Lib sc. 5; 1968; PAPER S.

# USE OF COMPUTER FOR THE SYNTHESIS OF CLASS NUMBER: A CASE STUDY WITH A FREELY FACETED VERSION OF COLON CLASSIFICATION.

(Non-conventional methods in document retrieval. 7).

A NEELAMEGHAN, *Documentation Research and Training Centre, Bangalore* 3, and S VENKATARAMAN*, *International Computers Ltd, Bangalore* 2.

[In an entry in the Catalogue-on-Tape (*See* Paper Q), slightly over 50 per cent of the space is taken up by the Feature Headings. Omitting the Feature Headings in the entries increased the number of entries scanned by the computer from about 6,000 per minute to 9,000 per minute. However, the provision of Feature Headings in the list of entries selected in response to a query facilitated scanning and picking out, by the reader, the entries for the most appropriate documents. To serve all these purposes, the Kernel Ideas of the subject of the document expressed in standard terms were fed into the computer when making the entry for a document forming the input. On the basis of a depth version of CC made available to the computer, it picked out and synthesised into a Class Number the corresponding Basic Facet number and Isolate numbers. Each entry in the Catalogue-on-Tape thus contained only the Class Number and the specification of the host document. The Class Number for the query was similarly synthesised by the machine, and compared it with the Class Numbers in the entries on the Catalogue-on-Tape. For each of the entries selected, the Basic Facet number and each of the Isolate Numbers in the Class Number is to be translated by the machine into the corresponding terms on the basis of the depth version of CC made available to it. These form the Feature Heading for the entry. The Steps and the Flow-charts for the synthesis of the

* Visiting Lecturer, DRTC, 1968.

Class Number alone are given. It is shown that, although classification as a whole is deemed an intellectual process, there are a number of steps which are of a clerical nature, and therefore, could be performed by the machine more efficiently and quickly. The particular advantage of a Freely-faceted version of CC in facilitating the synthesis of Class Number by the machine is pointed out.

## 0 Introduction

### 01 Scope of the Paper

In the experiments reported in the papers P, Q, and R in this issue, the Class Number for the subject of the input document and for the reader's query were constructed manually in the conventional way. This paper describes a feasibility study on the synthesis of Class Number by computer when Kernel Terms expressing a subject in the natural language are given. The Freely-faceted version of the Colon Classification (= CC) was used.

### 02 Phases of the Work

For convenience of flow-charting and programing, the work was divided into the following phases:

1 Preparation of the Schedule-on-Tape (*See* Sec 4);
2 Reading-in Kernel Terms (*See* Sec 5);
3 Distribution of Kernel Terms (*See* Sec 6);
4 Synthesising Class Number on the basis of Kernel Terms fed in random sequence (*See* Sec 7); and
5 Printing out Class Number (*See* Sec 8).

### 03 Towards an Integrated System

After the programs of the series of experiments are finalised, they can be conveniently linked up to develop an integrated system. The features of the system are described in Paper N, Sec 81.

## 1 Facility with CC Methodology

### 11 Isolating Clerical Job

Our general approach to the use of computer in document finding has been to pass on to the machine such jobs that require no judgment and therefore, are essentially clerical. Classifying is generally considered an intellectual work. A careful analysis of each of the processes involved in classifying documents would, however, indicate that some of the steps are clerical. One such analysis of the classifying process is provided by the Postulational Method (*See* also Paper R, Sec 23) (5). It will be seen that in Step 6 of the Postulational Method — that is, after the Kernel

Ideas of a document have been determined and arranged in a helpful sequence according to the Principles of Helpful Sequence and after the standard terms, as used in the schedules of the scheme for classification, have been used to name the Kernel Ideas — the next step is to look up the schedules of the scheme of classification to pick out the Basic Class Number and the Isolate Numbers and synthesise the Class Number using the appropriate Connecting Digits.

## 12 Schedule Look-up

It will be seen that the actual look-up of the schedules to pick out the number for each of the Kernel Terms does not involve judgment. If the schedules are fairly long, as they would be in a scheme for the depth classification of micro subjects, then even the look-up work to pick out the Isolate Numbers can be tedious and time-consuming for the classifier. This look-up work can, however, be done very fast by machine.

## 13 Sequence of Kernel Ideas

Determining the sequence among the Basic Subjects and among isolate ideas going with a Basic Subject is also an intellectual work. For, it involves the application of the Principles for Helpful Sequence. But in the CC System, the schedules themselves are constructed by applying these guiding principles. The Basic Subjects are arranged in a more or less helpful sequence according to certain guiding postulates. The facets and the isolates in the schedules for each of them are also arranged according to the Principles for Helpful Sequence. This gives a sequence of the Isolate Ideas that is helpful to the majority of the specialists in the subject concerned. The Basic Class Number and the Isolate Number simply fix the sequence determined in the idea plane and mechanise the arrangement.

## 14 Assigning Connecting Digits

Assigning the appropriate Connecting Digits to each of the Isolate Numbers involves intellectual work. However, if to each of the Isolate Numbers in the schedule, the appropriate Connecting Digit is already prefixed, the machine can pick up the Isolate Number along with its Connecting Digit. This facility is available with the CC System. In the CC System, a distinctive Connecting Digit is used for each isolate deemed to be a manifestation of one or other of the five Fundamental Categories. In the design of the schedule itself, the manifestation of each of the isolates is determined and assigned to the appropriate facet.

## 15 Presentation in the Schedule

In preparing the software for the computer, the above ideas have been made use of. For the subjects going with a particular Basic Subject, separate depth schedules are built up according to the methodology for the design and development of freely faceted schemes for depth classification developed in DRTC. In each schedule, the isolates have been patterned into the five Fundamental Categories — P M E S T — with the appropriate Rounds and Levels for each, wherever applicable. Thus, the schedules for [P] for the subjects going with a Basic Subject consist of schedules for [1P1], [1P2], etc. These are successively followed by a schedule for [MM] with the appropriate qualifiers for each of the isolates enumerated in it; the schedule for [MP] with the appropriate qualifiers for each of the isolates; the schedule for [E] followed by the appropriate schedules for [2P] for each of the [E] isolates; and the schedule for [S] consisting of schedules for [S1], [S2], etc, and the schedule for [T] consisting of schedules for [T1], [T2] etc.

In each of the schedules, the isolates are enumerated with the appropriate Connecting Digit prefixed to it. The Isolates in each schedule are given in the inverted sequence — that is, the sequence in which they will appear in the synthesised Class Number. Such a schedule has been put on a magnetic tape, called the Schedule-on-Tape, and made available to the computer to work with.

### 151 *Example of Schedule*

A specimen of a schedule for the classification of subjects going with the Basic Class MP85 Fountain Pen Production is given in Appendix 1.

## 16 Schedule Look-up by Computer

Consider a document with the following title:

Assembly of Japanese Pilot pens in India.

Reading through the document, we find that its subject is:

> "Assembly in India of Japanese Pilot pens, with Indian-made 14-carat gold nib, with irridium-tipped shorthand point, plastic barrel and silver cap, specially for the use of reporters".

The following Kernel Terms may represent the Kernel Ideas in the subject of the above document:

> Assembly, India, Fountain Pen, Pilot Brand,
> Japanese-make, Nib: Indian-make, 14-carat gold,
> Irridium-tipped, Shorthand point, Reporter-use.

These kernel terms, in random sequence, together with the bibliographical specification of the Host Document, are punched on cards and read by the machine. Although a few synonymous and near-synonymous terms for certain isolates may be given in the schedule, it is necessary

1 To use standard terminology in formulating the Kernel Terms; and

2 To present the words in a multi-worded Kernel Term in a predetermined sequence.

The objective here is to ensure correct matching of each of the Kernel Terms with the corresponding Isolate Term in the schedule. A few rules have been formulated for the purpose.

The machine compares each of the Basic Class Terms and Isolate Terms with each of the Kernel Terms fed in, starting from the beginning of the schedule. As the Schedule-on-Tape is presented in the Facet Structure sequence, the first term in the schedule to match with a Kernel Term will be the Basic Subject Term. The number against this term, which will be without a Connecting Digit prefixed to it, will be picked up. Then the machine scans the schedule for [1P1] for the subjects going with this Basic Class. Each of the Isolate Terms in [1P1] is compared with the Kernel Terms excepting the Basic Subject Term, which has already been disposed of. The Isolate Number against each of the Isolate Terms matching with the Kernel Term is picked up along with the Connecting Digit prefixed to it. In the case of isolates in [1P1] it will be a 'hyphen'. The machine then successively scans the schedule of [1P2]; then the schedule of [MM] and so on, until the corresponding Isolate Numbers for all the Kernel Terms have been picked up and assembled to form the Class Number. The assembly of the Basic Class Number and the Isolate Numbers will be in the sequence in which they are picked up by the machine. In a majority of the cases, the isolates will be those occurring in [1P1].

## 17 Provision for Devices of the CC System

CC uses devices such as the Chronological Device, Alphabetical Device, Geographical Device, and Subject Device. These Devices increase the hospitality of the system, help to conform to the principle of Consistent Sequence, facilitate the use of mnemonics, and considerably thin down the schedule. In the experiments carried out, programs are being drawn up for the computer to form Isolate Numbers according to the Chronological Device, Alphabetical Device, Geographical Device, and Numerical Device. While the program for the provision of the Subject Device can be worked out, it will have practical application only after all the schedules or at least the schedule of Basic

Classes of the CC system have been put on tape for the computer to use. This paper does not give the flow-chart, etc for the construction of Isolate Numbers according to those Devices. These will be reported in a later paper.

## 2 Essentials of the System

### 21 Analysis of Subject

The subject of a document, such as an article in a periodical forming the input to the system, is classified according to the Postulational Method, Steps 1, 2, and 5. This consists in picking out the Kernel Ideas including those that are implied, but not explicitly stated — such as, the Basic Class with which the subject may go — after studying the document (Step 1), eliminating the puffs and auxiliary words (Step 2), and expressing the Kernel Ideas in standard terminology used in the schedules of the scheme for classification (Step 5). The latter can be done conveniently if there is a print-out of the schedule. Provision has, however, been made for some of the possible synonyms of the Isolate Terms to be enumerated in the schedule itself. An alternative would be to build in a thesaurus. This would be convenient if the subjects of the schedule and of the queries of the readers are likely to be confined to the subjects going with a Basic Subject or with a few related Basic Subjects.

### 22 Format of Entry

The bibliographical specification of the Host Document — name of author, title of article, and abbreviated title of the periodical, number of the volume, issue number, year, inclusive pagination, and indication of the language of the original article — are also included in the entry. These are punched on card according to a pre-determined format and fed to the computer. The computer reads the Kernel Terms and constructs the Class Number for them on the basis of the built-in schedule. It stores on magnetic tape the Class Number and the bibliographical specification of the Host Document — that is, Kernel Terms are not included in the entry. A print-out can be made, if desired.

### 23 Query Formulation

#### 231 *Subject Analysis*

The reader's query on a subject is, for various reasons, usually likely to be of greater extension than what he is probably interested in at the moment. Therefore, it is necessary to make it co-extensive with his subject of interest *at the moment* such that the document selection is pin-pointed. This is, at present, done through a reader-librarian dialogue. Displaying to the reader the relevant portions of a well-constructed schedule used in the classification of the subjects of the documents forming the input to the system as a help in the reader-librarian dialogue stage has been discussed in another paper (2). The computer can be programmed to provide such a display of the schedule either by means of print-out or on a screen (1). But this would be a costly method at present.

The query is then facet-analysed according to the Postulational Method, Steps 1, 2 and 5 and the Kernel Terms formulated in the standard terminology of the schedule as was done for the subject of the input document (*See* Sec 21).

232 *Example*

Query.— Required a classified list of all documents on Pilot Fountain Pens with plastic barrel, and fine-point nib.

Conditions of acceptance of the documents selected:

1 All the isolate ideas 'Pilot', 'Plastic barrel' and 'Fine Point nib' should concurrently qualify 'Fountain Pen'.

2 Any other aspect of Fountain Pen Production may be dealt with in the documents.

3 A print-out in the Long Format.

233 *Construction of Class Number*

The Kernel Terms are: Fountain Pen, Pen brand=Pilot, Plastic barrel, Fine Point. The Kernel Terms are then punched on a card. The machine reads the card and constructs the Class Number for the subject of the query on the basis of the built-in schedule in the same manner as discussed in Sec 16.

24 Search and Out-put

The Class Number constructed for the query is then compared, Isolate Number by Isolate Number, with the Class Number of the entries in the Catalogue-on-Tape (*See* Paper R). In the integrated system (*See* Sec 03) each Isolate Number in the Class Number of the documents selected is to be translated into terms in the natural language by a schedule look-up. This is not discussed in this paper. The output is discussed in Paper R.

**3 Advantage**

31 Saving in Search Time

In the initial experiments, the entry for each input document consisted of the (CN), Feature Heading, and Specifications of the Host Document. With a maximum of 500 characters per entry a reel of magnetic tape of 2,400 ft can take approximately

36,000 entries, excluding other data. The machine takes about six minutes to scan this length of tape. In other words, about 6,000 entries are scanned per minute. It was noted that on an average, a little over 50 per cent of the characters in an entry was taken up by the terms in the Feature Heading. In the second series of experiments, the machine, after synthesising the Class Number on the basis of the Kernel Terms fed in, eliminated the Feature Headings. Thus, the number of entries stored per reel of tape was increased by about 50 per cent — that is, to about 54,000. The scanning time per reel remaining the same (6 minutes), the number of entries scanned per minute was increased to 9,000. While the Feature Headings in the entry for the input document is omitted, its helpfulness in giving the reader browsing facility can be had in the output.

## 32 Saving of Classifying Time

In classifying a subject of a document according to Postulational Method, Steps 1, 2, 3, and 6 will be the same both for a manual system as well as in feeding the computer with the Kernel Terms. However, Steps 4 and 5 are very important and take time because, the manifestation of each of the Kernel Ideas has to be determined and the Kernel Terms arranged in a helpful sequence according to the Principles for Helpful Sequence. In feeding the Kernel Terms into the computer, this need not be done. Advantage was taken of the fact that in the CC system, the schedule itself is built on the basis of these postulates and principles, such that in the synthesised Class Number the ideas represented by the Basic Class Number and by each of the Isolate Numbers will get automatically arranged according to the Principles for Helpful Sequence. This means a great saving in the classifying process, especially when one considers that there will be a larger number of isolates in a micro subject.

## 33 Help to Classifier

At the level of macro documents, the number of facets and isolates incident in a subject may only be two or three on an average. But in a micro subject, the number of isolates incident in a subject going with a Basic Subject may range from 5 to over a hundred, with an average of about 30. Looking up the schedule and collecting the Isolate Number for each of the isolate ideas and then synthesising the Class Number can be a time-consuming job. In the computer, on the other hand, this can be done very fast. It is possible to synthesise the Class Number for four or five different subjects almost simultaneously by feeding in the appropriate sets of Kernel Terms for the different subjects.

34 FEATURE HEADINGS IN PRINT-OUT

The program provides for different kinds of format for the print-out of the entries for documents selected in answer to a query. In the Longer Format, the Class Number, the Feature Headings, and the specification for the Host Document are given. Although the Feature Headings are not given in each of the entries for the documents forming the input to the system, the machine can provide a translation into natural language the Basic Class Number and each of the Isolate Numbers in the Class Number by looking up the built-in schedule. This is a process somewhat in reverse to that of synthesising a Class Number on the basis of the Kernel Terms fed in. The Feature Heading is of help to the reader in scanning a documentation list, to pick out the entries for the document most likely to be of interest to him at the moment. The details will be published in a later paper.

35 BROWSING FACILITY

Although the reader-librarian dialogue would have helped to narrow down the query to a considerable extent, there may still be the need to help the reader in the choice of the documents, particularly when the number of documents selected in answer to a query is large, say, more than fifty. A conventional method of helping the reader is to provide the facility for browsing among the entries in the list of documents selected. This involves the arrangement of the entries in a filiatory sequence. Such a sequence can be achieved by arranging the entries according to the CC Class Numbers assigned to them. As mentioned in Sec 34, the Feature Heading is also of help in browsing. The computer can be programmed to arrange the entries by Class Number.

36 "MECHANICAL" ABSTRACT

In providing Feature Headings — a translation in the natural language for a Class Number of a document — the machine actually gives an indicative abstract for the subject of the document. Depending upon the depth of analysis of the subject of the document, this abstract can be made as informative as possible. It has been shown (4) that with the facility for classifying in great depth according to the Freely-Faceted Analytico-Synthetic version of CC, such an indicative abstract of a document can be achieved. At present, this has been achieved with subjects dealing with Commodity Production. A method for breaking up a multi-focal document into sub-units to provide for the appropriate slant has also been demonstrated (3).

# 4/8 COMPUTER OPERATIONS

## 4 Preparation of Schedule-on-Tape

### 41 Objective

The objective was to transfer the manually prepared schedules of the scheme for classification of the subjects going with the Basic Subject Fountain "Pen Production" on to a magnetic tape. The Isolate Numbers, Isolate Terms etc were punched on card. The computer read each card checked the correctness of its sequence in the deck, and stored on a magnetic tape the data punched in it. The Schedule-on-Tape could be amended when required by using the ICL standard software program.

### 42 Presentation of Schedule

The lay-out of the schedule has been discussed in Sec 15. It is illustrated in Appendix 1.

### 43 Punched Card Format

The following format was used for punching the Isolate Numbers, Isolate Terms, etc.

| Column | Details |
|---|---|
| 1 to 8 | Isolate Number, punching to start from col 1 |
| 9 to 40 | Isolate Term allowing for a maximum of 32 characters per term |
| 41 to 42 | Blank |
| 43 to 44 | Reserved for Symbols to indicate to the computer to construct the Isolate Number either according to one of the devices of CC or to take the Isolate Number from the Device Tape. |
| 77 to 88 | Serial Number of card |

### 44 Sorting of Cards

The punched cards were sorted by the Serial Number using the Sorter. When the number of cards is large, the sorting can be done by the computer and the data on the cards transferred on to magnetic tape.

### 45 Flow-chart

Fig 1 in Appendix 2 gives a flow-chart indicating the steps in the operation. The steps are described in Sec 4A to 4J. The symbols CARD and SER represent the storage locations in the computer for the data read in from col 1 to 44, and the serial number from col 77 to 80 respectively, from the punched card. For convenience, the steps are marked S1.1, S1.2 etc, in the flow-chart. The number of cards read per minute was 300.

4A ZEROISING SER A (S1.1)

The location SER A was zeroised.

4B READING A CARD (S.2)

The computer read a card. The data punched in columns 1 to 44 were stored in CARD to CARD + 10, and those of columns 77 to 80 in SER B.

4C CHECKING FOR LAST CARD (S1.3)

Examined whether the card just read-in was the last card. If it was the last card, the program branched to Step S1.8. If it was not the last card, it proceeded to Step S1.4.

4D CHECKING FOR SEQUENCE (S1.4)

The number 1 was added to SER A, which was initially a zero. SER A+1 was compared with SER B. If SER A + 1 was not equal to SER B, the program branched to Step S1.9. If they were equal, it proceeded to Step S1.5.

*Annotation.*— The first card must bear the Serial Number 1. If it did not, it indicated an error in sorting.

4E TRANSFER TO TAPE (S1.5)

The contents of CARD to CARD + 10 were transferred on to magnetic tape.

*Annotation.*— CARD to CARD + 10 will contain the Isolate Number, the Isolate Term, and any symbol to instruct the computer to use the Device Tape, etc.

4F PRINT-OUT (S1.6)

The data transferred on to the magnetic tape was also printed out on the on-line printer for use as a check list.

4G UPDATING SERIAL NUMBER (S1.7)

The data in SER B was transferred to SER A, for use in Step 4 (S1.4) in the next cycle of operations. The program was then re-entered at Step S1.1.

4H REWINDING TAPE (S1.8)

When the last card was sensed in Step S1.3, the computer rewound the tape after an "End Sentinel" was written on it.

4J Error Condition (S1.9)

When SER A + 1 was not equal to SER B in Step S1.4, the computer indicated an error in the sequence of the punched cards.

## 5 Reading — in Kernel Terms

51 Objective

In Sec 22, it has been mentioned that the input entry consisted of Kernel Terms in standard terminology used in the schedules of the scheme for classification and the specification of the Host Document. It has also been mentioned in Sec 23 that the reader's query was facet-analysed and each kernel idea represented by a Kernel Term. To facilitate the synthesis of the Class Number on the basis of these Kernel Terms in the input entry and in the query, the Kernel Terms were first read into the computer and distributed.

52 Punched Card

The Kernel Terms in each input entry or query as the case may be were punched on cards. An input entry or a query could have up to 40 terms of 32 characters per term. This provision can be extended to include more number of Kernel Terms. The terms were punched continuously in columns 1 to 76 of the card, without intervening blanks. The terms were separated from each other by a comma (,). The last term was suffixed with a virgule (/) immediately after the comma (,). The Serial Number of the card was punched in the columns 79 and 80. The last card of each group had two asterisks (**) punched in columns 77 to 78.

53 Sorting

The cards were sorted in the sequence of the Serial Numbers.

54 Storage of Data

The data from the punched cards were read by the computer and stored in locations CARD to CARD + 330. This storage capacity was calculated as follows:

| Maximum Data to be Stored | N of Word Spaces |
|---|---|
| 40 Kernel Terms of 32 characters each | 320 |
| 40 Commas | 10 |
| 1 Virgule | 1 |
| Total | 331 |

### 55 Flow-chart

Fig 2 in Appendix 3 gives a flow-chart indicating the steps in the operation. The steps are described in Sec 5A to 5J. For convenience, the steps are marked S2.1, S2.2 etc, in the flow-chart. Provision has also been made in the program to check that the total number of characters in the Kernel Terms did not exceed the limits imposed by the program.

### 5A Zeroising SER A (S2.1)

The location SER A was zeroised.

### 5B Filling up CARD with Space (S2.2)

The locations CARD to CARD + 330 were filled with space.

### 5C Setting Value of Modifiers (S2.3)

The values of m and n used as Address Modifiers were made equal to 0 and 18 respectively.

*Annotation.—* The data from the first card would be stored in CARD to CARD + 18, that from the second in CARD + 19 to CARD + 36, and so on.

### 5D Reading-in a Card (S2.4)

The computer read-in a card. The data from columns 1 to 76 were stored in CARD(m) to CARD(n), those from columns 77 and 78 in CHK, and those from columns 79 and 80 in SER B.

### 5E Checking for Sequence (S2.5)

The number 1 was added to SER A, which was initially a zero. SER A + 1 was compared with SER B. If SER A + 1 was not equal to SER B, the program branched to Step S2.9. If they were equal, it proceeded to Step S2.6.

*Annotation.—* The first card must bear the Serial Number 1. If it did not, it indicated an error in sorting.

### 5F Checking for Last Card (S2.6)

Examined whether the card just read-in was the last card by checking whether CHK was equal to ** (2 asterisks). If it was the last card, the program branched to the next Phase S3. If it was not the last card, the program proceeded to Step S2.7.

### 5G Setting Values of Modifiers (S2.7)

The values of m and n were each increased by 19.

*Annotation.—* This facilitated the storing of the data read-in from the next card in the next 19 word-spaces.

5H UPDATING SERIAL NUMBER (S.28)

The data in SER B was transferred to SER A, for use in Step S2.5 in the next cycle of operations. The program was then reentered at Step S2.4.

5J ERROR CONDITION (S2.9)

When SER A + 1 was not equal to SER B in Step S2.5, the computer indicated an error in the sequence of the punched cards.

## 6 Distribution of Kernel Terms

61 ALLOCATION OF STORAGE AREA

The contents of locations CARD to CARD + 330 were rearranged as separated fields in an area called TABL, to facilitate comparison of each of the Kernel Terms with each of the Isolate Terms in the schedule. As a term may have a maximum of 32 characters in it, an 8-word storage location was required for it in TABL. An additional word space was also provided to store the indicative symbols included in the schedule. Thus, a total of 369 words — TABL to TABL 368 — were reserved to hold the Kernal Terms. As the number of terms for each document was variable, the first word of the TABL following that of TABL for holding the last term in a set of terms, was filled with $$$$ (4 dollar signs).

62 FLOW-CHART

Fig 3 in Appendix 4 gives a flow-chart indicating the steps in the operation. The steps are described in Sec 6A to 6K. For convenience, the steps are marked S3.1, S3.2 etc in the flow-chart.

6A FILLING TABL WITH SPACE (S3.1)

The locations TABL to TABL + 368 were filled with spaces.

6B SETTING VALUE OF MODIFIERS (S3.2)

The values of M, N, and n were each made equal to 0.

*Annotation.*— M indicated the character address in CARD, N the character address and n the starting word address in TABL.

6C CHECKING FOR LAST TERM (S3.3)

Checked whether CARD(M) was equal to a virgule (/). If it was, the program branched to Step S3.9. If it was not, it proceeded to Step S3.4.

*Annotation.*—When a virgule (/) was sensed in this step, it indicated that there were no more terms to distribute.

**6D Checking for Comma (S3.4)**

Checked whether CARD(M) was equal to a comma (,). If it was, the program branched to Step S3.7. If it was not, it proceeded to Step S3.5.

*Annotation.*— It will be remembered that the Kernel Terms were separated from each other by a comma.

**6E Transfer to TABL (S3.5)**

The character sensed in Step S3.4 was transferred to TABL(N) (n).

*Annotation.*— As the initial values of N and n were each zero, the first character will be stored in the 0th character position of TABL.

**6F Increasing the Value of M and N (S3.6)**

The values of M and N were each increased by 1. The program was then reentered at Step S3.3.

**6G Updating the Value of M (S3.7)**

When a comma (,) was sensed at Step S3.4, the value of M was increased by 1.

*Annotation.*— The comma was not stored in TABL. Therefore, the value of M was increased by 1 to check the next character of CARD.

**6H Setting Values of Address Modifiers (S3.8)**

The value of N was made equal to 0 and that of n increased by 8. The program was then reentered at Step S3.3.

*Annotation.*— The next character, if not a virgule (/), is to be stored in a separate field. The starting character address for any field is 0. Hence the value of N was set as 0, and that of n increased by 8, indicating the starting address for the next term. Each term had a nine-word area in the TABL — that is, TABL(0) to TABL(8).

**6J Setting the Value of Address Modifiers (S3.9)**

When a virgule (/) was sensed at Step S3.3, the value of N was made equal to 0 and that of n increased by 8.

**6K Storing End-of-Field Symbol (S3.10)**

$$$$ (4 dollar signs) were stored in the positions 0, 1, 2 and 3 of TABL(n). The program then entered that next Phase S4.

*Annotation.*— When a virgule (/) was sensed in Step S3.3, it indicated that there were no more terms for distribution. The 0th word of the following term was filled with $$$$ to avoid unnecessary search in the next phase of the program.

6L ILLUSTRATION

The distribution of the kernel terms in TABL is shown in Fig 4 in Appendix 5.

## 7 Synthesis of Class Number

71 OBJECTIVE

In this phase an entry from the Schedule-on-Tape was read-in and stored in a location called BUF to BUF + 9 in the computer. BUF 0 and BUF + 1 would contain the Isolate Number and BUF + 2 onwards the Isolate Term (*See* Sec 43). This was compared with the fields in TABL. When an equality was found between the Isolate Term and the Kernel Term, the Isolate Number in BUF 0 and BUF + 1 was stored in a location called CN, starting from CN. The spaces in the Isolate Number location were ignored as the punching always started, with column 1 to form character 0 of BUF(0). When an equality was not found, the next entry in the Schedule-on-Tape was read and the process of comparison repeated. The last word — that is TABL(8) — was filled with four asterisks (****) to indicate that no further comparison with this term was to be done. A full-file search was done, the end being signalled by the End Sentinel on the tape. When this was encountered the program entered the next phase.

72 STORAGE AREA

A maximum of 40 kernel terms each with a maximum of 8 characters would require 80 words of storage in the computer. This location was called CN.

73 FLOW-CHART

Fig 5 in Appendix 6 gives a flow-chart indicating the steps in the operation. The steps are described in Sec 7A to 7V. For convenience, the steps are marked S4.1, S4.2 etc, in the flow-chart.

7A FILLING CN WITH SPACE (S4.1)

The locations CN to CN + 79 were filled with spaces.

7B SETTING VALUE OF MODIFIERS (S4.2)

The modifier P for character, and p indicating starting address in CN, were each made equal to 0.

7C SETTING UP VALUE OF MODIFIER OF TABL (S4.3)

The value of m was made equal to 0 and that of n to 2. The value of C, which was later used as a counter, was made equal to 0.

*Annotation.*— The Isolate Term would occupy word 2 onwards in BUF.

7D READING-IN FORM SCHEDULE-ON-TAPE (S4.4)

The computer read-in an entry from the Schedule-on-Tape and stored in BUF to BUF + 9.

7E CHECKING FOR END SENTINEL (S4.5)

Checked for End Sentinel. If it was found, the program branched to the next phase. If not, it proceeded to Step S4.6.

7F CHECKING FOR LAST TERM IN TABL (S4.6)

Checked whether word 0 in TABL following that of the last term had four dollar signs ($$$$). If they were found, the program was re-entered at Step S4.3. If they were not, it proceeded to Step S4.7.

*Annotation.*— At this stage, in the first cycle of iteration, the value of m would be 0. Hence, the location checked by the computer would be word 0 in TABL.

7G UPDATING MODIFIER (S4.7)

The value of m was increased by 9.

*Annotation.*— The last word in each field of TABL would contain 4 asterisks (****). If an Isolate Number had been found for the kernel term, the value of m at this stage in the first cycle would be 0.

7H CHECKING FOR FOUR ASTERISKS (S4.8)

Word in TABL(m) was checked for four asterisks. If found, the program branched to Step S4.21. If not, it proceeded to Step S4.9.

7J DEMODIFYING THE ADDRESS INDICATOR (S4.9)

The value of m was reduced by 8, to indicate word 0 of TABL.

*Annotation.*— This step would be entered if word 0 of TABL was not having $$$$ and word 8 was not having **** indicating that a comparison should be made in this part of the TABL. The value of m at this stage in the first cycle would be 8. The Kernel Term in BUF would be located from BUF + 2 onwards. This would be compared word by word with that in TABL 0 to 7.

7K TEMPORARY STORING OF THE VALUE OF m (S4.10)

The current value of m was stored in a location called A for use at a later stage.

7L COMPARISON WITH BUF (S4.11)

TABL(n), the starting value of n being 2, was compared with BUF(m), the value of m being 0 in the first cycle. If equality was not found, the program branched to Step S4.19. If not, it proceeded to Step S4.12.

7M UPDATING THE VALUE OF m, n AND C (S4.12)

The values of m and n were each increased by 1 to deal with the next seven words — that is, 28 characters — one by one. The value of C was increased by 1, to indicate that one word had been compared.

7N CHECKING FOR END OF COMPARISON (S4.13)

Checked whether the value of C was equal to 8. If it was not, the program was reentered at Step S4.11. If it was equal to 8, the program proceeded to Step S4.14.

*Annotation.*— The number of words to be compared was 8. The value of m and C were each increased by 1 after each word was compared. After each comparison the value of m would be either 8 or something else in later iterations for other fields.

7P STORING FOUR ASTERISKS (S4.14)

Four asterisks were stored in TABL (m + 1).

*Annotation.*— Entry into this Step indicated a successful matching of a Kernel Term with an Isolate Term. This filled in the last word position of TABL four asterisks to indicate that the Kernel Term would require no further comparison in the later cycles.

7Q CHANGING VALUES OF ADDRESS MODIFIERS (S4.15)

The values of N and n were made equal to 0 and 2 respectively.

7R CHECKING UP OF SPACE (S4.16)

The character (N) of BUF(m) was checked for space. If it was not space, the program proceed to Step S4.17. If it was space, the program was reentered at Step S4.3.

7S STORING OF ISOLATE NUMBER (S4.17)

The digit — that is, first digit of the Isolate Number in BUF + 2 — was stored in CN(P) (p).

7T MODIFYING THE VALUE OF P AND N (S4.18)

The values of P and N were each increased by 1 to deal with the next digit in BUF and CN. The program was then reentered at Step S4.16.