SØCCER - A Concordance Program

Guy T. Hochgesang

1. Introduction

SØCCER* was written in response to a need for a program which could produce concordances of long texts. The program was written with three major objectives:

1. It should be sufficiently fast to permit concordances of moderately long texts.(30,000 - 300,000 words)
2. It should produce concordances in an easily-read format. "Tokens" or words in the concordance should be listed with as much context as possible to reduce the need for references to the original text.
3. It should be simple to use, requiring a minimum amount of effort to set up for each text.

Existing programs generally fail to meet one or more of these objectives. These programs usually do not satisfy the most crucial criterion, that of speed of processing. In order to make it economically feasible to produce concordances of moderately long texts, a more efficient program was needed. By using a balanced-merge tape sort with overlapped I-Ø, SØCCER achieves a significantly faster rate of processing than other concordance programs. It is believed that this speed has been achieved without sacrificing anything in the way of simplicity for the user or the utility of the concordance produced.

---

*SØCCER : Smart's Own Concordance Constructor, Extremely Rapid.

2. The Concordance

A. Definitions

SØCCER divides the standard character set into two groups: alphabetic characters and special characters. Alphabetic characters are defined to be the characters included in the concordance, while special characters are those characters to be ignored during the generation of a concordance. A token is defined to be any string of consecutive alphabetic characters delimited by special characters, while a type is defined to be a class of identical tokens. As an example if one defines the alphabetic characters as the letters of the alphabet, the string of characters to be or not to be contains the six tokens to, be, or, not, to, and be, but it contains only the four types to, be, or, and not.

B. The Input Text

The input text to SØCCER should be punched on cards in columns 1-72. Columns 73-80 of the cards are ordinarily ignored and may be blank or contain serial numbers. These cards must then be transferred to the INPUT tape in unblocked BCD records of thirteen or more machine words. No special typing conventions are necessary in punching the text cards. The end of the text must be indicated by a card with "*STØP" punched left-justified in columns 1-6, or by an end-of-file on the INPUT tape following the last card of the text.

C. Processing the Text

Before starting to process the input text, SØCCER first reads the control cards from A2. (An explanation of the control cards will be found in Part 4 of this report.) When the START control card is found, processing

of the text begins from the INPUT tape.

As the cards of the text are read in, they are numbered sequentially and then written out on the ØUTPUT tape to provide a listing of the text. Cards which have a "*" (the "space control character") in column one are listed double-spaced; i.e., a blank line precedes their listing. In addition, cards which have either a "*" or a "$" (the "skip control character") are not included in the concordance. In effect this causes cards with an asterisk or dollar sign in column one to be interpreted as comment cards.

Any card which does not have the skip control character or space control character in column one is included in the concordance. These cards are scanned for the tokens of the concordance in the following steps:

1. The right-most non-blank character is found. If this character is not a hyphen (a minus sign) step 2 is taken. If this character is a hyphen, the character and all blanks to the right of it are deleted. The next card is scanned from left to right for alphabetic characters, with the scan terminating at the first special character. These alphabetic characters (if any) are then appended to the card with the hyphen and step 2 is taken. This procedure allows one to hyphenate words from one card to another, provided that the hyphen follows immediately after the last alphabetic character on the first card and that the syllable on the second card starts in column one. Such hyphenated words appear in the concordance with the syllables properly joined together and the hyphen deleted.

2. If n consecutive blanks appear on the card, n-1 of these blanks are deleted to allow as much significant context as possible to be included with the tokens in step 3.

3. The card is then scanned for tokens. As each token is found it is written out on an intermediate tape, SMRTAP, along with

the context in which it occurs, and the number of the card in which it is included. If a restriction or selection list has been specified by the use of a RESTRICT or SELECT control card, this list is scanned before the token is written on the intermediate tape. If the RESTRICT card is used, tokens appearing on the restriction list are not written on SMRTAP; and if a selection list is specified by a SELECT card, only those tokens appearing on the selection list are written on SMRTAP. Thus the use of a restriction list provides a method for excluding common or unwanted words ("the", "of", "and" etc.) from the concordance, while the use of a selection list enables one to include only certain words in the concordance, excluding all others. Only one of the two types of lists may be active during a single run. It is permissible not to use either type of list, i.e., to use neither the RESTRICT or SELECT control card.

When SØCCER hits a card with "*STØP" left-justified in columns 1-6, or an end-of-file on the INPUT tape, the assumption is made that the entire text has been processed. The tokens appearing on SMRTAP are then sorted into alphabetical order, using the scratch tapes, as described in Part 3. After the sort of SMRTAP some control information about the sort is written on the ØUTPUT tape. This is further described in Part 6B. The concordance is then written on the ØUTPUT tape following the complete listing of the text already produced. The concordance consists of an alphabetical listing of the tokens on SMRTAP, along with the context in which the tokens originally occurred, and the number of the card from which the tokens were taken. The number of occurrences (i.e., number of tokens) of each type is also given. Figures 1 and 2 show fragmentary examples of a typical text listing and concordance.

While writing the concordance on the ØUTPUT tape, SØCCER keeps track of some useful statistics, which are written out following the concordance

CONCORDANCE OF ADI SHORT PAPERS FOR 1963 10/09/65 PAGE 116

THIS PHASE OF THE PROBLEM IS DEPENDENT UPON DEVELOP- 6454
MENT OF WIDER USE OF MICROPRINT MATERIAL AMONG OUR 6455
POTENTIAL AUDIENCE, TO INCREASE THE HABIT OF SUCH USE 6456

Excerpt from the listing of a text concorded by SØCCER.

(The line numbers to the right of each line are automatically provided. In Figure 2 below is listed the occurrence of "WIDER" in line 6455.)

Figure 1

CONCORDANCE OF ADI SHORT PAPERS FOR 1963 10/09/65 PAGE 1748

WIDER

DEPENDENT ON DEVELOPMENT OF WIDER USE OF MICROPRINT M 6455
BACK TO THE SYMBOL OR, IN A WIDER SENSE, TO SYMBOLIC 9803
TH,. HOWEVER, PRINTING WITH WIDER LINE SPACING CAN BE 10417
S CAN BE MATCHED TO 3 KW OR WIDER BAND WIDTHS UP TO T 12082
NUMBER OF OCCURRENCES 4

WIDESPREAD

ROLE .. THAT OF PREVENTING WIDESPREAD FOLLOWING OF F 671
OMPUTER TECHNOLOGY AND MORE WIDESPREAD DISTRIBUTION O 1294
AND CONCEPTS MAKE SWIFT AND WIDESPREAD COMMUNICATION 11767
NUMBER OF OCCURRENCES 3

WIDEST

CH IS CLASSIFICATION IN ITS WIDEST SENSE, IS THEREFOR 5040
BLE IN ORDER TO PROVIDE THE WIDEST POSSIBLE SYSTEM FL 8840
NUMBER OF OCCURRENCES 2

Excerpt from a concordance produced by SØCCER.

(The amount of context provided by SØCCER has been reduced to allow it to fit on this page. A normal run of SØCCER would produce about twice as much context for each token. The numbers to the right of each line refer to the number of the line of text in which the listed token originally occurred.)

Figure 2

listing. These statistics include a count of the number of tokens processed; the number of types found; the number of types with only one token, two tokens, three tokens, etc., up to the number of times each type occurred on the restriction (or selection) list. Finally SØCCER writes out the time it took to process the text and exits.

D. The Output Format

All of the output produced by SØCCER is written on the ØUTPUT tape in blocked BCD records of varying length. This tape includes a carriage control character for each line and is meant to be printed off-line with a printing program capable of handling blocked records and printer carriage control.

3. Tape Usage

A. Control Cards

The control cards for each run of SØCCER must be included as DATA cards on the system input tape, A2. SØCCER always reads these card images from A2 before beginning to process the text.

B. The INPUT, ØUTPUT, and SMRTAP Tapes

As explained in Part 2B, the INPUT tape contains card images of the text to be processed. This tape may be mounted on any tape unit except those used for the ØUTPUT and SMRTAP tapes. The most convenient unit to use for the INPUT tape may be A2. In this case the cards of the text are simply submitted as DATA cards immediately following the control cards on A2. The logical tape number of the INPUT tape is controlled by the INPUT control card.

The ØUTPUT tape contains the listing of the text and the concordance,

and is normally the system output tape, A3. The logical tape number of the ØUTPUT tape may not be changed from A3 to another unit without changing the source deck for SØCCER as described in Part 7.

SMRTAP is the tape used as an intermediate tape for SØCCER and is defined to be B2. Again this tape number may not be changed unless the SØCCER source deck is changed.

C. Scratch Tapes

The sort of SMRTAP done by SØCCER is a tape-merge sort requiring an equal number of scratch tapes on A-channel and B-channel. The number of scratch tapes required on each channel is equal to the order of the merge and is controlled by the ØRDER control card. Below is a table of the scratch tapes used by SØCCER:

Channel A	Channel B
A4	B2
A5	B3
A6	B1
A7	B5
A8	B6

If a merge-sort of order n is specified by the use of the ØRDER control card, the first n tapes on both channels A and B will be used as scratch tapes. (Example - a merge of order 2 would use A4, A5, B2, and B3.) The maximum order of the sort is five, while the minimum order is two. In general a high order sort will be faster than a lower order sort and consequently the processing will be faster.

Note that it is possible, by use of the INPUT control card, to specify an INPUT tape which will also be used as a scratch tape. For instance, the use of the control card INPUT 4 would specify the INPUT tape to be A4, which is also used as a scratch tape. In cases such as this, the text on the INPUT

tape will be destroyed, when the tape is also used as a scratch tape.

Examples of tape usage are given in Part 5.

4. Control Cards

As previously explained, the control cards for SØCCER must appear as DATA cards on A2, and are always read by SØCCER before processing. Each control card contains one control instruction, punched left-adjusted to column one, except for types to be entered into a restriction or selection list. These types are punched, one per card, left-adjusted to column 7 and must immediately follow the RESTRICT or SELECT control card. The START control card must be the last control card used. Further ordering of the control cards is not necessary. The START card must be used for every run of SØCCER but any of the other control cards may be omitted. A listing of the twelve control instructions is given below:

INSTRUCTIØN	FUNCTIØN
START	This card signals the end of the control cards and causes processing of the text to start immediately.
TITLE	The information punched in columns 7-60 of this card is used as a page heading, and will appear at the top of each page of the concordance with the date and page number. If this card is not used no page heading, date, or page numbers will be used.
IDEN	This card causes the serial number, punched in columns 73-78 of the text cards to be used to identify the lines of text in the text and concordance listings. If this card is not used, columns 73-78

of the cards of text are ignored. Instead the lines of text in the text listing are numbered sequentially and these numbers are used to identify lines of text in the concordance listing.

ØRDER — The order of the merge-sort is set equal to the integer punched in column 7 of this card. The order must be less than six and greater than one. If this card is not used the order of the merge is five.

INPUT — The logical tape number of the INPUT tape (the tape containing the text) is set to the integer punched left-adjusted to column 7 of this card. If this card is not used the input tape is A2 (logical number 5). The INPUT tape cannot be A3 or B2 (logical 6 or 2).

ALPH — The single character punched in column 7 of this card is defined to be an "alphabetic" character (see Part 2A). The alphabetic characters are initially defined to be the letters of the alphabet and will be interpreted as such if this card and the SPEC card are not used. If column 7 of this card is left blank, the character blank is defined to be an alphabetic character.

SPEC — The single character punched in column 7 of this card is defined to be a "special" character (see Part 2A). The set of special characters initially contains all characters which are not letters of the alphabet, and will remain as such if the ALPH and SPEC cards are not used. If column 7 of this card is blank, the character "blank" is defined to be an ordinary special character and the deletion of consecutive blanks described in Part 2C does not occur. If column 7 of this card contains

a hyphen (minus punch), a hyphen is defined as an ordinary special character and the hyphenation feature described in Part 2C is not in effect.

RESTRICT

This card specifies that the types punched on the cards that follow are to be included in a restriction list (see Part 2C). The types to be included in the restriction list are punched one per card left-adjusted to column 7 on cards following the RESTRICT card. Occurrences of these types in the text are not listed in the concordance. This card may not be used in addition to the SELECT card. If this card is not used, no restriction list is used for the text. As many as 205 types may appear on the restriction list.

SELECT

This card specifies that the types punched on the cards that follow are to be included in a selection list (see Part 2C). The types are punched one per card, left-adjusted to column 7 on cards following the SELECT card. Occurrences of only these types will be listed in the concordance. The SELECT card may not be used in addition to the RESTRICT card. If the SELECT card is not used, all types not on the restriction list (if any) are listed in the concordance. As many as 205 types may appear on the selection list.

SKIP

The single character punched in column 7 of this card is defined to be the "skip" control character (see Part 2C). All cards of the text with this character included in column 1 will be numbered and listed in the text, but will not be included in the concordance. A blank in column 7 of the SKIP card causes the skip control option to be deleted; i.e., no cards of the text are "skipped". If the SKIP card is not used, the skip control character is a

dollar sign ($).

SPACE — The single character punched in column 7 of this card is defined to be the "space control character" (see Part 2C). All cards of text having this character in column 1 are numbered and listed in the text preceded by a blank line, but are not included in the concordance. A blank in column 7 of the SPACE card causes the space control option to be deleted; i.e., no cards of the text are "spaced" over in this manner. If the SPACE card is not used the space control character is an asterisk or "star" (*).

NØDUMP — Before starting its tape sort, SØCCER normally dumps core storage onto tape B4 to make available as much room as possible for internal sorting. After the sort, the program restores core and returns tape B4 to its original position. If B4 is not available for this purpose the NØDUMP card should be used to suppress the core dump. Most installations use B4 as the system punch tape, and since B4 is restored by SØCCER after the core dump, B4 should be available. Using this core dump speeds up the tape sort for long tasks.

5. Examples of SØCCER Usage

Example A

a) Input Deck

```
Binary deck  for SØCCER (labeled "SCR")
Binary decks for SPECTR (labeled "SMER" and "CVRT")
Binary deck  for INØT   (labeled "INØT")
Binary deck  for CLØCK  (labeled "CLØCK2")
*^^^^^^DATA
TITLE SAMPLE SØCCER RUN
```

```
START
Deck of cards for text
*STØP
End-of-file card
```

b) Explanation

The INPUT tape is A2, since no INPUT control card is used, and therefore the text follows the two control cards as DATA cards. The tape sort is of order five, since no ØRDER control card is used, and the tapes A4, A5, A6, A7, A8, B2, B3, B1, B4, and B5 are used for scratch tapes as explained in Part 3C.

The title "SAMPLE SØCCER RUN" appears as a page heading for the concordance since the TITLE control card is used. The "START" card tells SØCCER to begin processing the text, while the "*STØP" card indicates the end of the text. No restriction or selection list is used.

Example B

a) Input Deck

```
Binary decks as in Example A
*^^^^^^DATA
INPUT^12
ØRDER^3
ALPH^^0
ALPH^^1
ALPH^^2
ALPH^^3
ALPH^^4
ALPH^^5
ALPH^^6
ALPH^^7
ALPH^^8
ALPH^^9
RESTRICT
^^^^^^ØF
^^^^^^THE
START
End-of-file card
```

b) Explanation

The INPUT tape is A7 (logical number 12) and a tape containing the text to be processed must be mounted on unit A7. The text must be the first file on this tape and must end with a "*STØP" card or an end-of-file. Since a tape sort of order three is specified, tapes on units A4, A5, A6, B2, B3, and B1 are used as scratch tapes.

The ten ALPH control cards imply that numbers are included in the concordance. None of the occurrences of the types "ØF" or "THE" will be listed in the concordance due to the RESTRICT control card.

6. Subroutines used by SØCCER

A. INØT

INØT is a double-buffered input-output routine which uses the data channel traps to allow simultaneous data channel transmission and central processing unit operation. This allows SØCCER to overlap input-output and CPU operations to such an extent that it becomes tape-limited in most of its processing; i.e., it can process text as fast as it can read and write the necessary tapes.

The FAP calling sequence to write a BCD record with INØT is:

```
TSX $ØDEC,4     write BCD entry point;
TSX TAPE,0      location of word containing FØRTRAN tape number;
TSX FIRST,0     address of lowest word in buffer;
TSX HALF,0      word count of one half of buffer;
TSX BLØCK,0     word count of block to be transmitted;
TSX RECØRD,0    record size.
```

Each call to ØDEC starts the transmission of a buffer-half of words to the specified tape unit, starting with the lower half of the buffer. When ØDEC returns, the results of the previous transmission are returned in the AC as follows:

prefix:	zero if normal transmission, -1 if a tape error, -2 if end-of-tape was hit;
decrement:	count of words transmitted;
tag:	zero;
address:	address +1 of last word transmitted.

The FAP calling sequence to read a BCD record is:

TSX $IDEC,4	read BCD entry point;
TSX TAPE,0	location of word containing FØRTRAN tape number;
TSX FIRST,0	address of lowest word in buffer;
TSX HALF,0	word count of one half of buffer;
TSX BLØCK,0	word count of block to be transmitted;
TSX CØDE,0	zero to read one record; one to read a block of records.

The first call to IDEC reads a record (or records, if CØDE does not equal zero) into the lowest buffer half and starts transmission of a second record (or block of records). Subsequent calls to IDEC finish the last transmission and start a new one. Exit data is returned in the AC as for ØDEC, except that a prefix of minus two indicates an end-of-file rather than an end-of-tape.

Since each call to ØDEC or IDEC finishes a transmission and starts a new one, a special entry point, IØEND, is needed to terminate a transmission. Exit data is returned as for ØDEC and IDEC, and the calling sequence is:

```
TSX $IØEND,4
TSX TAPE,0.
```

When ØDEC, IDEC, or IØEND return negative exit data for a transmission to or from a particular unit, they refuse all subsequent transmissions to or from that unit until the unit is reset by a call to IØRDY. A call to IØRDY causes any transmission for that unit to be dropped immediately. The unit is reset so that the next call to ØDEC or IDEC for that unit is treated as the first use of the unit. The calling sequence is:

```
TSX $IØRDY,4
TSX TAPE,0.
```

B. SPECTR

SPECTR[2] is a tape sorting routine) modified from an earlier program called SMERSH[3]) written by Michael Lesk of the SMART project. SMERSH in turn is a modified version of SWISH[4], written at the Mitre Corporation in 1962. SPECTR does a tape merge sort, using the data channel traps, with internal radix exchange sorting.

Because the efficiency of the sort depends highly on the amount of memory available, SPECTR uses the area between the program and common breaks for buffer space, and attempts to enlarge this area by dumping as much core as possible onto B4.

Sorting by SPECTR is done in two basic steps. First the input tape is read and converted into sorted sequences. The keys upon which items are sorted are converted according to the collating sequence given in the CVRT table before sorting, and are restored during the last pass. After the blocks of items are sorted they are written onto the first set of scratch tapes. The second step consists of merging the sorted sequences from one set of scratch tapes onto the other. These merge passes are repeated until the sorted sequences are merged onto one output tape.

The messages printed by SPECTR give the following information: the input tape unit, the scratch tape units, the occurrences of an EØF on the input tape, the number of items and records to be sorted, the number of merge passes and sorted sequences, the maximum record capacity (computed

---

2. SPECTR: Smersh Program Edited by Change Tape Reads
3. SMERSH: Stolen from Mitre, Extensively Revised Sort-Harvard
4. SWISH: Sorting With Incredible Speed-Hollerith

for a density of 800 bpi), the timing of the sort and merge passes, and the output tape unit.

In the FAP calling sequence to SPECTR, all arguments must be symbolic (i.e., they must be the address of a location containing the actual argument). In addition, all actual arguments are FØRTRAN integers unless otherwise specified.

The calling sequence is:

```
TSX $SMERSH,4    entry to sort a BCD tape; use SPECTR for a binary
                      tape;
TSX ITEM,0       number of words per item
                      + sign for core dump on B4
                      - sign for suppress dump;
TSX IREC,0       number of words per input record;
TSX ØREC,0       number of words per output record;
TSX P,0          = 0 for no padding of short output records
                 = 1 for padding of short output records;
TSX PAD,0        6 Hollerith padding characters
                      (if P = 0, then PAD = 0);
TSX ØRDER,0      order of merge;
TSX SMRTAP,0     logical number of input tape
                      + sign to ignore input tape after use
                      - sign to rewind input tape after use;
TSX FILES,0      number of input files
                      + sign for all files on same reel
                      - sign for each file on separate reel;
TSX S11,0        number of first tape in first set of scratch
                      tapes;
TSX S12,0        number of second tape in first set of scratch
                      tapes;
... ...          etc. for remaining tapes in first set;
TSX S21,0        number of first tape in second set of scratch
                      tapes;
TSX S22,0        number of second tape in second set of scratch
                      tapes;
... ...          etc. for remaining tapes in second set;
TSX KEYS,0       number of keys on which to sort;
TSX PKEY1,0      position relative to beginning of item of leftmost
                      character of first key;
TSX LKEY1,0      number of characters in first key;
TSX PKEY2,0      same as PKEY1 for second key;
TSX LKEY2,0      same as LKEY1 for second key;
... ...          etc. for remaining keys;
TSX ØTAP,0       output tape number
                      (will equal 0 if an error occurs).
```

C. CLØCK

CLØCK is a subroutine which reads the printer clock. The FAP calling sequence is:

```
TSX $CLØCK,4
TSX TIME,0
```

This call causes the time in tenths of minutes to be stored in location TIME as a FØRTRAN integer. The time in floating-point minutes and tenths of minutes is returned in the accumulator.

7. Some Details About the SØCCER Program

A. Source Deck Changes

Three sections of SØCCER are purposely written in such a way as to facilitate changes in the source deck. These are the program parameters, the tape assignments, and the character tables.

Cards SCR01210 to SCR01680 of the source deck define program parameters which control the format of the concordance listing. These parameters may be changed in accordance with the instructions given in the program documentation on the cards. In particular, if it is necessary to obtain an unblocked output tape for printing the text and concordance, the parameter MAXREC should be changed to "MAXREC EQU 22".

The tape assignments are controlled by cards SCR17420 to SCR17720. If it is necessary to change the scratch tapes or output tape to different units, this may be done by changing the appropriate cards.

The two tables of characters on cards SCR07950 to SCR08600, and SCR09670 to SCR10320 give the basic assignments of characters to the "alphabetic" or "special" character sets. If one wishes to change the

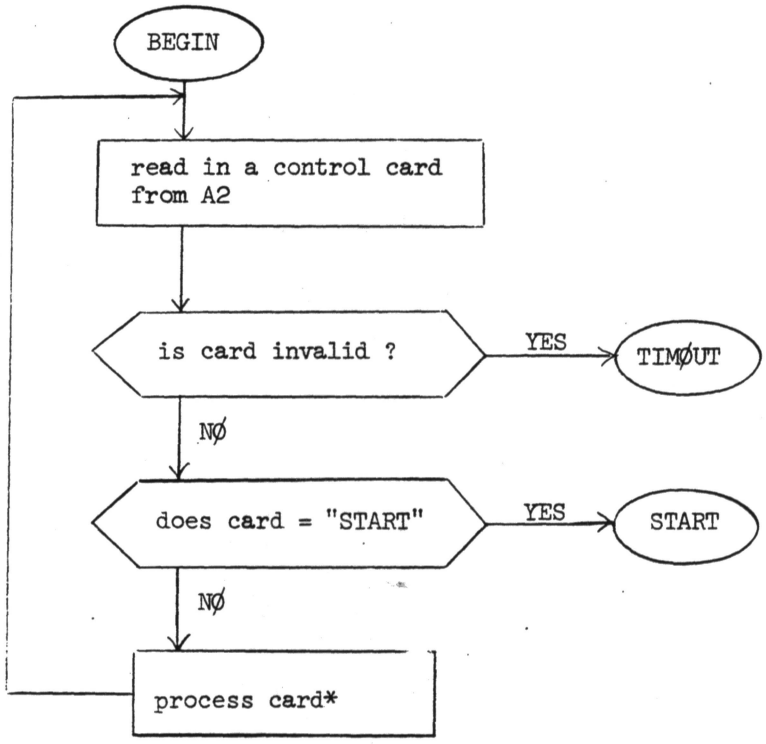
assignment of a character to one of these sets, and does not wish to use the ALPH or SPEC control cards for each run of SØCCER, these tables may be changed. Such changes are made by changing the transfer address for the appropriate character.

B. Timing

The time taken by SØCCER to process a text is dependent on the length of the text, the number of tokens included in the concordance, the tape assignments, and the order of the merge for the tape sort. In general it is faster to have the input tape on channel A and to use the highest possible merge order for the sort.

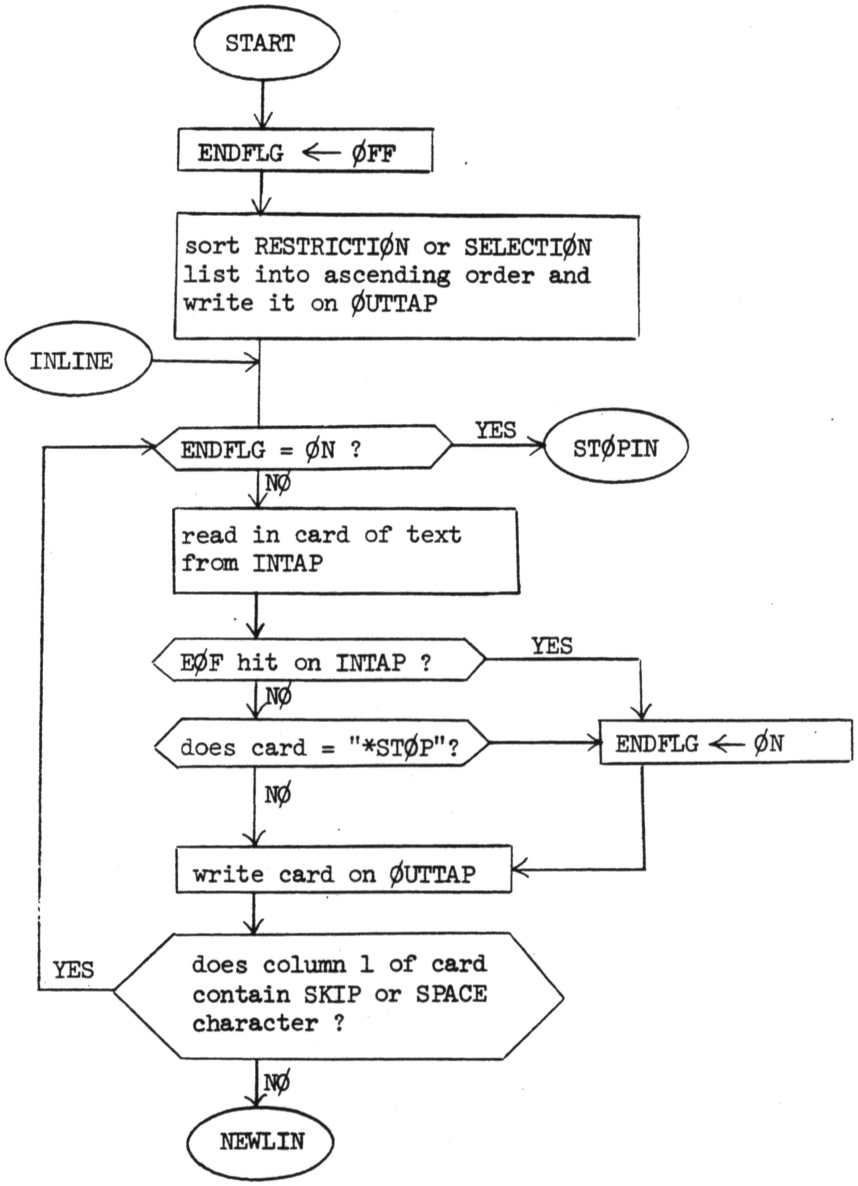
If one considers the merge order to be constant, then the processing time is roughly linearly dependent on the number of tokens in the text. Most of the runs of SØCCER to date have used a merge of order four and a restriction list which excluded between 40 and 50 per cent of the tokens from the concordance. Under these circumstances a text of 33,042 tokens (4318 cards) took 12.7 minutes of execution time, producing 597 pages of output. A text containing 113,130 tokens (13,356 cards) produced 1770 pages of output in 36.9 minutes of execution time, listing 7925 types containing a total of 58,190 tokens. The remaining 54,940 tokens were included on the restriction list.

Appendix

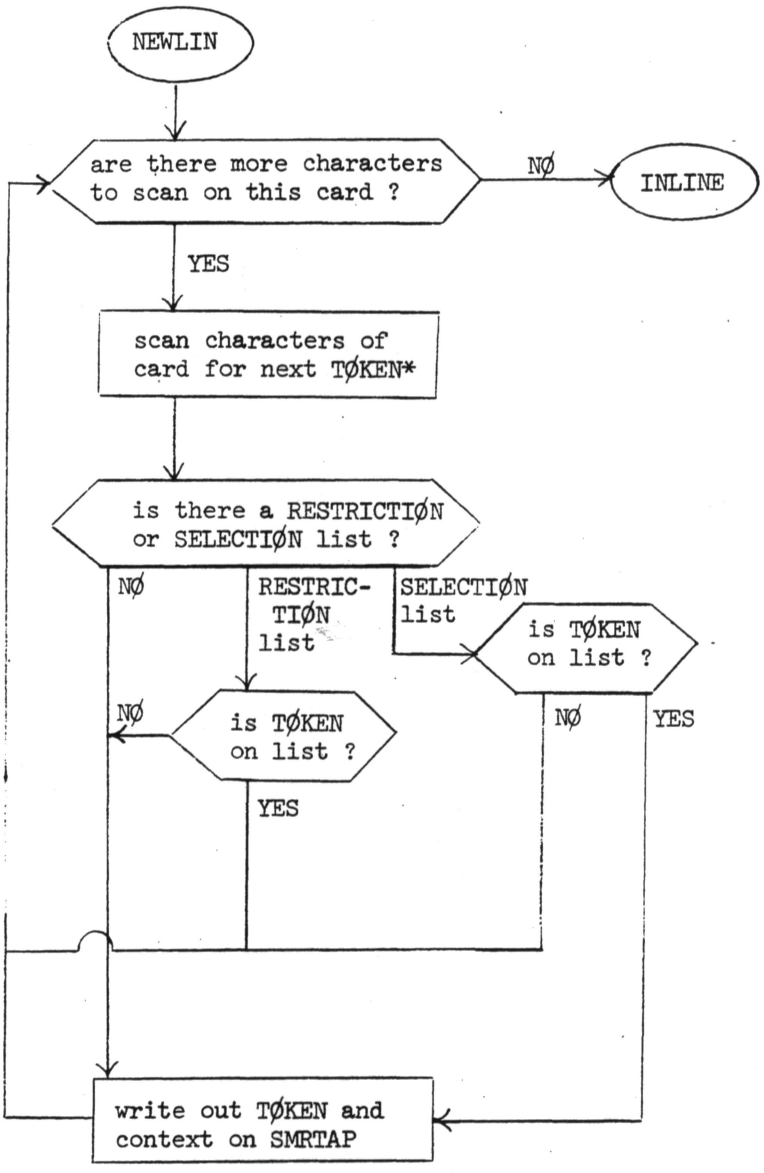


Flowchart of SØCCER Program

* See note on last page of flowchart

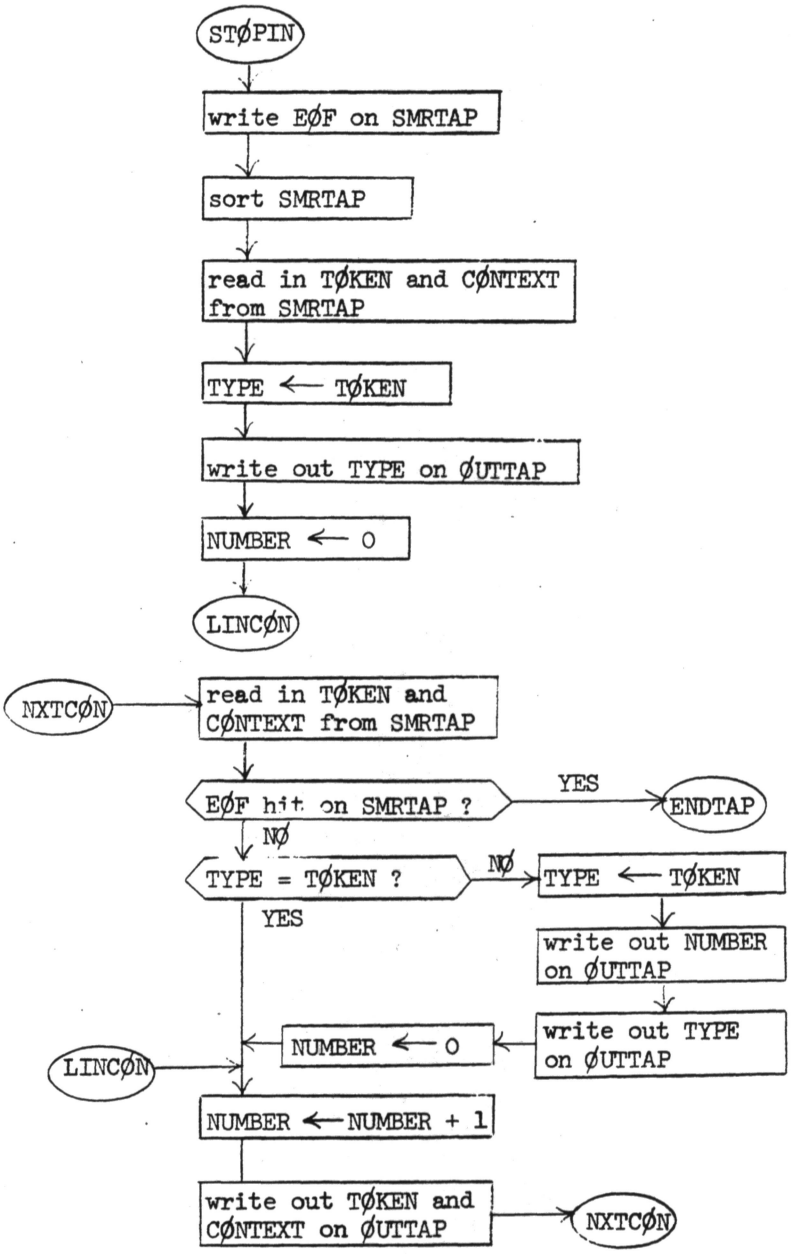


Flowchart of SØCCER (continued)

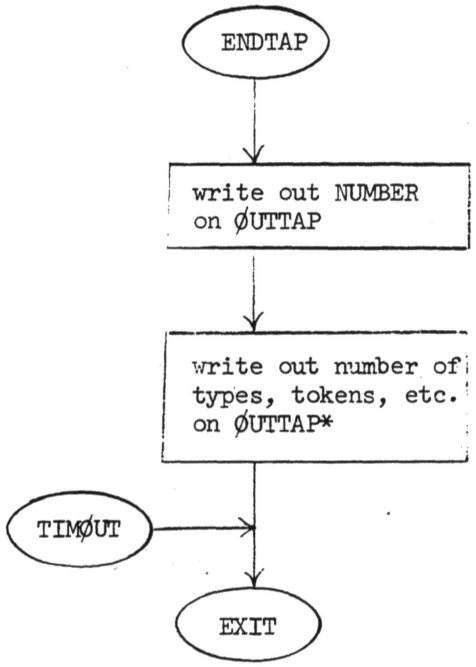


Flowchart of SØCCER (continued)

* See note on last page of flowchart



Flowchart of SØCCER (continued)



Flowchart of SØCCER (continued)

* Note: This flowchart represents only the main logic of SØCCER. Details of certain operations are omitted but are included in the documentation of the source deck.