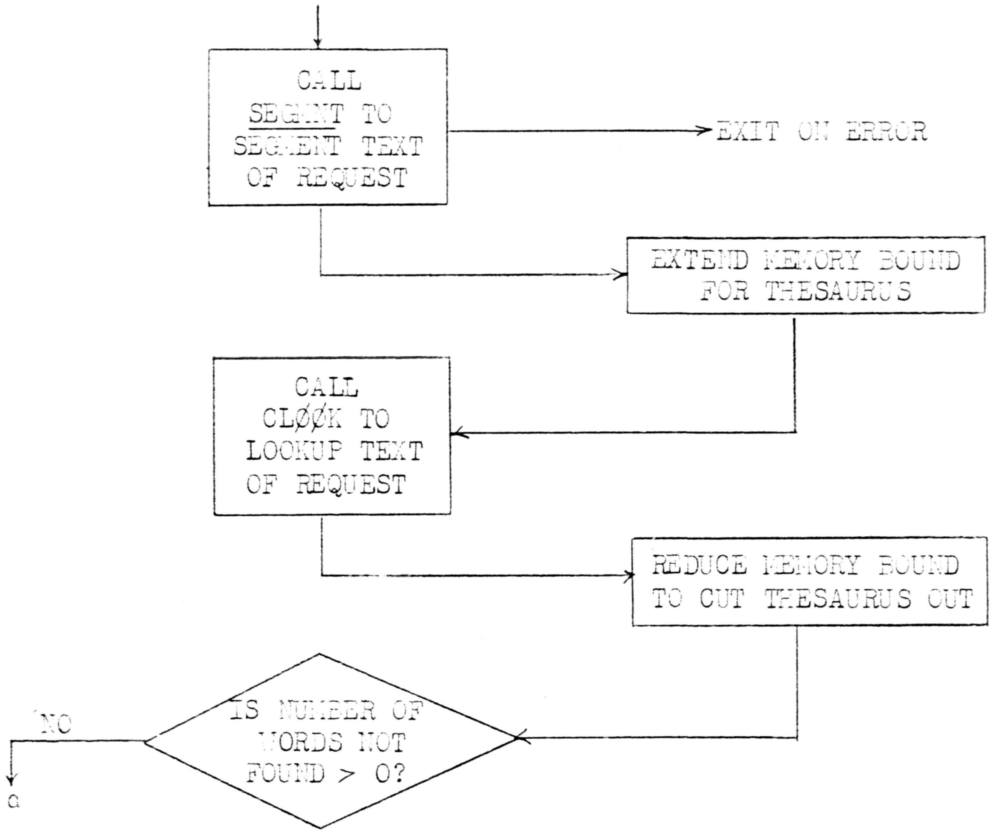
[19] (Same as 9 except response is NO)

(SMART now deletes all auxiliary files that it has created.)

[20] GOODBYE, ITS BEEN A PLEASURE WORKING WITH YOU.

(SMART now returns control to the CTSS monitor)

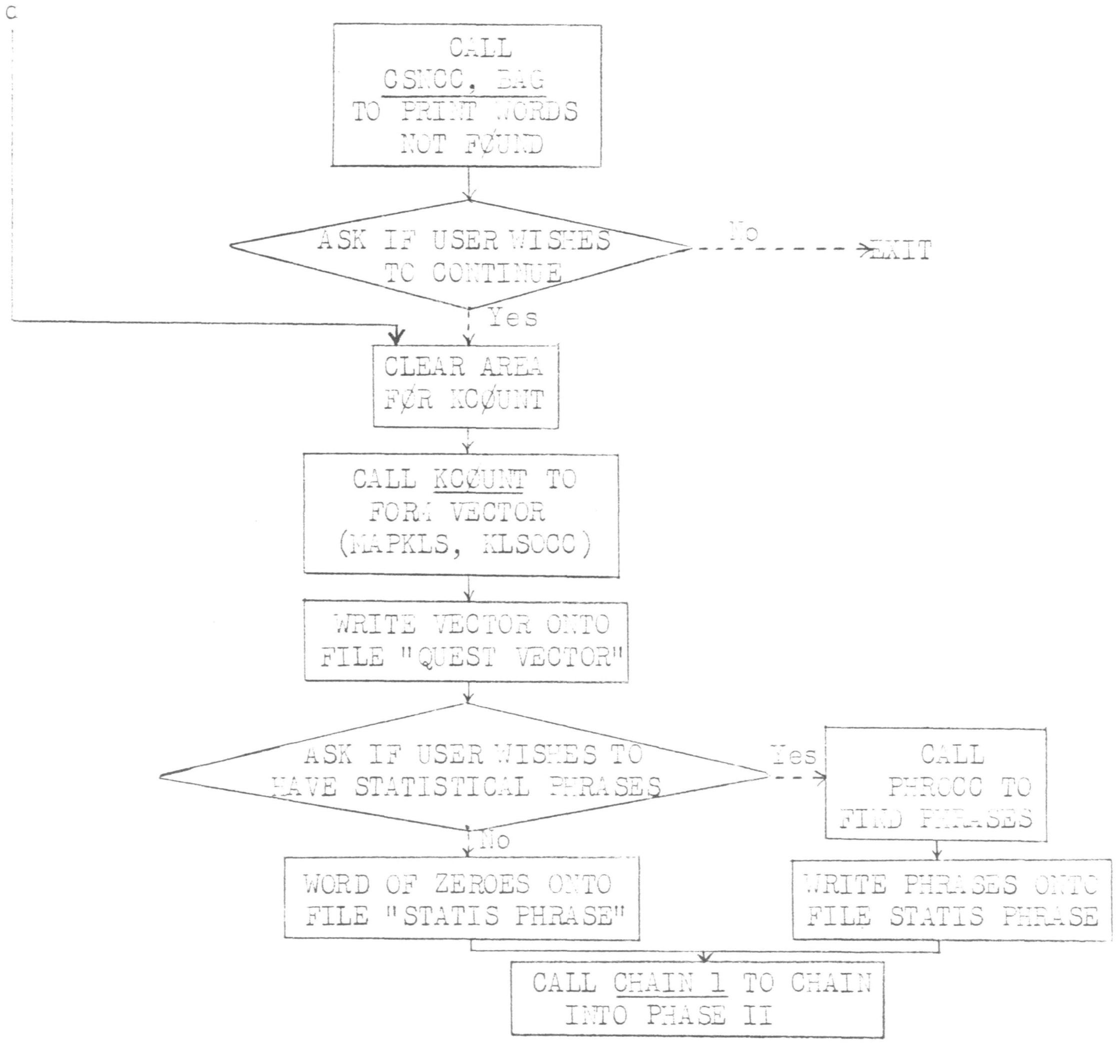
APPENDIX



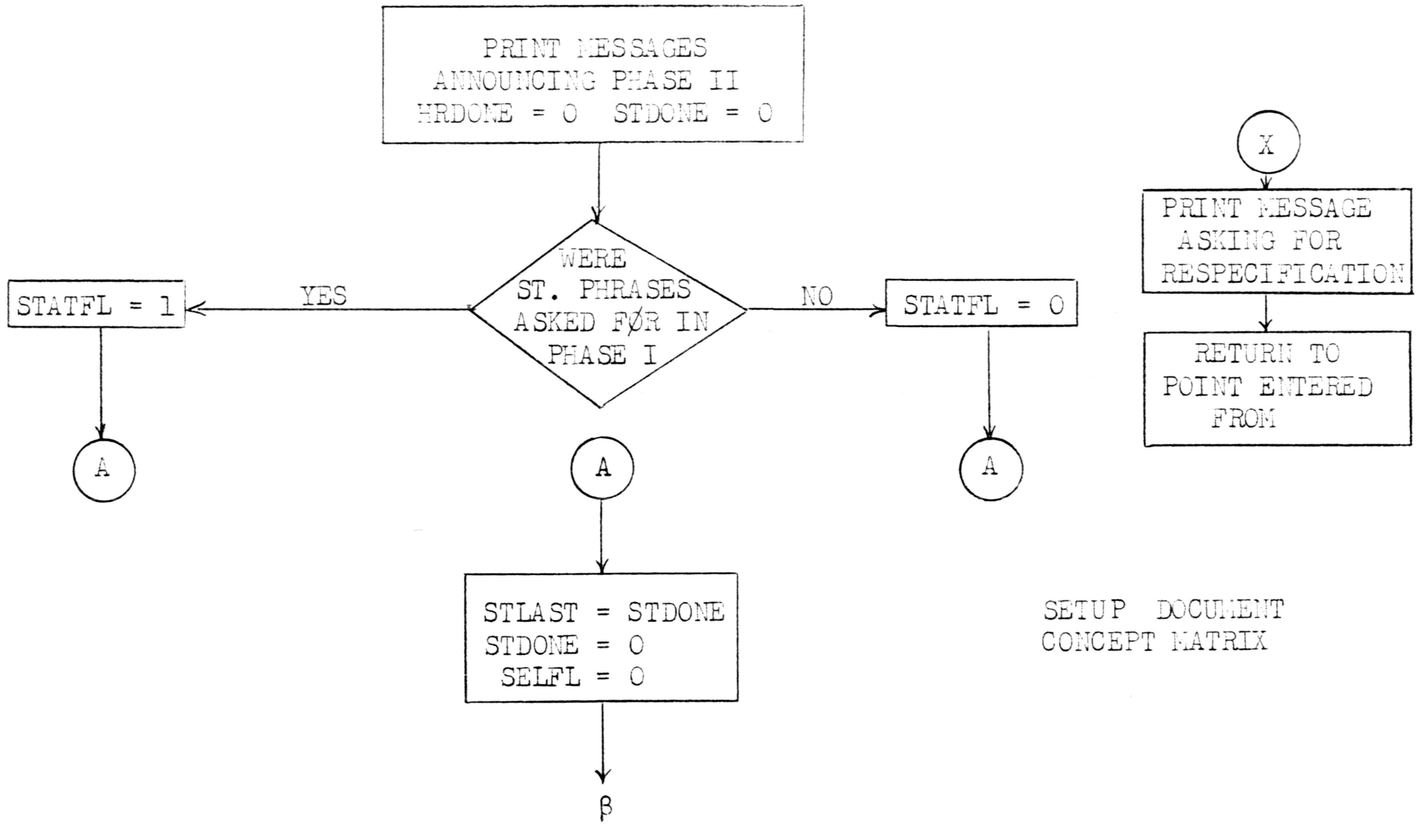
Note: broken arrow
indicates user option

CTSS SMART PHASE I – PROCESSING OF REQUEST TEXT

Flowchart 1



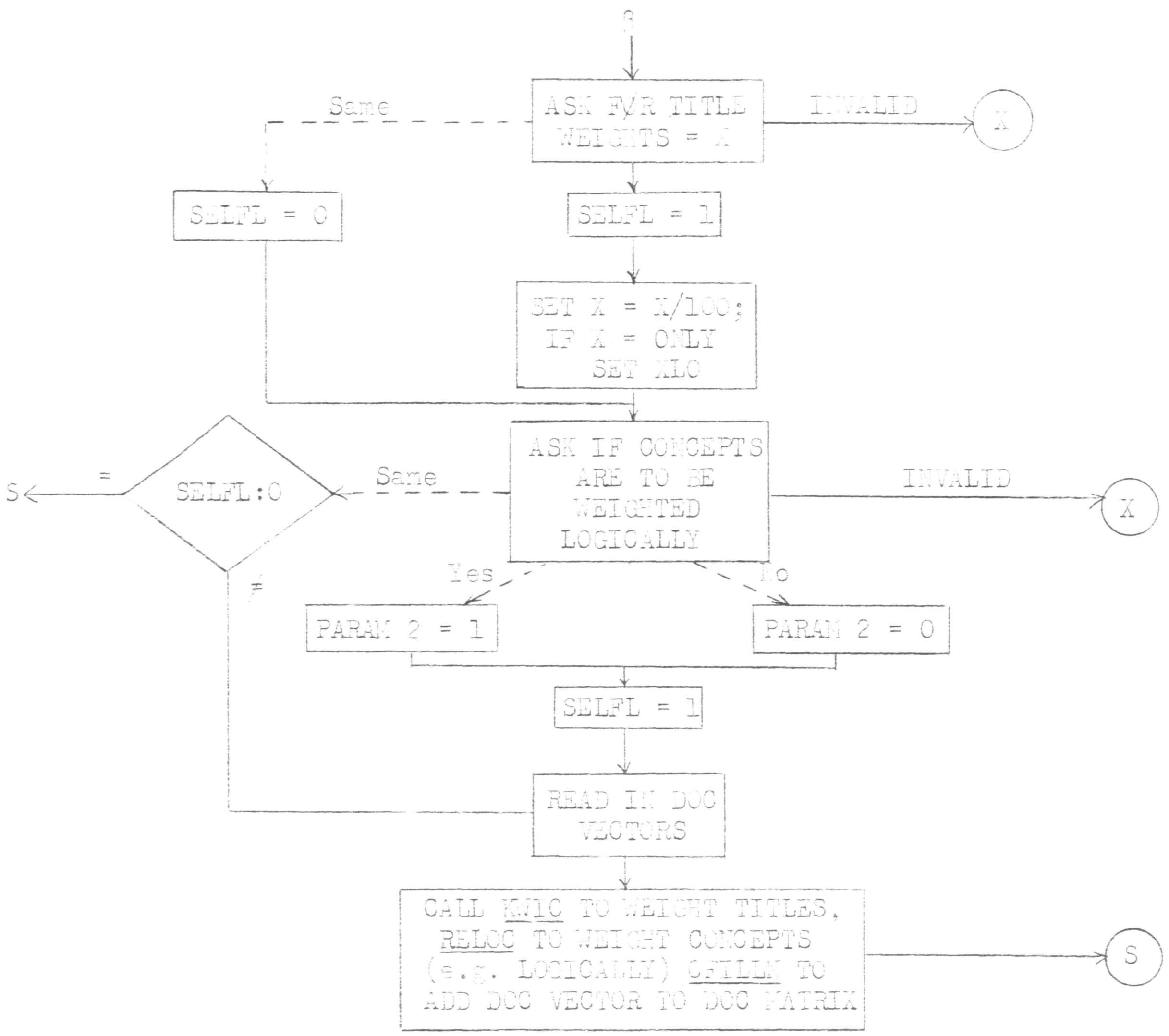
Flowchart 1 (continued)



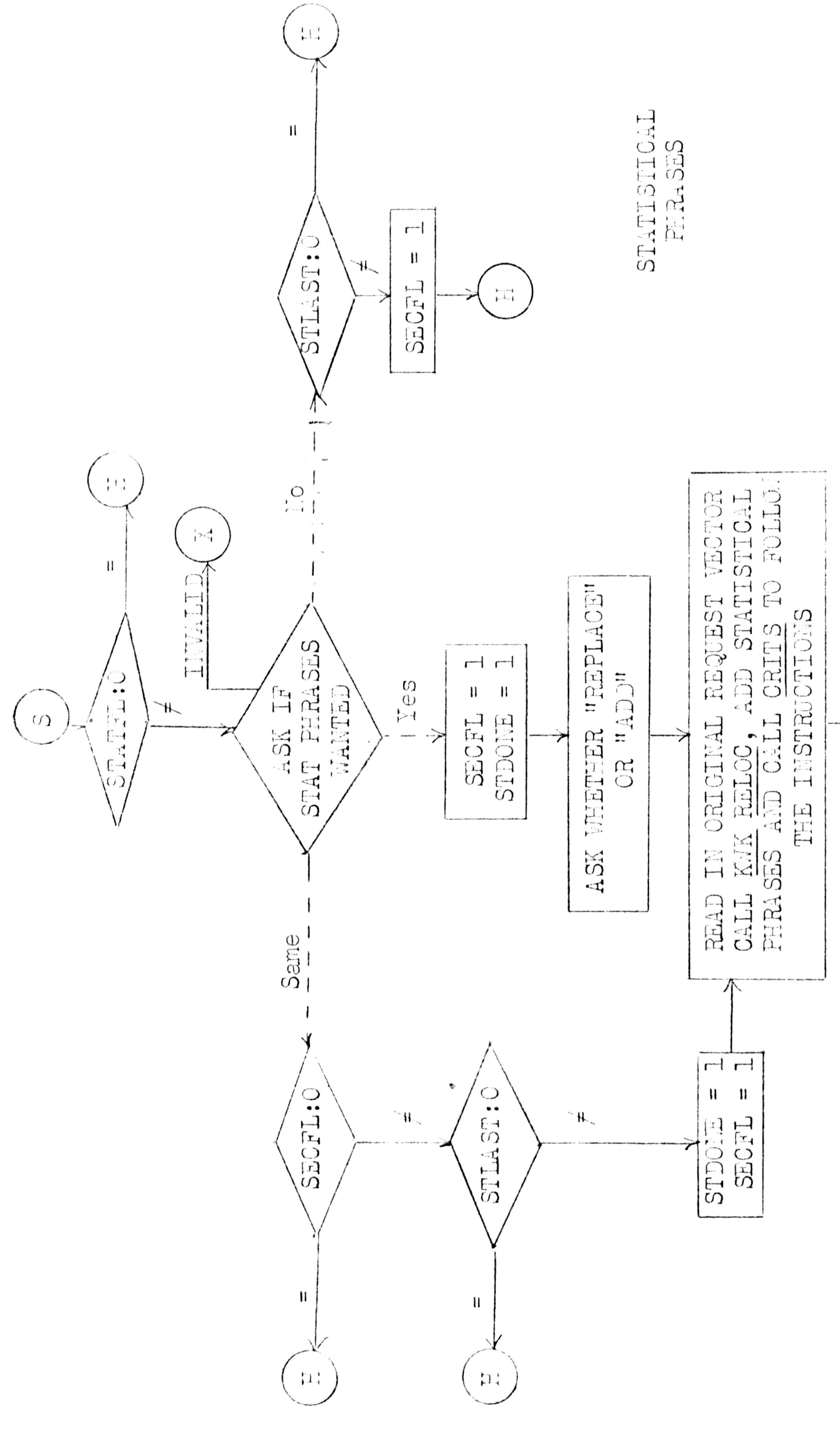
Note: the branches labeled
"Same" can apply only after
the first pass

CTSS SMART PHASE II — PROCESSING OF REQUEST VECTOR

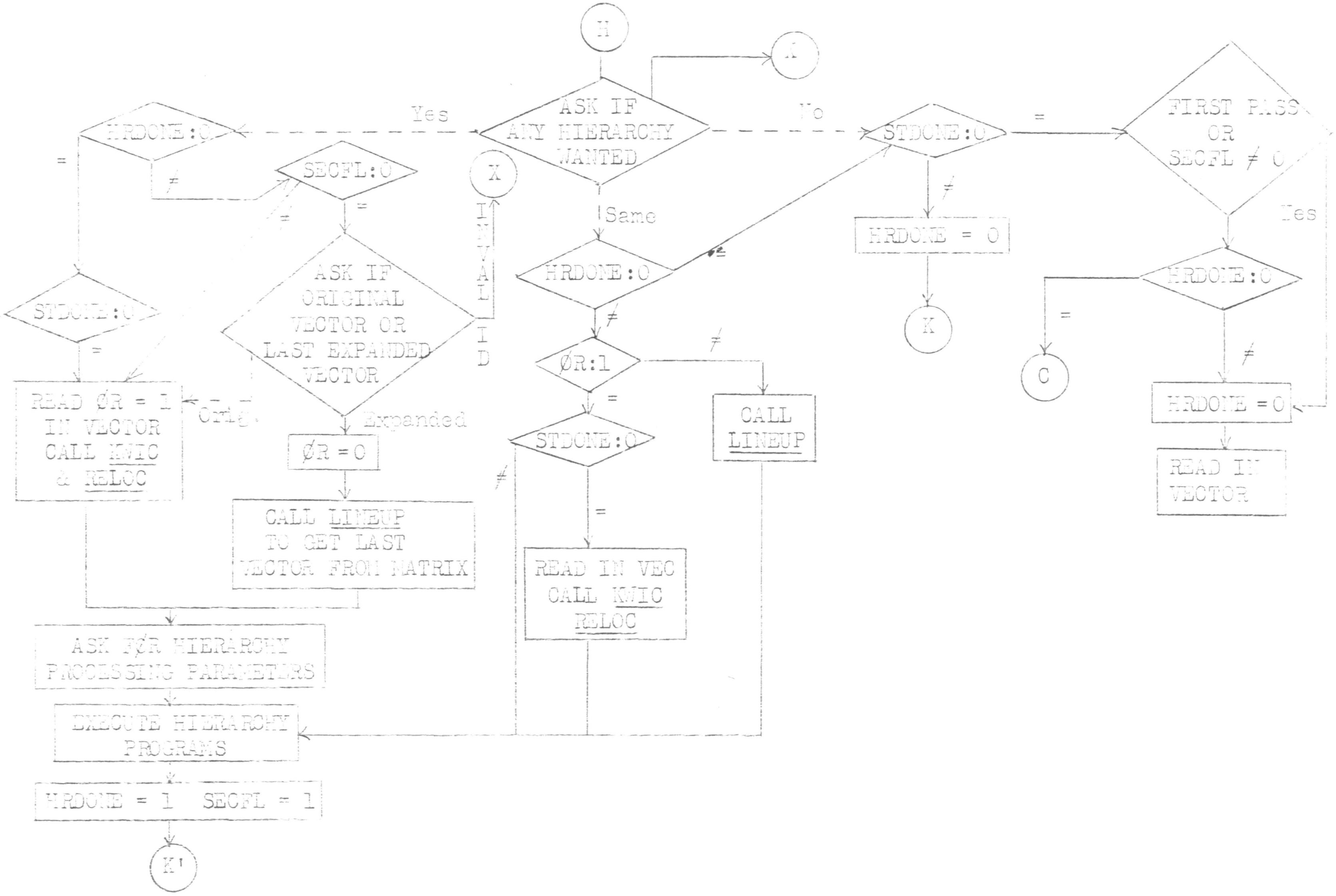
Flowchart 2



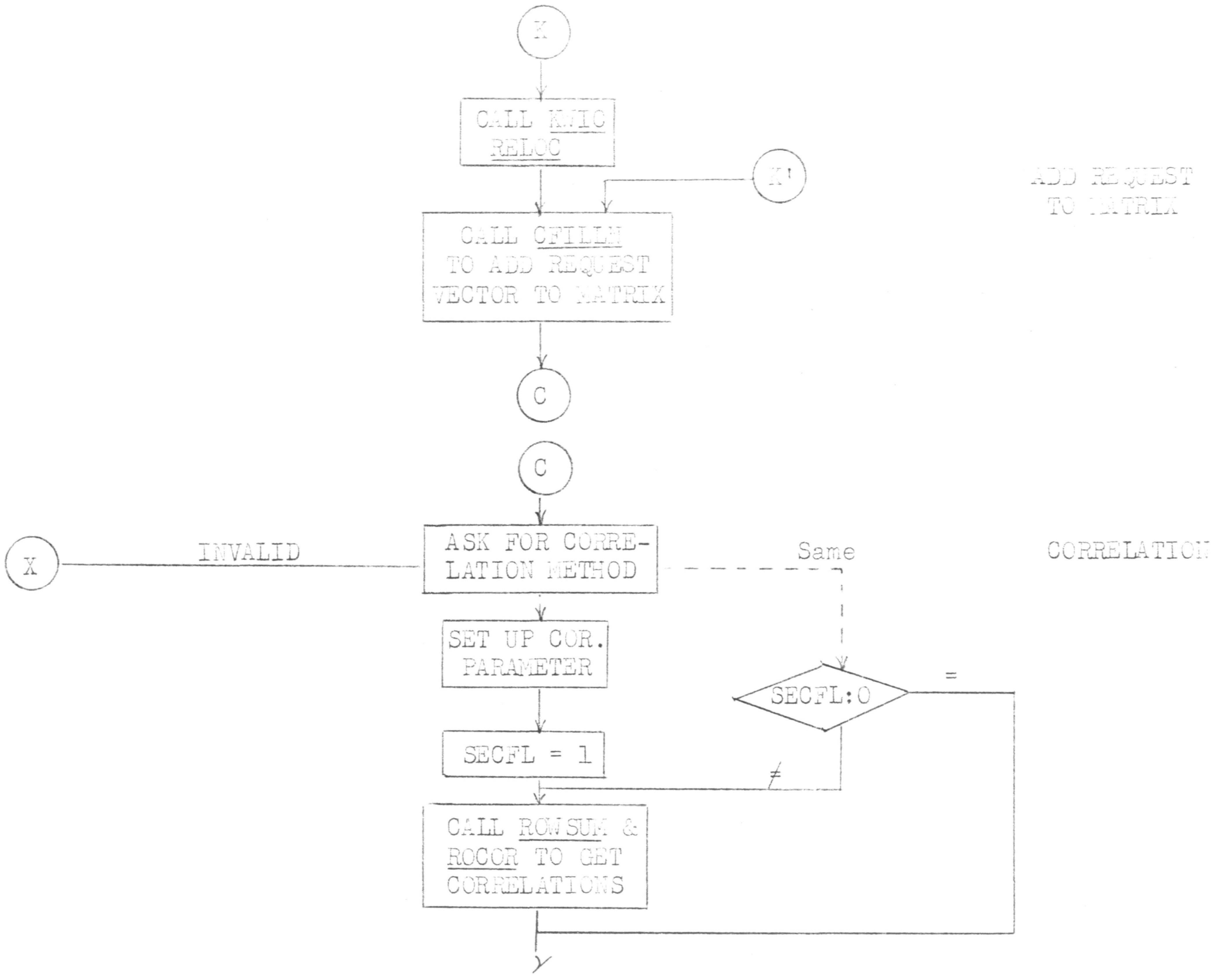
Flowchart 2 (continued)


S
STATFL:0
=
H
≠
ASK IF STAT PHRASES WANTED
INVALID
X
No
STLAST:0
=
H
≠
SECFL = 1
H
STATISTICAL PHRASES
Yes
SECFL = 1
STDONE = 1
ASK WHETHER "REPLACE" OR "ADD"
READ IN ORIGINAL REQUEST VECTOR CALL KWK RELOC, ADD STATISTICAL PHRASES AND CALL CRITS TO FOLLOW THE INSTRUCTIONS
H
Same
SECFL:0
=
H
≠
STLAST:0
=
H
≠
STDONE = 1
SECFL = 1

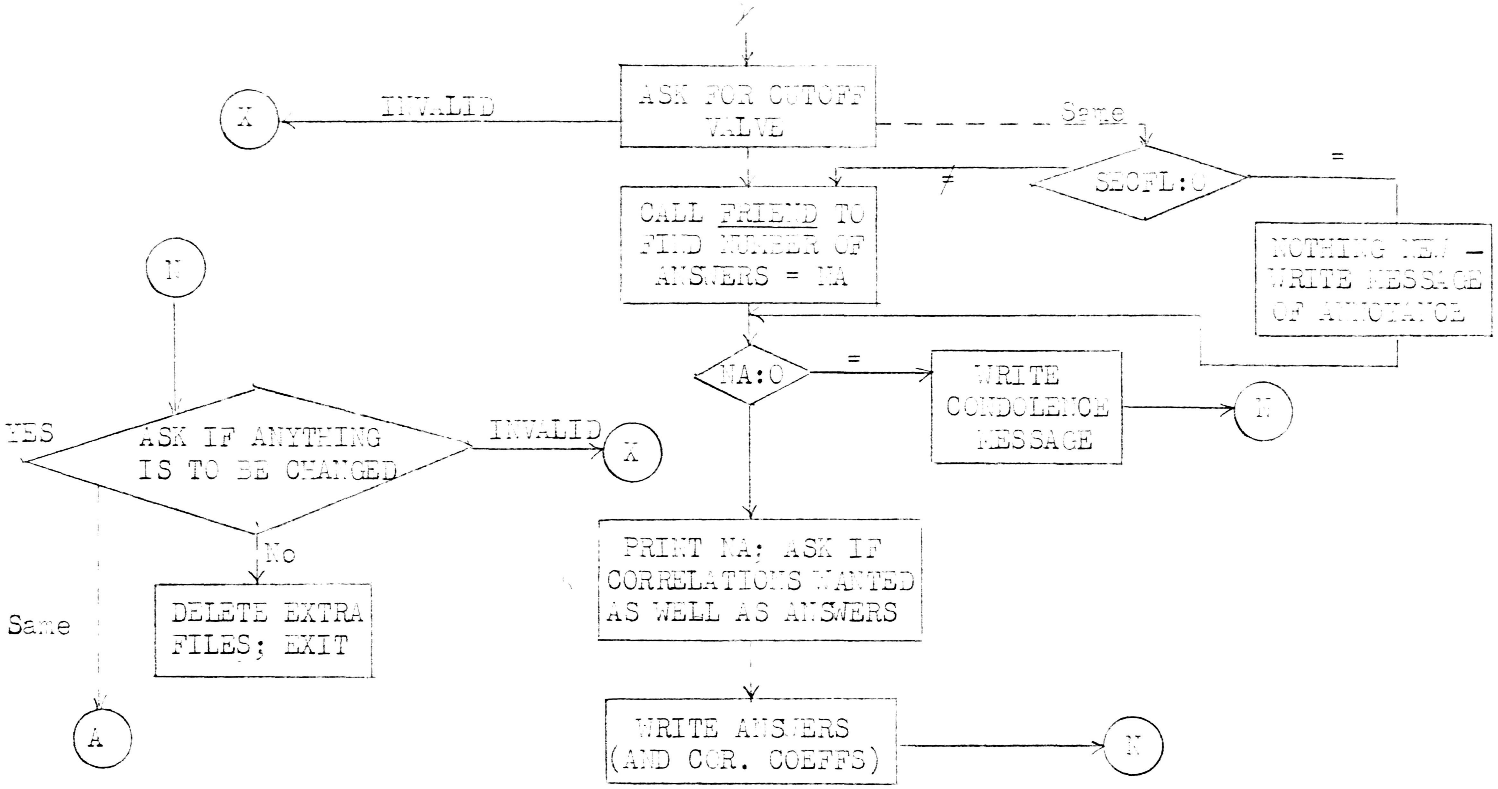
Flowchart 2 (continued)


H
ASK IF ANY HIERARCHY WANTED
A
Yes
No
HRDONE:0
=
≠
SECFL:0
=
≠
X
INVALID
Same
STDONE:0
=
ASK IF ORIGINAL VECTOR OR LAST EXPANDED VECTOR
Orig.
Expanded
READ ØR = 1 IN VECTOR CALL KWIC & RELOC
ØR = 0
CALL LINEUP TO GET LAST VECTOR FROM MATRIX
HRDONE:0
≠
≠
ØR:1
=
≠
STDONE:0
=
≠
CALL LINEUP
READ IN VEC CALL KWIC RELOC
ASK FØR HIERARCHY PROCESSING PARAMETERS
EXECUTE HIERARCHY PROGRAMS
HRDONE = 1 SECFL = 1
K'
STDONE:0
=
≠
HRDONE = 0
K
FIRST PASS OR SECFL ≠ 0
Yes
HRDONE:0
=
≠
C
HRDONE = 0
READ IN VECTOR

Flowchart 2 (continued)


X
CALL KWIC RELOC
X'
CALL CFILLM TO ADD REQUEST VECTOR TO MATRIX
ADD REQUEST TO MATRIX
C
C
X
INVALID
ASK FOR CORRE- LATION METHOD
Same
CORRELATION
SET UP COR. PARAMETER
SECFL:0
=
SECFL = 1
≠
CALL ROWSUM & ROCOR TO GET CORRELATIONS
Y

Flowchart 2 (continued)


ASK FOR CUTOFF VALVE
INVALID
X
Same
SEOFL:0
=
≠
CALL FRIEND TO FIND NUMBER OF ANSWERS = NA
NOTHING NEW — WRITE MESSAGE OF ANNOYANCE
NA:0
=
WRITE CONDOLENCE MESSAGE
N
N
YES
ASK IF ANYTHING IS TO BE CHANGED
INVALID
X
No
DELETE EXTRA FILES; EXIT
Same
A
PRINT NA; ASK IF CORRELATIONS WANTED AS WELL AS ANSWERS
WRITE ANSWERS (AND COR. COEFFS)
N

Flowchart 2 (continued)