(1) Identification:

(a) a unique serial number;

(b) a "subject heading" (six alphabetic characters) to indicate the subject which the tree embodies;

(c) optionally, from one to three "output concept numbers" to provide for nesting of phrases (as described below).

(2) Structural information consisting of two logical (binary) incidence matrices to indicate which syntactic elements depend on which others. One of these matrices indicates the direct connections between elements (hereafter called "nodes"), the other shows indirect (removed) connections.

(3) Syntactic information: the syntactic function of each of the nodes (elements) of a phrase.

(4) Semantic information: the semantic values of the various nodes.

(5) Internal control information, e.g., a count of the number of nodes in the phrase.

The serial number serves only as an identifier whereby the editing routine can determine which trees must be deleted. The "output concept number" does <u>not</u> appear in the output of the criterion routine. Rather, it is used internally by the routine to indicate when one tree appears in a sentence as a part of another (as for example, "information retrieval" is a part of "automatic information retrieval").

Each phrase is considered to have one or more "key" nodes in which the "meaning" of the phrase is concentrated. This key node is usually the top node of the phrase: the noun in a noun phrase, the verb in a clause, etc. When a tree is found to match a given part of a sentence, the output concept numbers for that tree are recorded as semantic values pertaining to the sentence nodes which mate to individual tree nodes. Thus, if the sentence were "Automatic information retrieval is useful.", and 301 were an output concept number attached to a tree in which the term "information" modifies the term "retrieval," then the semantic value (concept number) 301 would be attached to the term "retrieval."

Because of the relative complexity of this specification, and because the criterion trees are stored as binary records on magnetic tape (see Fig. 3), an updating routine is required to create and maintain the criterion tree dictionary. Two routines, discussed in Sec. VIII of Information Storage and Retrieval, Report No. ISR-7, have been programmed to perform this task. One, called UPCRIT, edits an existing file of trees; the other, called MAKTRE, creates new tree records from alphabetic input.

Unfortunately, the original MAKTRE routine proved to be inadequate. In the first place, the routine requires a rather involved procedure for describing the structure of trees, even though in fact only about 14 different types of trees have actually occurred in practice. Secondly, all semantic information must be repeated anew for each tree, whereas practical experience reveals that the tree dictionary consists largely of groups of

generator which is equal to zero; this is illegal, as is a generator (or concept number) greater than 3072. A left parenthesis immediately followed by a right parenthesis is a valid format; but it defines a relation consisting of a single generator of value zero, and is therefore illegal as mentioned above. This error is defined as type V and gives rise to a diagnostic on tape A3.

D. Specifications Field

This field extends from the column following the dollar sign up to (but not including) the first space (blank). It may not be empty; this is illegal (error type S). It consists of a series of subfields, called tree specifications, separated by commas. Each tree specification causes exactly one tree to be written.

Each tree specification contains a tree type number. This is an unsigned decimal integer, greater than zero but not greater than the number of tree types available (this number is currently 13). A tree type number that falls outside these limits constitutes a "specification" (S-type) error. Appendix 1 gives the tree types currently available.

Following the tree type number may appear a specification for the serial (identification) number for the tree. If it is absent, the tree will be written with a serial number of zero. If present, the serial number part may take one of three forms:

(1) a slash, followed by an unsigned decimal integer;

(2) the single character "plus"; or

(3) the single character "asterisk."

These identification number specifications control a counter called "S" (for "Serial number"). The first type causes the counter S to be reset and replaced by the integer following the slash. The next type (character "plus") causes 1 to be added to the contents of S; the character "asterisk" causes no change in S. In any of these three cases, the final contents of S become the identification number of the tree.

3. Continuation Cards

It may happen that there is not enough space on a card image to record all the information desired. A means is provided for indefinitely extending any card image. This consists of the use of "continuation cards" which are treated as extensions of preceding cards. Continuation cards are identified by five blanks in columns 1-5, and an asterisk in column 6.

Continuation cards may legally follow only certain cards: those with no blanks in columns 7-72 and those in which a minus sign (11 punch) precedes all spaces (in columns 7-72). The reason for these restrictions is that a blank is considered to terminate the information on a card, whereas a minus sign indicates that a continuation is expected. If a continuation card appears under other circumstances, the error is type C (for "Continuation"); if one occurs at the start of the data, the error is

type B (for "Beginning"). If a card containing a simus sign is not followed by a continuation card, the error is type Q (for "seQuence").

4. Internal Processing

After reading each card image from tape, TRECND checks columns 1-6 to see whether it is a continuation card. If it is a continuation card, and the preceding card either had data to column 72 or contained a minus sign, then the data in columns 7-72 of this card are attached to the data from the preceding card. If it is not a continuation card, columns 1-6 are further checked to ensure that they satisfy the requirements of Part 2.

The card image is now scanned character by character, starting in column 7. This is done by means of a routine called GETFØR (for "GET character FØRward"), which not only retrieves the character from memory, but also identifies it by type: digit, letter, etc. Each special character is a distinct type. This information is provided by setting a specific bit in the sense-indicator register; one bit position corresponds to each type.

A different program section corresponds to each field and subfield mentioned in Part 2. Transfers of control occur among these various sections according to the types of character found by the routine. If any character appears where it is not expected, a format (F-type) error occurs. Every time a character is fetched, a check is made to see if it is blank or minus. If it is, the next card image is read. If no continuation is implied, processing resumes on the old card, using the blank that was fetched (a minus always implies a continuation).

The routine maintains a list of the relation generators which are given by the input. When the dollar sign is detected, these are sorted into numerical order by value (concept number). Then the specifications field is scanned. For each tree type, a table in memory already contains the direct and indirect connection matrices, the syntactic relation generators (already sorted), and a vector giving the node(s) to which the output concept number is to be taken as pertaining. This information is combined with the identification number (if any) from the same specification, and with the relations and output concept numbers already found. Then this data are written on tape. A count of the number of trees is updated, and the next tree specification is examined. When all the information on a card image (and its continuations, if any) is exhausted, the next card image is read. A card with a slash in column 1 terminates the process, and is the normal exit.

5. Error Conditions

A variety of checks is made to ensure that the input data are correct. Any violation of these checks will cause an output message on A3 (the system output tape). The remainder of the current card image and all following continuation cards will be skipped and processing will resume with the first noncontinuation card. If twenty errors of any type occur during one call of TRECND, it is assumed that the wrong deck is being processed. The result is that the input tape (A2) is backspaced one record, sense light 1 is turned ØN, and TRECND exits immediately. The same action occurs if an end

of file is detected in the input, or if a redundancy check persists for ten tries in either input or output. (The diagnostic messages themselves are not redundancy checked.) Just before exiting, whether due to error or from detection of a "STØP" sentinel, TRECND uses (STH) to write a message on A3, giving the count of trees written and of errors found. This occurs even if no errors occurred.

The errors detected by TRECND fall into one of several classes. First there are machine errors and tape malpositionings, which result in immediate termination (KILLx, ERRØRI); then card format (ERRØR-F), continuation sequence (-B,-C,-Q), concept number range (V), tree specification (S), and count limit (M) checks. The latter occur whenever there are more than 36 nodes, or 64 relation generators, or three output concept numbers, or more nodes specified than a tree type contains. Appendix 2 contains a list of these error types and the diagnostic messages which they produce, as well as the action taken after the diagnostic is written.

6. Use of a TRECND Program

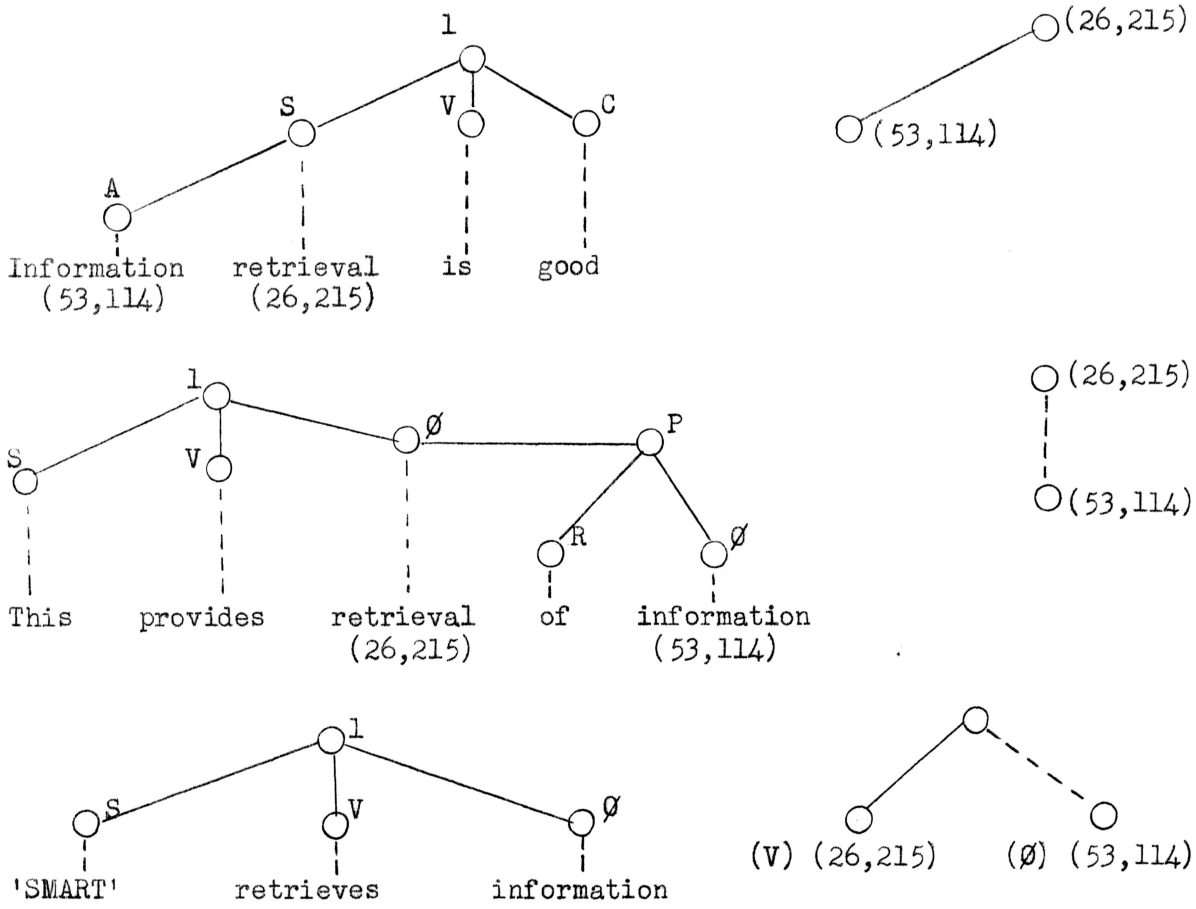
This part will describe the use and purpose of TRECND, together with some examples. A criterion tree such as is produced by TRECND is a prototype, or model, for a part of a sentence. It serves as a sample of a typical phrase having a definite meaning: for example, "information retrieval" where "information" modifies "retrieval." To construct a tree that will be useful in the SMART system, it is necessary to think of several sentences which use the phrase, then diagram them in the method used by the multiple-path English

syntactic analyzer, and finally to abstract the common features of these sentences (as shown in Fig. 1(a)).

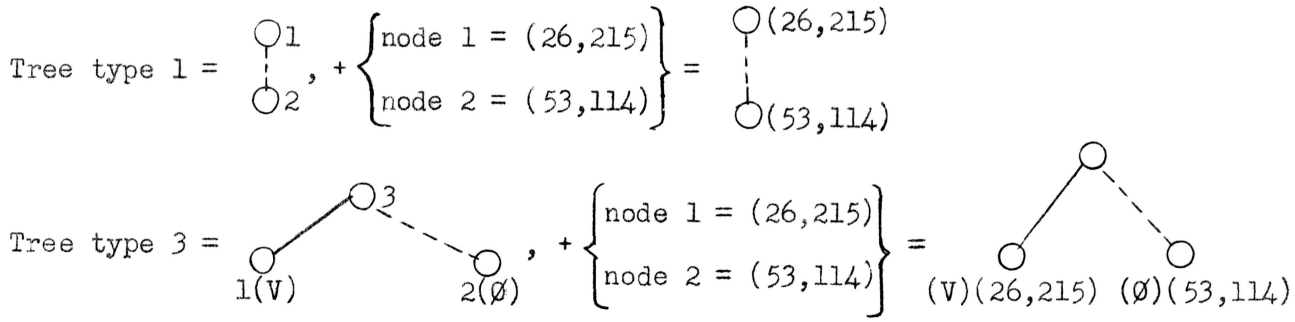
The second step is to compare these abstracted phrases with the list of standard types given in Appendix 1. In the example, we see that the first phrase is of type 7, the second of type 1, and the third of type 3. (Since replacing a direct dependence by an indirect dependence merely generalizes the tree, we may consider the first sentence as belonging to type 1 as well as type 7.) It is seen that in all three cases, concept numbers 53 and 114 ("information") pertain to node number 2, and concept numbers 26 and 215 ("retrieval") pertain to node 1. (The node numbering of the various tree types was selected to promote this kind of favorable occurrence.) This is shown schematically in Fig. 1(b).

It is now necessary to assign a BCD subject heading (e.g., "INFRET"), serial numbers for the two trees (e.g., 6,7), and an output concept number (301). A card is then constructed as shown in Fig. 2.

Figure 3 shows the trees that result from this card, as they appear on tape (tape unit B5). There is an extra field, called "Ø" for "Øutput concept number," which is not mentioned in previous reports. This new format is fully compatible with the old binary tape format; the Ø field is ignored by UPCRIT and by previous versions of the criterion routine CRITER. This field consists of one binary word, whose first six bits form a vector identifying the "key" node(s) of the tree. The remaining 30 bits are divided into three 10-bit subfields, each containing one output concept number.



(a) Sample sentences referring to "information retrieval," and abstracted phrases



(b) Relation of Fig. 1(a) to tree types 1 and 3

Construction of Criterion Trees for "Information Retrieval"

Figure 1

Step 1. The BCD index occupies columns 1-6:

INFRET
1....6

Step 2. The output concept number follows:

INFRET = 301
7..10

Step 3. The concept numbers for node 1:

INFRET = 301 (26,215)
11.....18

Step 4. A slash separates the numbers for node 1 from those for node 2:

INFRET = 301 (26,215) / (53,114)
19.....27

Step 5. A dollar sign introduces the specifications field:

INFRET = 301 (26,215) / (53,114) $ 1 / 6
28..31

(which specifies a tree of type 1, serial number 6)

Step 6. A comma separates the first specification from the next:

INFRET = 301 (26,215) / (53,114) $ 1 / 6 , 3 +
32.34

(specifying a tree of type 3, serial number 7(= 6 + 1))

Step 7. A blank terminates the card:

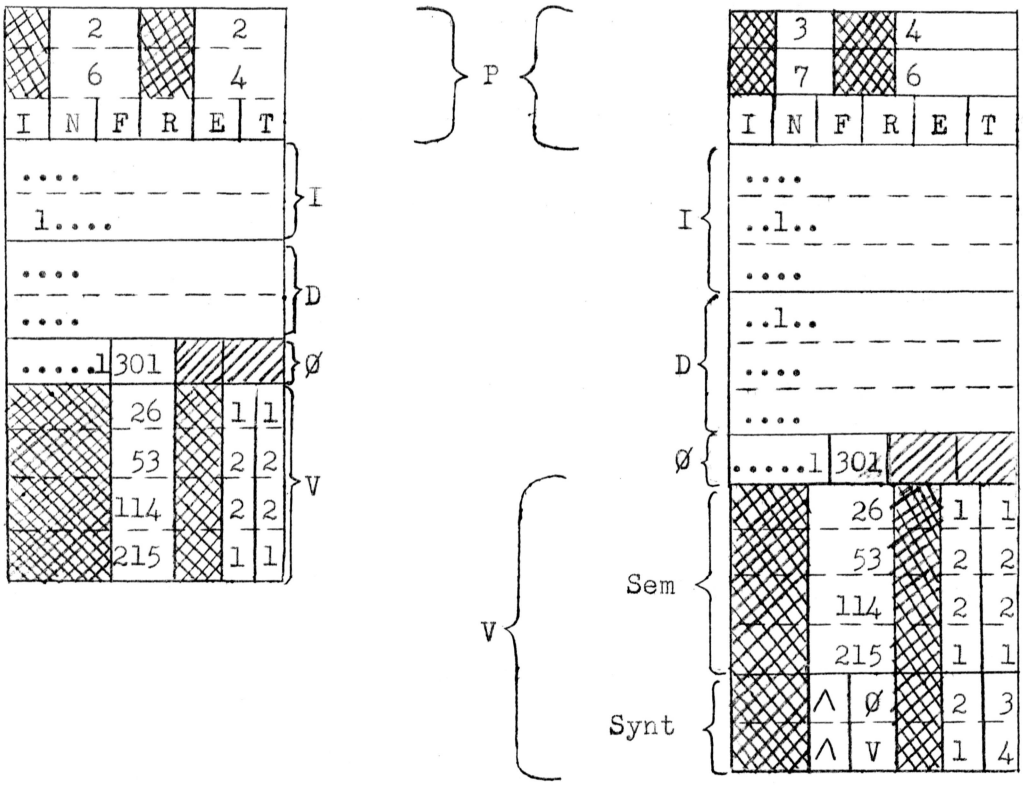
INFRET = 301 (26,215) / (53,114) $ 1 / 6 , 3 + Δ
35

Construction of a Card to Make Criterion Trees

Figure 2

```
Column Number 1...5....10....15....20....25....30....35
              INFRET= 301( 26,215)/ (53,114)$ 1/ 6, 3+
```

(a) Card input to TRECND



(b) Binary trees generated from Fig. 3(a)

```
1...5....10....15....20                                    72...76
INFRET                                                        6
       1  X (26,215)                                         6
       2  1I (53,114)                                        6
INFRET                                                        7
       1  3D (26,215)                                        7
       2  3I (53,114)                                        7
       3  X                                                  7
```

(c) Cards required by MAKTRE for same result

Binary Tree Resulting from Card of Fig. 2, and a Sample of Input Required by MAKTRE

Figure 3

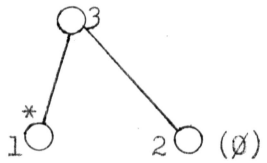
Figure 3(c) shows the seven cards required by MAKTRE to generate the same two trees. The two routines produce equivalent output, except for the fact that MAKTRE does not produce an Ø field. In this respect, therefore, TRECND is more general than MAKTRE.

APPENDIX 1

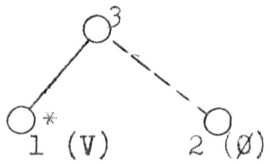
STANDARD TREE TYPES

I

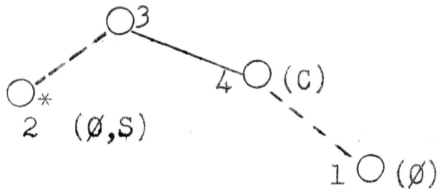
II



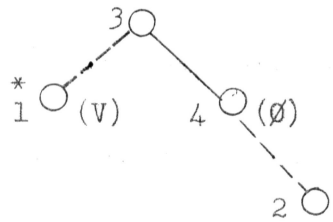
III



IV



V

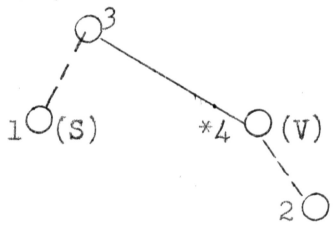


VI

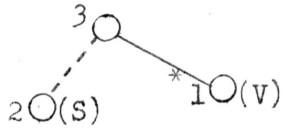
VII

VIII

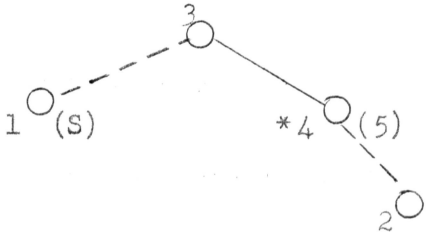
IX



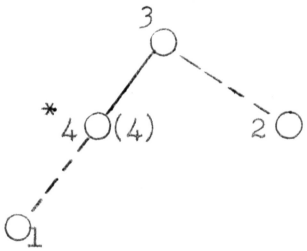
X



XI



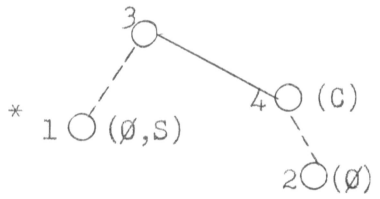
XII



Note: The asterisk shows "key" nodes, i.e., the nodes to whose correspondents the output concept number will be attached after a match is found.

APPENDIX 1 (continued)

XIII



APPENDIX 2

ERROR TYPES

Whenever one of the errors in the following list is encountered, the indicated message is written on tape A3. This message is preceded by "$\wedge$TRECND$_{\wedge\wedge}$-$_{\wedge\wedge}$" and followed by a copy of the card image to which the message applies. The equals sign immediately precedes the first column of the card image:

Type	Action	Message
ERRORB	R	DATA MAY NOT BEGIN WITH CONTINUATION CARD =
ERRORX	R	BCD INDEX IS IN ERROR, =
ERRORQ	Q	SEQUENCE ERROR. NO CONTINUATION CARD FOLLOWS =
ERRORF	R	FORMAT OF THIS CARD IS IN ERROR. =
ERRORM	R	SOME SIZE OR COUNT LIMIT IS EXCEEDED IN =
ERRORC	R	A CONTINUATION CARD IS NOT REQUIRED HERE =
ERRORI	S	THIS ERROR IS THEORETICALLY IMPOSSIBLE. RUN BAD =
ERRORS	R	ERROR IN SPECIFICATION OF TREE TYPE. CARD =
ERRORV	R	INCORRECT SEMANTIC VALUE (CONCEPT NO) IN =
KILLR	S	EXCESSIVE REDUNDANCY ON READIN OF =
KILLW	S	EXCESSIVE REDUNDANCY ON WRITE =
KILLF	S	END OF FILE ON INPUT, =
KILL	S	IMPOSSIBLE EØF OR REDUN AFTER BACKSPACE =
TAPOV	S	END OF TAPE DETECTED ON LIBRARY TAPE, CARD =

Following each type in the above table is a BCD symbol, either "Ø," "R," or "S." These indicate the action taken after the diagnostic is printed. "Ø" means "process the next card image"; "R" means "skip continuation cards, process next noncontinuation card"; and "S" causes an exit with sense light 1 turned "ØN."