XIV. Concept-Concept and Document-Document Correlations

M. Cane and G. Shapiro

1. Introduction

This section describes the algorithm employed to perform concept-concept and document-document correlations. The algorithm uses a tape containing a number of vectors and correlates each vector with all of the others; all correlations exceeding a user specified cutoff value are written out. The user also specifies the correlation mode (either cosine or overlap) to be employed in performing the correlations. The general method is to first fill a buffer with as many vectors as possible, correlating each of them with the others in the buffer. After the buffer is filled, the remaining vectors on the input tape are passed through and correlated with all the vectors already in the buffer. As the vectors are correlated they are written out onto a new tape. After all vectors on the first tape have been processed in this manner, the vector tape just created is used as the input tape for the next correlation pass. The vectors on this new tape are again read into a buffer and correlated with each other. When the buffer is filled, the remaining vectors are again passed through core and written onto a new tape. This process continues until all vectors have been correlated with all others.

Part 2 of this section describes the operation of this algorithm in greater detail. Part 3, which follows, describes how the algorithm is used for concept-concept correlations. The final part discusses its application

to document-document correlations.

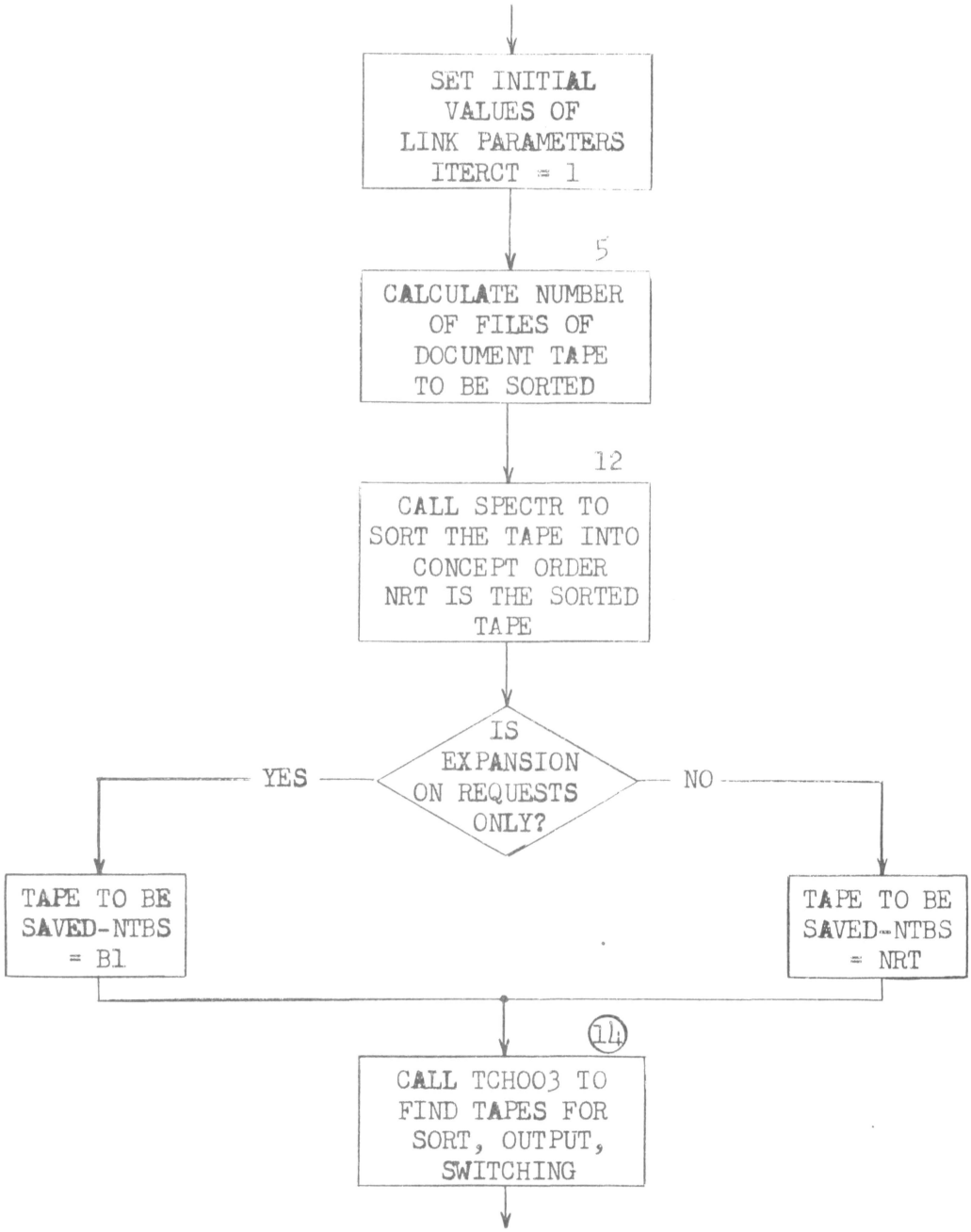
2. Correlation Algorithm

The general outline of the correlation algorithm is shown within the dotted lines in Flowchart 1. Each call to SPASSI or CCPASS corresponds to a correlation pass — i.e., the correlation of a single buffer load of vectors with all the vectors that remain.

Subroutine CPASSI executes the first correlation pass. It uses six arguments: NIT, NOUT, N2T, IDCCUT, IFIN, and ITERCT. NIT is the logical number of the input tape; NOUT is the number of the tape which is to receive the correlations which are written out; N2T is number of the tape which receives the vectors that didn't fit into the buffer space; IFIN is set to non-zero by CPASSI only if all vectors have been correlated with all others. IDCCUT and ITERCT control the format of the input tape for the concept-concept correlations; these are explained in the next part.

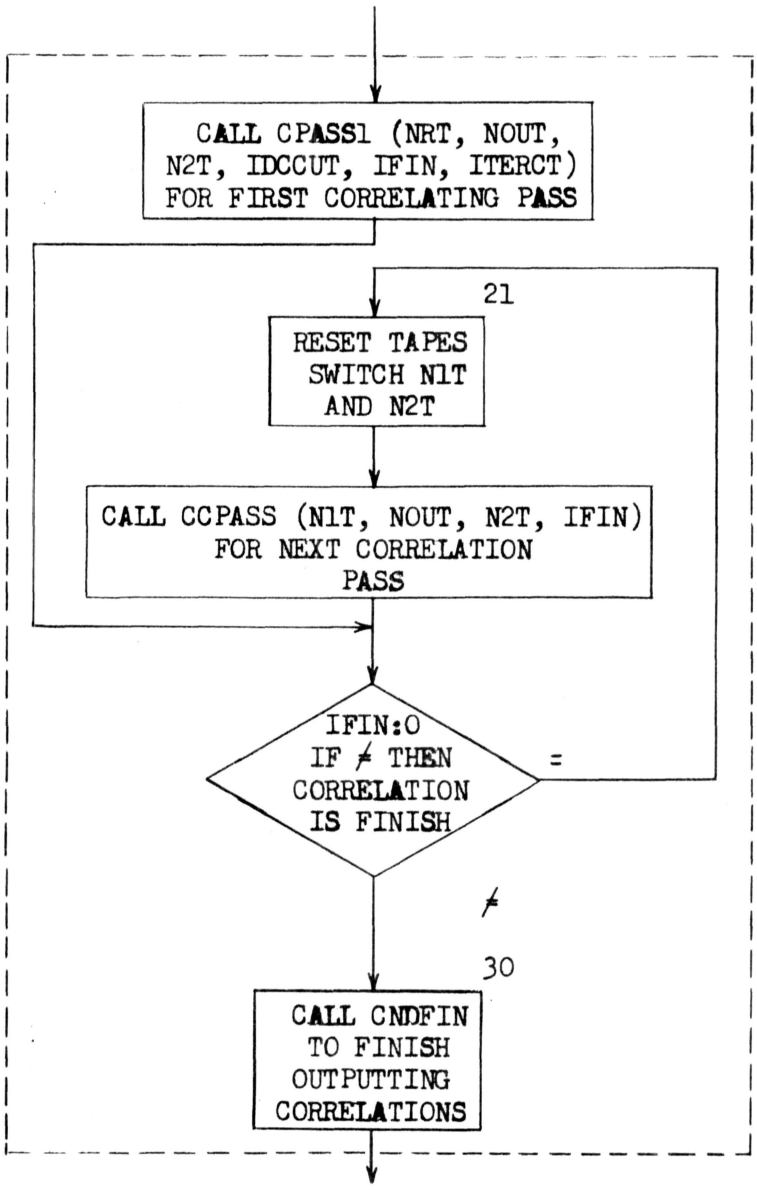
Subroutine CPASSI is described in Flowchart 2. The operations of CPASS 1 may be understood more easily if the subroutines called by this routine are examined first:

(1) CFETCH and LFETCH. CFETCH is an initialization entry for LFETCH. It has five arguments--NIT, LWH, ICON, ITERCT and IDCCUT. NIT, ITERCT and IDCCUT are identical with the arguments to CPASSI. LFETCH will return a code to LWH and an identifier to the two word FORTRAN array ICON. When LFETCH is called it returns the next vector entry to the accumulator. (Each vector entry has a concept number in the decrement

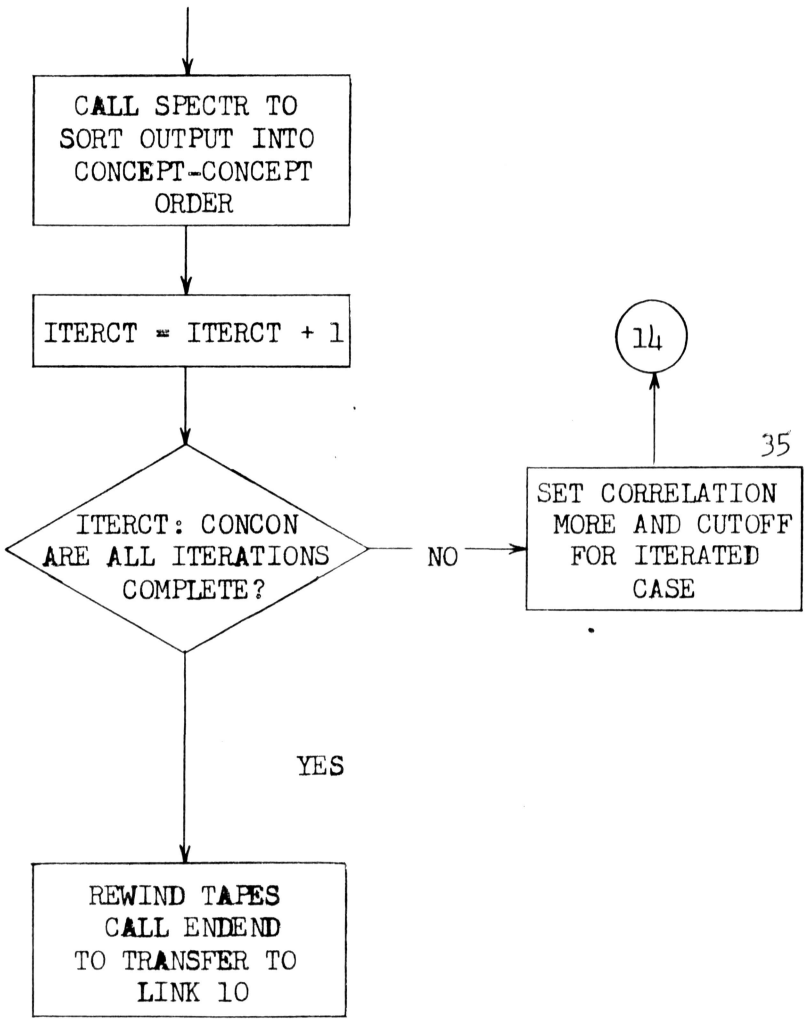


SUPER9 – SUPERVISOR FOR LINK 9

FLOWCHART 1



FLOWCHART 1 (CONTINUED)



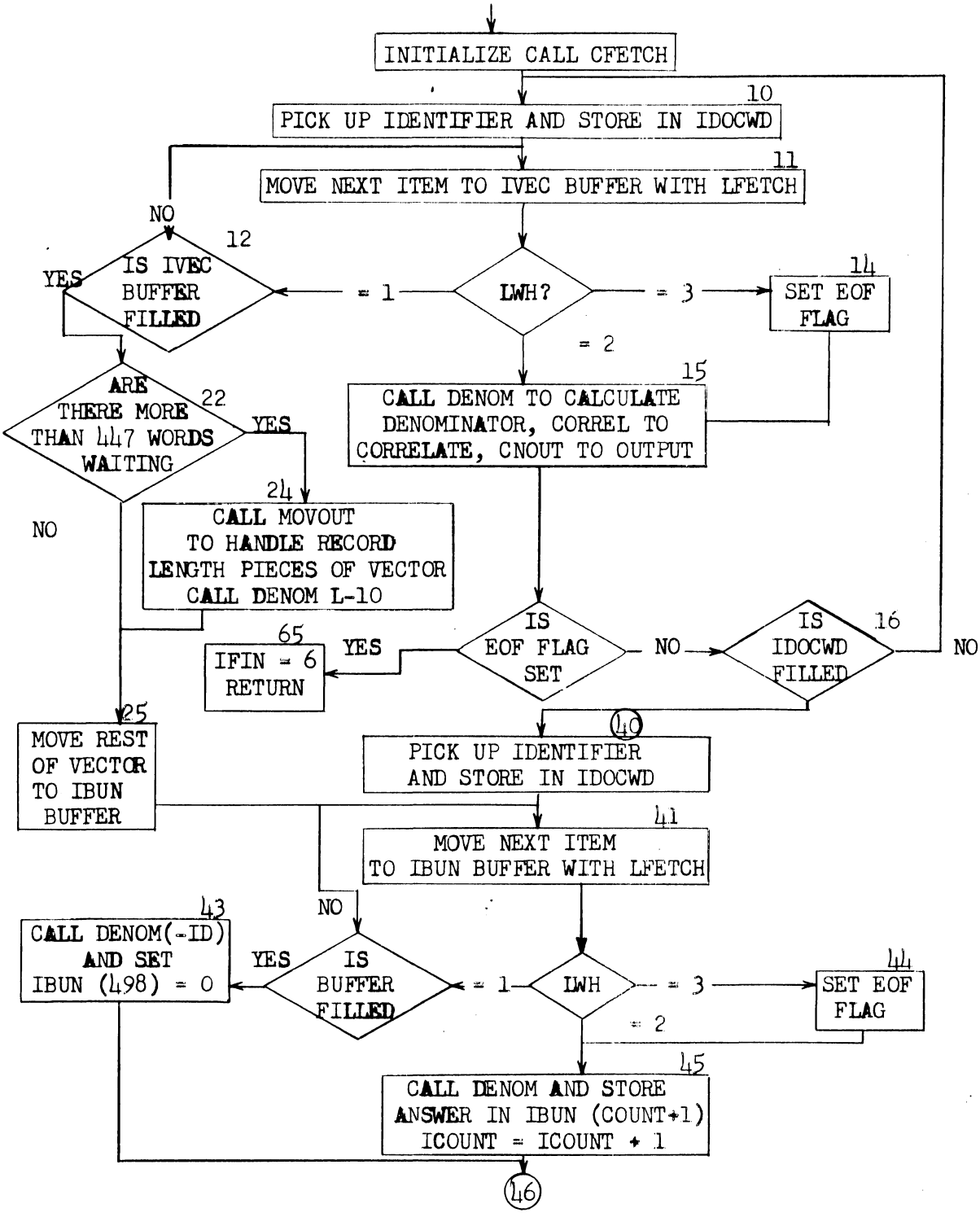
FLOWCHART 1 (CONTINUED)

and a weight in the address.) LWH is set to 2 if the present entry is the last entry of the vector, and to 3 if it is the last entry on the tape. It is set to 1 otherwise. When LWH is 2, ICON will contain the identifier for the next vector.

(2) The other subroutines act on an array of five word items called IDOCWD. The first two words of each item contain the identifier for the vector. The decrement of the third word is the index in the buffer IVEC of the first word of the vector; the address points to the word following the last word of the vector. These items are put into the array by the program that fills the IVEC buffer (CPASS1 in this case). The fourth word of the item is the denominator for the vector. It is entered in the Ith item of IDOCWD by calling DENOM (I). If DENOM is called with a negative argument, a partial denominator is formed since the vector given is not complete, so that a full denominator cannot as yet be calculated. The fifth word of each item is entered into the array by subroutine CORREL. Calling CORREL (I) causes the correlation of vector I with vector J to be entered in the last word of the Jth item of IDOCWD, for $J \leq$ NREQ. NREQ identifies a location which is 99 words down from the top of common. When CORREL is called with a negative argument a partial correlation is performed.

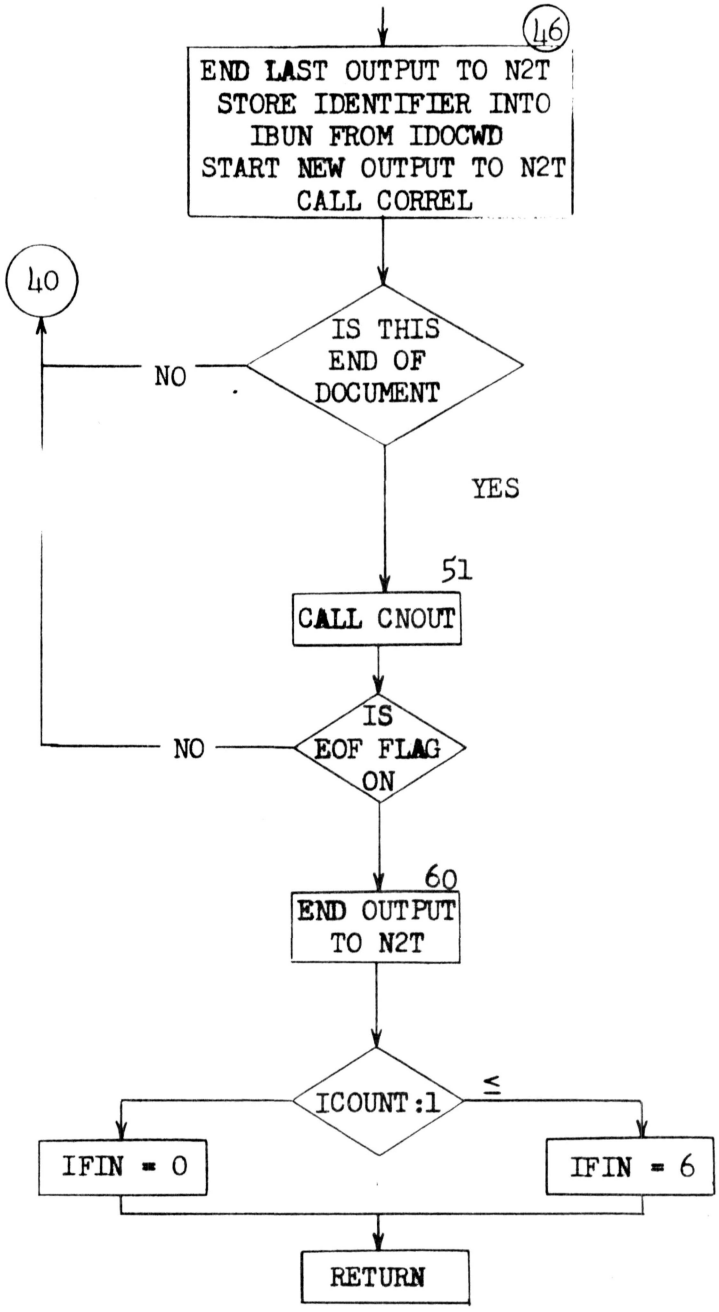
(3) CNOUT searches IDOCWD and writes out all correlations above the cutoff value. Two arguments are used: The first is the logical number of the output tape; the second is I, the index of the item in IDOCWD which is being correlated with the NREQ vectors.

The section of CPASS1 from box 10 of Flowchart 2 up to box 40 fills a buffer named IVEC and correlates these vectors with each other. Three



SUBROUTINE CPASS1

FLOWCHART 2



FLOWCHART 2 (CONTINUED)

conditions cause control to be transferred out of this section:

(1) The input tape is exhausted. In this case all vectors have been correlated so IFIN is set greater than zero.

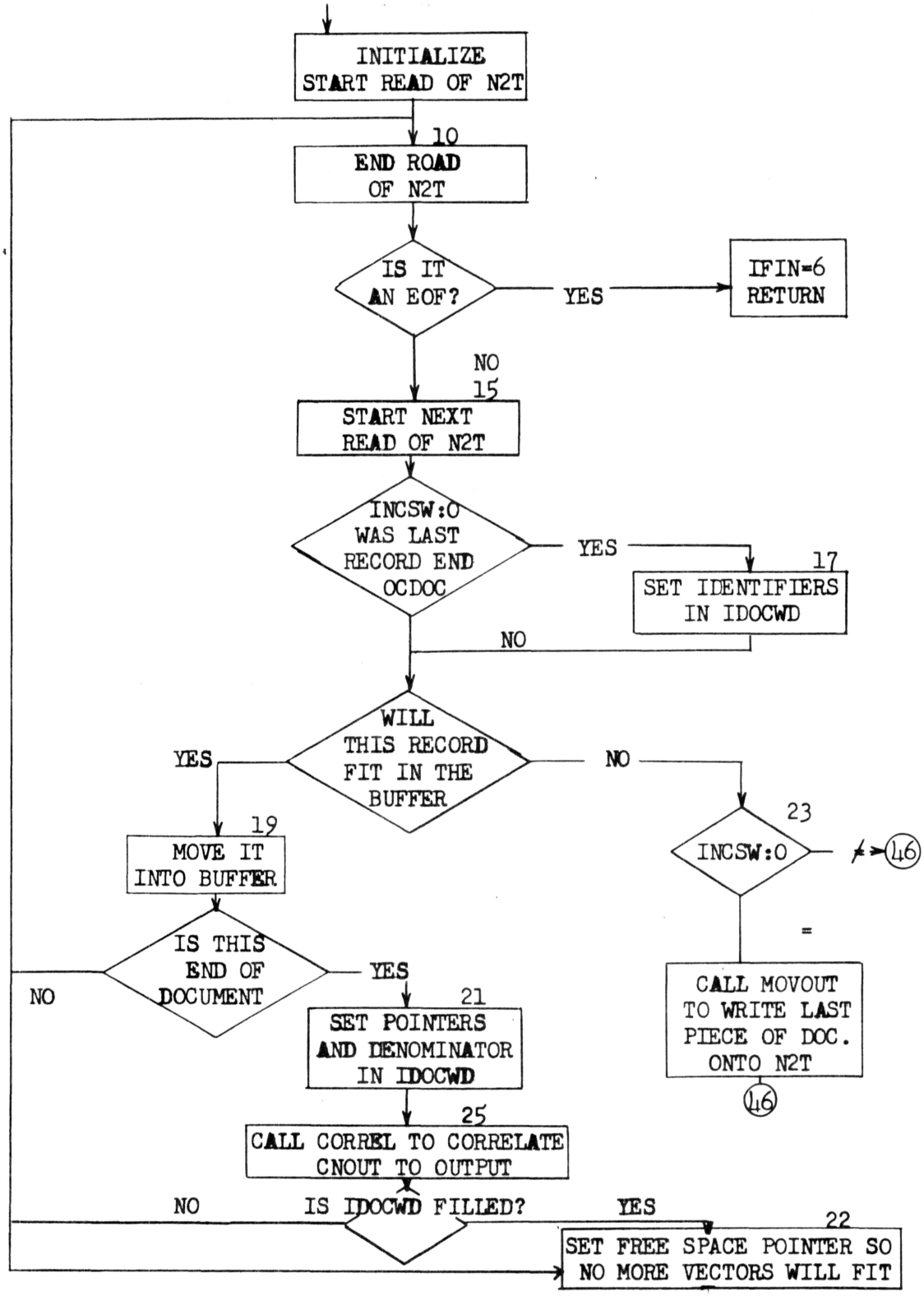
(2) The IDOCWD buffer is filled. Control then passes to box 40.

(3) The IVEC buffer is filled. Control is eventually transferred to the section beginning at box 40, which creates the vector tape for the next pass. First, however, it may be necessary to start this tape by writing out initial pieces of the vector that were put into IVEC. Subroutine MOVOUT accomplishes this.

The section of code beginning at box 40 is essentially the same as the code that preceded it, with the important difference that it also writes out the vectors. Each vector is written out in records not larger than 500 words. The first two words of each record contain the identifier for the vector; the third word contains the denominator if this is the last record of the vector; it is zero otherwise. The remaining words contain the vector entries.

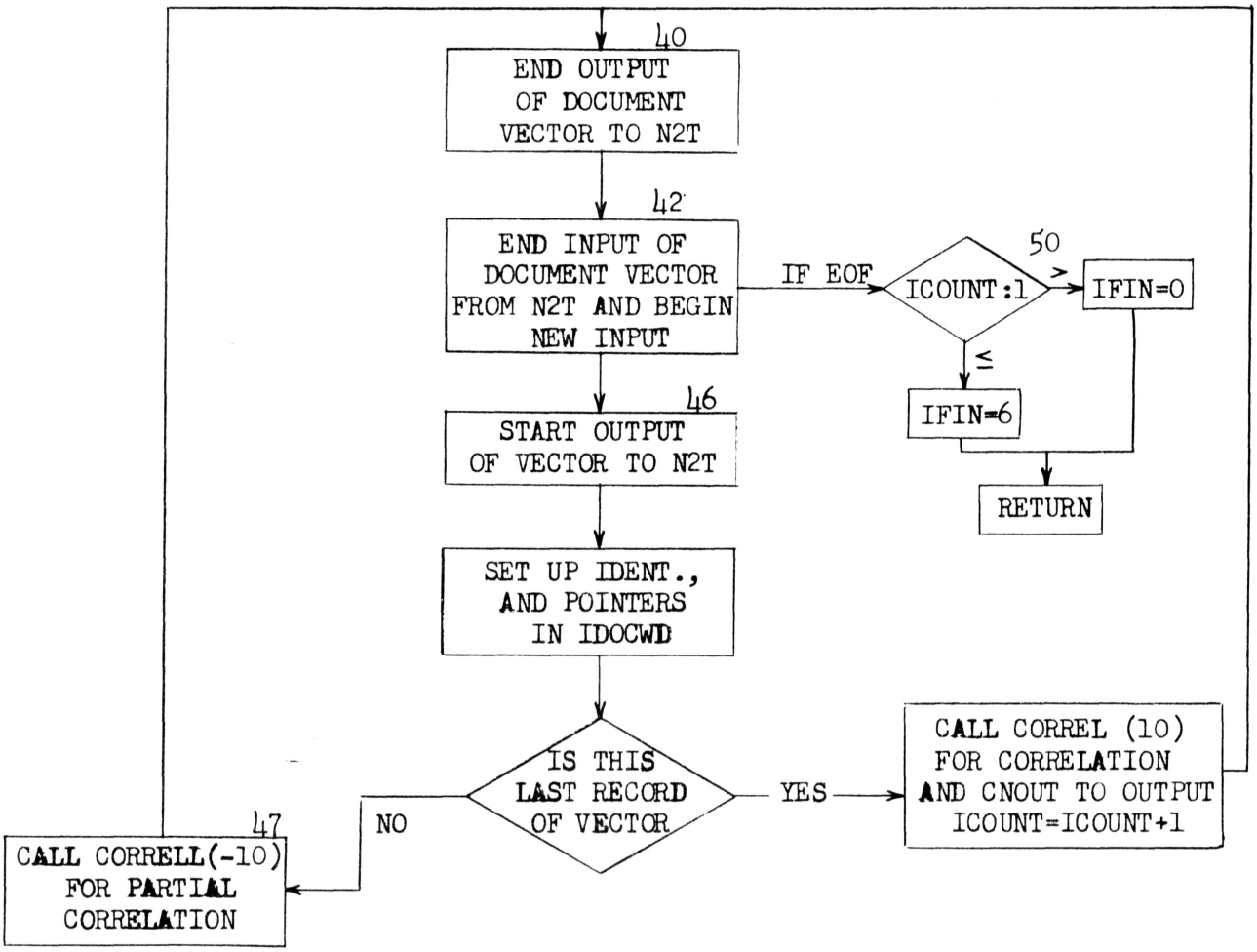
Subroutine CCPASS has four arguments — NIT, NOUT, and IFIN, corresponding to the arguments of CPASS1. CCPASS is very similar to CPASS1 as can be seen from Flowchart 3; the difference is that CCPASS has to deal only with the vectors in the form created by CPASS1. All entries are already set up, the denominators are all calculated, and so on.

All I-O is double buffered, and it makes use of the INOT trap controlled input-output routine.



SUBROUTINE CCPASS

FLOWCHART 3



FLOWCHART 3 (CONTINUED)

3. Concept-Concept Correlation

The input to the concept-concept correlation link is a tape (B1) filled with two word items: the first word contains a concept number and a weight, packed in decrement and address; the second word contains the document number in the decrement. There document numbers are created by link 8. The tape also contains entries for which the last bit of the address field of this word is "on." In this case, the first word contains the actual BCD name of the document.

This tape is sorted into concept order:
The document-concept matrix that was on the tape is then effectively transposed into a concept-document matrix. For each concept number LFETCH will return vector entries which contain the document number in the decrement and the weight in the address. Then vectors are then vectors of documents all identified by a given common concept. LFETCH ignores all items whose document numbers are less than IDCCUT. Since IDCCUT is set to the highest numbered request; the effect is to ignore all occurrences in requests.

The output routine CNOUT sets up two word items. The first contains the numbers of the two concepts whose correlated was higher than the cutoff value; these are packed two to a word. The second word contains the correlation coefficient. CNOUT double-buffers its output, only outputting when a buffer has been filled. CNOFIN is a final entry which writes out all items that are waiting in the buffer when correlations are completed.

After the concept-document tape has been completely correlated, the output tape is sorted into order according to the first word of each item

(that is, on the concept-concept word).

At this point it is possible to iterate the concept-concept correlations. The sorted output tape may be viewed as concept vectors over the space of concepts. When CFETCH is called with ITERCT greater than one, it expects the input tape to be a concept-concept tape. It takes the first half word (the first concept number) as the vector identifier, and the second half-word as the decrement of the vector entry. Each entry is assigned a weight of 1 (in the address field); thus the concept-concept matrix becomes a binary matrix. The results of each correlation are iterated in this manner until ITERCT equals CONCON. Control is then transferred to link 10.

4. Document-Document Correlation

The document-document correlation program attempts to find out which documents in the input collection are closely related by correlating their concept vectors. In practice, either overlap or cosine correlations are permitted. The main use of the document-document correlation is to expand request answers by including along with each document in the answer all documents which correlated with it above some fixed cutoff.

The basic algorithm used for document-document correlation is the same as that described in Part 3 of this section. The document-document programs are simpler in certain respects, the basic simplification being that the document-vector tape, as produced by the lookup, is already in sorted order so that no tape sort need be used initially. Also, no provision is made for iterating concept-concept correlations.

The document-document programs use the same routines CCPASS, CPASS1, and MOVOUT that are also used by the concept-concept programs. The programs CFETCH, LFETCH and CNOUT which they call must, however, be different since the formats involved are different.

The document-vector input tape contains 200 word records. Each word is of the form

CON NO.		WEIGHT

where CON NO. is a concept number and WEIGHT a weight. The individual document vectors are sorted on the concept number. The last word of a document vector is a word of zeroes, used as a flag. The last record of a document may be short. Each document is preceded by a 12-word record containing BCD identification. The versions of LFETCH and CFETCH used by the document-document programs accept this new format but otherwise they act in the same way as the versions earlier described. When a new document is started, LFETCH returns the second and third words of the BCD identifier as identification to the calling program. These two words are later written out by CNOUT with the correlation.

The output tape for the document-document correlations is written in records containing 100 five-word items. Each item looks like

wd	1	2	3	4	5
	DOCNM 1		Correl	DOCNM 2	

where correl is the two's complement of the correlation between documents

with identifiers DOCNM1 and DOCNM2. This output task is performed by a modified version of CNOUT. This program writes out only those items with a correlation above a user supplied cutoff.

After the correlation has been completed, the output tape is sorted by SPECTK into document order. The correlations are complemented so that the sort will order the documents correlating with a particular document in decreasing order of correlation. This sorted tape is then used to expand the request-document correlation tape. This correlation tape has the same five-word item format as that mentioned above. It is sorted into document order, a simple double-buffered merge is performed to combine it with the document-document tape, and the output tape is then sorted back into request order. The section of Flowchart 1 enclosed in dotted lines represents the supervisor program also for the document-document correlation. The merge mentioned above is completely straightforward. All sorts are done using the tape sort SPECTR.