AN APPLICATION OF COMPUTER PROGRAMMING TO THE RECONSTRUCTION OF A PROTO-LANGUAGE

Stanton P. Durham and David Ellis Rogers
University of Michigan

1.Purpose. This paper illustrates the use of a computer program as a tool in linguistic research. The program under consideration produces a concordance on words according to phonological segments and environments. Phonological segments are defined as a predetermined set of consonants and vowels. An environment is defined as the locus of occurrence of any of the phonological segments. The concordance facilitates the recognition of sound correspondances that lead to the reconstruction of a proto-language.

2.0.Program Description. The program for production of the concordance was written in the SNOBOL4 programming language, which was selected because of its pattern matching capabilities.[1] The summary Flow Chart of the program, found in §7, should be adequate for the experienced reader. Nevertheless, a few general comments are in order.

2.1.Initialization. All patterns to be used in the program are created during the Initialization. As originally conceived, the program was composed of one long run where

---

[1]For a full exposition of SNOBOL4, see Griswold, R.E., Page, J.F., and Polonsky, I.P., The SNOBOL4 Programming Language. Holmdel, New Jersey: Bell. 1968.

all steps of the analysis were carried out. However, due to problems of internal storage caused by the numerous data, it was decided to run the program in two passes, each of which is explained below.

2.2.Pass One. During the first pass of the program all cards of an item are read. An item is defined as the Classical Latin dictionary form, followed by the author's phonemicization of the Latin form according to the most conservative estimate of the maximal phonological system capable of giving rise to the various dialects of spoken Latin. These two elements are followed optionally by the modern reflexes in as many as eleven dialects from the area commonly subsumed under the heading Franco-Provencal. An optional comment concludes the item.

As the items are read, determination is made of the largest size of each element for later column alignment in the print-out. Each item is then stored as a string named after the sequential number assigned to the item, and the phonological form on which the concordance will be based is selected. The phonological form is then analyzed in order to retain the generic types and specific segment-environments occurring in that phonological form. A generic type is defined as a consonant or vowel in a given environment, as for example, word-initial consonant

or tonic free vowel. A specific segment-environment is defined as one certain consonant or vowel in a given environment, as for example, word-initial P or tonic free A. For each specific segment-environment found, a list is created composed of the numbers of the items containing that specific segment-environment.

As all items are read and analyzed, errors in phonological form are outputted. After the analysis of all input items is completed, a generic type list is examined for a specific segment-environment. From the list named after that specific segment-environment the numbers of the items containing the specific segment-environment are obtained and the items are written onto tape in condensed form for accessing in Pass Two.

2.3.Pass Two. During the second pass, the condensed listings are accessed from the tape, along with the largest size of each element, and alignment of columns is made just prior to printing for easier reading of the print-out.

3.0.Specification. The program is designed to list all words in a dialect (for example, Latin or a present-day dialect of Latin) containing a specific segment in a given environment. The order for the production of the lists is outlined in the following paragraphs.

3.1.Single Consonants. All words containing single conson-

ants are listed according to two parameters: a predetermined order of those consonants and, within that parameter, according to the following environments: word-initial, geminate, syllable-initial, syllable-final, and word-final position. A geminate consonant is defined as a long consonant, sometimes described as double, occurring across syllable boundaries.

3.2. Clusters. A cluster is defined as the occurrence of two or more consonants in immediate succession in the same syllable. All words containing clusters of two consonants are listed according to the parameters of §3.1, but in reverse order. This order of consonants is the same as the order used to list single consonants with the additional stipulation that the value of the consonant in left-most position have precedence over the one in right-most position, as in any dictionary order.

Lists of words containing clusters of three or more consonants follow, according to the same parameters as those specified for two-consonant clusters. Where there are different numbers of consonants (three or more) in the clusters, the words are listed giving the highest value to the consonant in the n^{th} position, followed by the consonant in the $n + 1^{th}$ position (counting consonants from left to right), and according to the predetermined order of con-

sonants. For example, given the predetermined order KPWFS TDMNRBLJGQXZ, listings of words having the following clusters in word-initial position would appear in this order:

all	words	containing	word-initial	KLJ
"	"	"	"	KJ
"	"	"	"	STR
"	"	"	"	STRJ
"	"	"	"	STJ
"	"	"	"	MJ.

3.3.Sequences. A sequence is defined as the occurrence of two or more consonants in immediate succession across syllable boundaries, the syllable boundary always being indicated by a period. Words containing strong sequences, composed of a geminate consonant plus at least one additional consonant, are listed first, and the sequence is abbreviated $C_1.C_1C^n$. The order of the listing is the same as that specified in §3.1, except that there is only one environment, "strong." For example, given the predetermined order KPWFSTDMNRBLJGQXZ, listings of words having the following strong sequences would appear in this order: K.KL, K.KJ, P.PJ, T.TJ, N.NTR, N.NTRJ.

Following the words containing strong sequences, all other sequences are listed. The first words listed are those with two-consonant sequences; that is, sequences with only one consonant on either side of the syllable boundary, abbreviated $C_1.C_2$. Then all words containing sequences of a single consonant followed by a syllable boundary,

followed by more than one consonant, abbreviated $C_1.C_2C^n$, are listed; followed by words containing all other sequences, abbreviated $C^nC.C(C^n)$, where the parentheses indicate optionality. In each of these listings the consonant or syllable boundary in the n^{th} position has higher value than the consonant in the $n + 1^{th}$ position (consonants being counted from left to right). For example, if the following sequences were found, they would be listed in the following order: M.P, R.L, N.STJ, R.LJ, NT.T, NT.TJ.

3.5.Vowels. Words containing vowels are listed next, according to the following ordered parameters and subparameters: stress (tonic, pre-tonic, post-tonic), length (long, non-long), position (in free or checked syllable), and the predetermined order of vowels. For example, given the predetermined order IEAOU, the listings would occur in the following order:

all words containing long tonic free I
" " " " " " E
" " " " " " A
" " " " " " O
" " " " " " U,

and so on, through the long tonic checked vowels, the non-long tonic free and checked vowels, the long pre-tonic free and checked vowels, the non-long pre-tonic free and checked vowels, etc., until all possible combinations of parameters have been listed.

3.6.Special Environments. Listed lastly are occurrences

of the so-called velar consonants /k g kw gw/ (symbolized in the program as K,G,Q,X) followed by a front vowel or /j/. These lists are called "special" and are printed as a separate portion of the print-out, because of the well-known phenomen of the palatalization of these Latin consonants plus a front vowel or /j/.

3.7.Errors. Toward the end of the first pass, before the condensed listings are outputted onto the tape, certain errors are printed out. Errors may be errors in phonological form, as for example, use in the phonological form of a consonant symbol that has not been pre-defined, failure to punch a syllable boundary, or failure to punch length or stress symbols; or the error may be the lack of occurrence of the phonological form for which the program is searching.

3.8.Alignment. The second pass is almost entirely composed of the subroutine in which the elements of an item are aligned in columns in the listings on the basis of the number of characters in the longest occurrence of that particular element.

4.0.Instructions to User. A system of symbolization for vocalic and consonantal specific segments must be decided on. During the processing there must be only one computer character for each segment the computer will examine. If

it is necessary (because of the non-availability of many customary linguistic symbols as characters in the computer alphabet) to encode the data with two symbols for one phonological segment, the program should have all the double symbols used and the corrseponding single computer symbols by which the program will process the data defined. Because of peculiarities in the program it is also necessary to change any numbers, V, or C, that may be used as consonantal or vocalic symbols in the phonological form to be concordanced on to some other unique computer symbol. For example, if theta and delta are encoded as TH and DH, and if the symbols C, V, and 5 are used in the input program and in the representation of a specific phonological segment, then the following two statements should be inserted at the appropriate place in the program:

```
EXT3 = 'TH DH C V 5 '
INT3 = 'a b c d e '
```

where a, b, c, d, and e are unique symbols belonging to the character set of the particular computer, and different from other symbols punched in the phonological form to be concordanced on. In the present program double symbols are freely used in the transcription of the dialect reflexes. If a concordance is to be produced on the basis of one of the dialects, the above modifications must still apply.

4.1. Restrictions. The present program is designed to con-

cordance on the second element of an item, the phonological representation of the spoken Latin form. To produce a concordance on a dialect, the phonological form to be concordanced on must be redefined.

A special environment may be searched for and listed separately by means of the insertation of a statement defining an appropriate pattern in the Initialization of the program, and by the placement of a search for that pattern in the body of the program. If one is producing a concordance on a particular dialect, then special environments may be defined according to symbols used in that particular dialect.

4.2. Encoding of the Data. All cards will have information beginning in column one and may have information punched continuously through column seventy-five. Columns seventy-six through eighty may be uniquely sequentially numbered for each entry (column seventy-nine having units position and column eighty being saved for insertions). A linguistic unit may be split between cards; in such cases no hyphenation will be needed. That is, in all instances the information beginning in the first column of the second and subsequent cards of an entry will be abutted to the seventy-fifth column of the previous card.

The first card of an item will begin in column one with the dictionary entry of the Latin word, with both

vowel length and stress indicated, followed by two spaces. Indication of stress is redundant, stress being predictable in Classical Latin. However, stress is indicated in dictionary fashion, as an aid toward rapid recognition of the proper stress by the reader. Though the accusative singular of Classical Latin nouns is the citation form, with few exceptions, for the first element, the final m has in all instances been omitted. Thus, where the noun nox is cited, it is spelled NO-CTE, rather than NO-CTEM, to save space, and because texts which cite spoken Latin nouns usually cite such nouns without final m. The asterisk is used to indicate an unattested Classical Latin form, in most instances taken from Wilhelm Meyer-Lübke's Romanishces Etymologisches Wörtebuch, but in a few instances posited by the author. In all cases where words of Germanic or Celtic origin have been latinized in spelling, they are also preceded by an asterisk. In Latin dictionary forms of more than one word, the words are separated by a plus, which is removed at the end of the program.

The second linguistic information, the phonemicization of the spoken Latin word, is followed by at least one space. The dialect entries follow, each composed of, first, the identifying abbreviation enclosed in parentheses and second, the reflex in that dialect, preceded by one space and followed by at least one space. At least one

space is obligatory after each dialect entry, but more spaces facilitate correction of an erroneously punched form. An optional comment concludes the item; the abbreviation for the comment, (COM), must precede the comment and be followed by one space. When dialect identifying abbreviations are used in the comment, they must not be enclosed in parentheses, lest the computer mistake one of these abbreviations for the actual identifier. An end-of-item slash completes the item, and a single space is obligatory before the slash.

5. Example. The examination of one item will suffice to illustrate the working of the program. Let us suppose the item currently under consideration by the computer is the Latin word alteru. The data cards containing this word and its reflexes will have the following information:

A-LTERU A-L.TRU (B) <:-.TRE (V) ?:-.TRO (O)
A:-.TRO (A) O-TR (R) O-.TR@ (S) >:-TR (P)
O:-.TRU (N) A:-.TR> (COM) S IS PLU AND FINAL
VOW OF R,P,N ALL SEMI-PRONOUNCED. B,V HAVE FEM
<:-.TRA, >:-.TRA. / 000001

where < stands for /ɑ/; ?, /ʌ/; @, /ə/; and >, /ɔ/.

After the entire item has been read into computer memory, and determination has been made as to the size of each entry relative to the individual entries of all other

items, a search is made for the so-called "special" environments, at C1 in the Flow Chart. None of these environments are applicable in the case of alteru. Therefore, these searches will fail, and the next search will be for a word-initial consonant or consonants, at C2 in the Flow Chart. In the case of alteru this search, too, will fail, and the next search will be for a vowel, at A8 in the Flow Chart. A tonic vowel in a checked syllable will be found at A8.2 and A8.6, and in the subroutine B, tonic checked A will be queued to the string containing all tonic checked vowels, and the item number will be queued to a string containing the numbers of all items having a tonic checked A.

The next search will be for a consonant or consonants in all possible environments, beginning at A10 in the Flow Chart. Searches for a strong sequence or a geminate consonant will fail. At A12 the search for a sequence will be successful, the sequence found being L.TR. Once more, subroutine B is entered, the sequence L.TR is queued to the string labeled "sequence C.CC" at B1.1, if this is the first occurrence of L.TR, and the item number is queued to the string containing the item numbers of all items having the sequence L.TR at B1.2. Next, at A13, the syllable-final L, and at A14, the syllable-initial cluster TR, will be queued respectively to the strings containing syllable-final consonants and syllable-initial clusters, and the

item number will be queued to the strings containing the numbers of all items having syllable-final L's in the one case, and to the string containing the item numbers of all items having syllable-initial TR in the other.

The subsequent search for a post-tonic vowel will succeed at A8.3, and the vowel U in free syllable (in fact in word-final position) will, in subroutine B, be queued to the string of post-tonic vowels in free syllables, the item number being queued to the string containing the item numbers of all items with post-tonic free U. At this point, return is made out of subroutine B to the beginning of the program for the reading of the next item.

After all items have been read and operated on, the strings and their headings stored in computer memory are outputted in condensed form onto magnetic tape. The item alteru will be found under the following headings: syllable-final L, syllable-initial TR, sequence L.TR, tonic checked A, and post-tonic free U.

In Pass Two, the tape will be read, and the listings will be printed with the elements of each item aligned in columns.

6.0. Linguistic Conclusions. During the course of reconstruction, one interesting question that arose was the following: do the so-called Franco-Provencal dialects

really show final (post-tonic) vowels, as for example, in the above-mentioned Latin etymon, alteru? With all items containing reflexes of Latin post-tonic free U in one convenient list, checking the possible correspondances is made much easier. Alteru, for example, shows the following correspondances:

dialect	B	E
"	O	O
"	P	U
"	N	>
"	H	(unavailable)
"	V	O
"	D	(unavailable)
"	C	"
"	A	zero
"	R	@
"	S	zero,

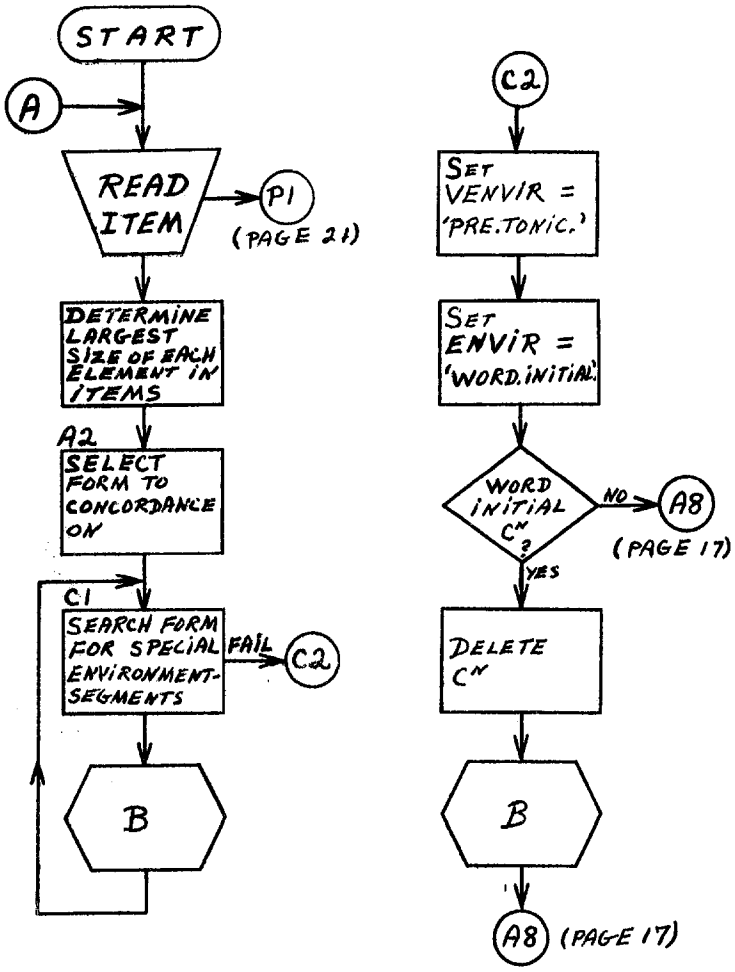
as do all examples of U after a consonantal sequence. However, for most other examples of Classical Latin post-tonic free U, all dialects show zero. On the basis of all examples under the heading "post-tonic free U" one may conclude that there is a reflex of Latin post-tonic U in these dialects as a support vowel after an otherwise unpronounceable sequence. Furthermore, this support vowel keeps the quality of its phonological ancestor.

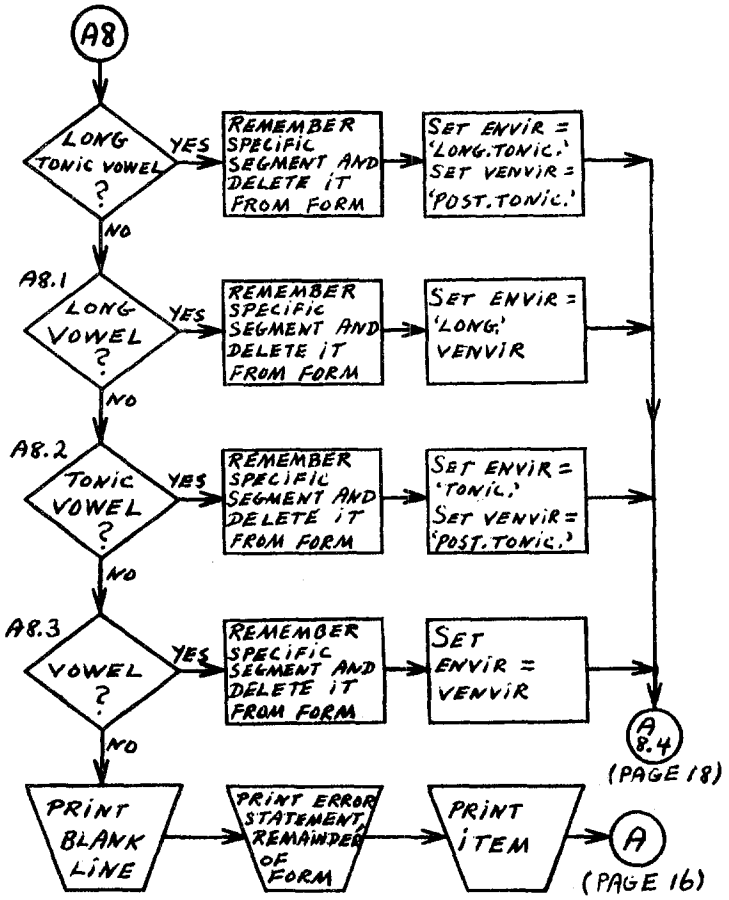
Such questions as this are capable of much more rapid, if not surer, solution, by consultation of the listings on the computer print-out, than simply by means of the examination of index cards, where examples might be skipped over. The number of examples available for examination is

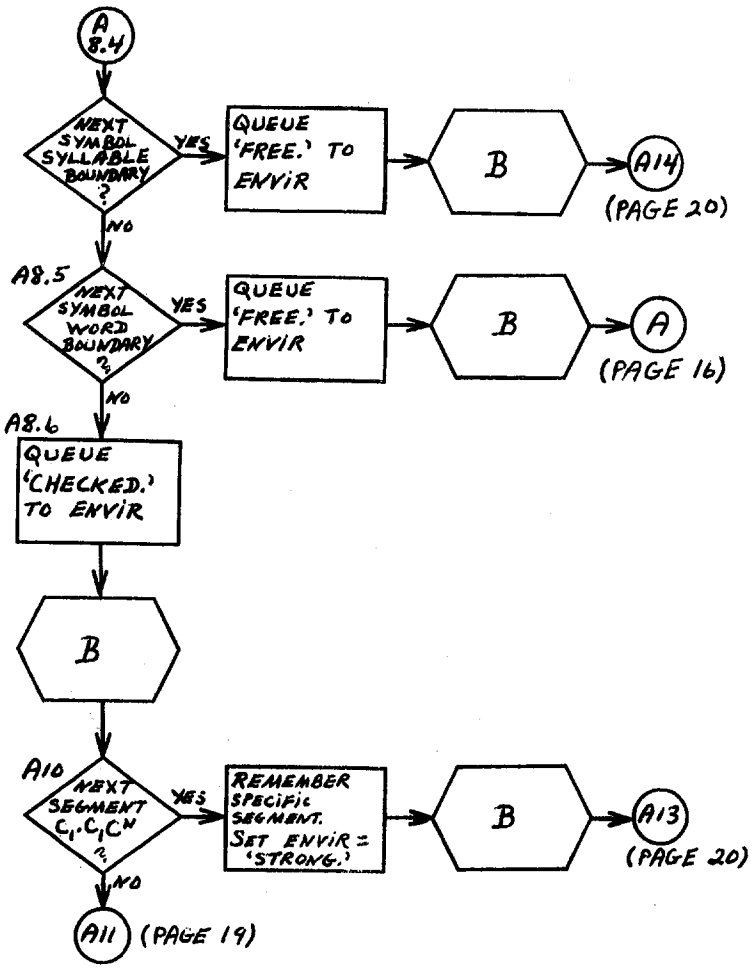
greatly increased as well. Since the data are so numerous with this method, very comprehensive analysis is required of the linguist.

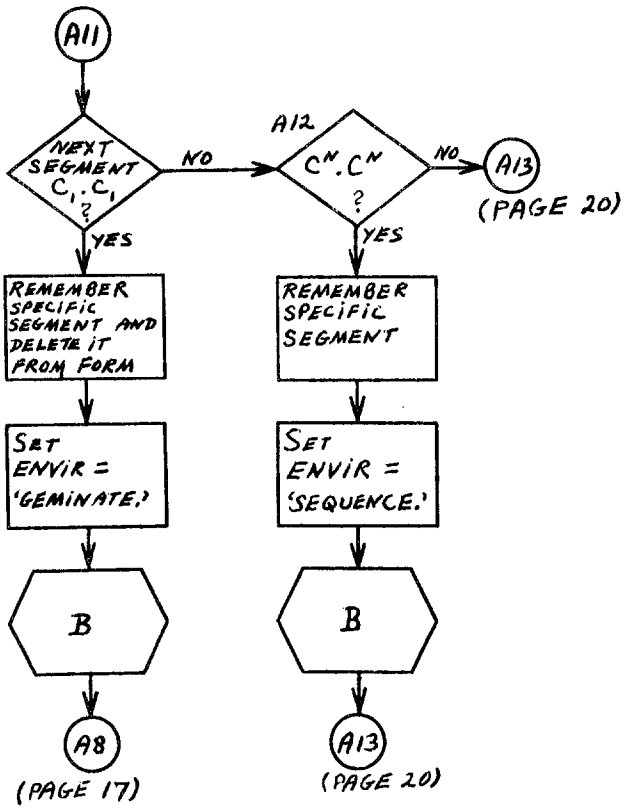
This program is general enough to be applicable in the compilation of a concordance for any group of related dialects for which such a concordance would be useful. For example, in a proposed reconstruction of Proto-Slavic, present-day reflexes of a selected corpus could be encoded and the concordance produced on any one of the dialects selected.

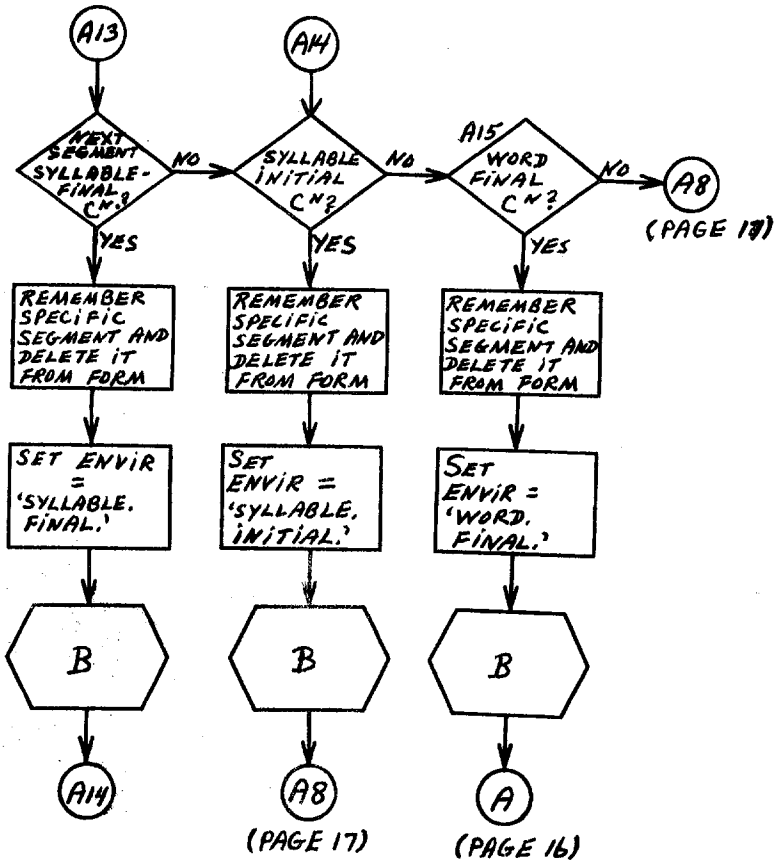
The chief advantages of the use of the computer to produce such a concordance are the increased facility for the exhaustive handling of a large amount of data (as compared to the customary handling of data on index cards), and the avoidance of many time-consuming searches through many lists of forms for occurrences of a specific segment in a specific environment, since all such lists are readily available on the print-out.

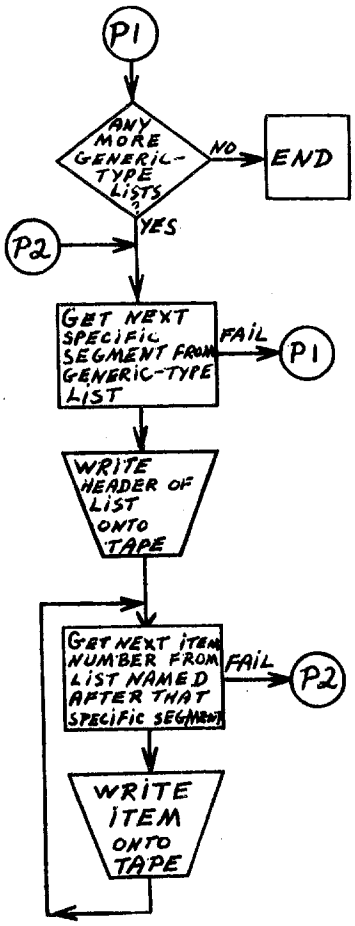

START
A
READ ITEM
P1
(PAGE 21)
DETERMINE LARGEST SIZE OF EACH ELEMENT IN ITEMS
A2
SELECT FORM TO CONCORDANCE ON
C1
SEARCH FORM FOR SPECIAL ENVIRONMENT-SEGMENTS
FAIL
C2
B
C2
SET VENVIR = 'PRE.TONIC.'
SET ENVIR = 'WORD.INITIAL.'
WORD INITIAL C^N ?
NO
A8
(PAGE 17)
YES
DELETE C^N
B
A8
(PAGE 17)


A8
LONG TONIC VOWEL ?
YES
REMEMBER SPECIFIC SEGMENT AND DELETE IT FROM FORM
SET ENVIR = 'LONG.TONIC.' SET VENVIR = 'POST.TONIC.'
NO
A8.1
LONG VOWEL ?
YES
REMEMBER SPECIFIC SEGMENT AND DELETE IT FROM FORM
SET ENVIR = 'LONG.' VENVIR
NO
A8.2
TONIC VOWEL ?
YES
REMEMBER SPECIFIC SEGMENT AND DELETE IT FROM FORM
SET ENVIR = 'TONIC.' SET VENVIR = 'POST.TONIC.'
NO
A8.3
VOWEL ?
YES
REMEMBER SPECIFIC SEGMENT AND DELETE IT FROM FORM
SET ENVIR = VENVIR
NO
A 8.4
(PAGE 18)
PRINT BLANK LINE
PRINT ERROR STATEMENT, REMAINDER OF FORM
PRINT ITEM
A
(PAGE 16)

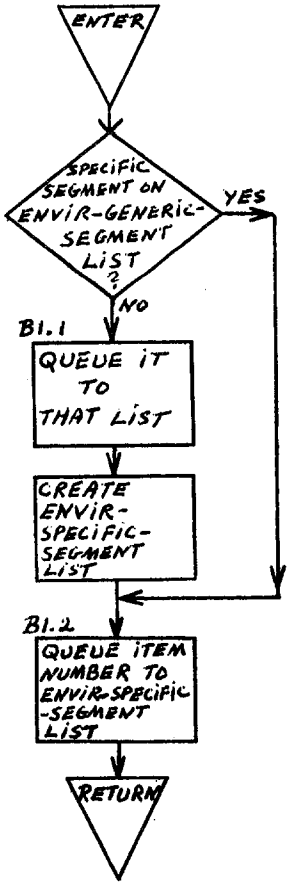

A 8.4
NEXT SYMBOL SYLLABLE BOUNDARY ?
YES
QUEUE 'FREE.' TO ENVIR
B
A14
(PAGE 20)
NO
A8.5
NEXT SYMBOL WORD BOUNDARY ?
YES
QUEUE 'FREE.' TO ENVIR
B
A
(PAGE 16)
NO
A8.6
QUEUE 'CHECKED.' TO ENVIR
B
A10
NEXT SEGMENT $C_1.C_1C^N$?
YES
REMEMBER SPECIFIC SEGMENT. SET ENVIR = 'STRONG.'
B
A13
(PAGE 20)
NO
A11
(PAGE 19)


A11
NEXT SEGMENT $C_1.C_1$?
NO
A12
$C^N.C^N$?
NO
A13
(PAGE 20)
YES
YES
REMEMBER SPECIFIC SEGMENT AND DELETE IT FROM FORM
REMEMBER SPECIFIC SEGMENT
SET ENVIR = 'GEMINATE.'
SET ENVIR = 'SEQUENCE.'
B
B
A8
(PAGE 17)
A13
(PAGE 20)


A13
A14
A15
NEXT SEGMENT SYLLABLE-FINAL C^N?
NO
SYLLABLE INITIAL C^N?
NO
WORD FINAL C^N?
NO
A8
(PAGE 17)
YES
YES
YES
REMEMBER SPECIFIC SEGMENT AND DELETE IT FROM FORM
REMEMBER SPECIFIC SEGMENT AND DELETE IT FROM FORM
REMEMBER SPECIFIC SEGMENT AND DELETE IT FROM FORM
SET ENVIR = 'SYLLABLE. FINAL.'
SET ENVIR = 'SYLLABLE. INITIAL.'
SET ENVIR = 'WORD. FINAL.'
B
B
B
A14
A8
(PAGE 17)
A
(PAGE 16)


P1
ANY MORE GENERIC-TYPE LISTS ?
NO
END
YES
P2
GET NEXT SPECIFIC SEGMENT FROM GENERIC-TYPE LIST
FAIL
P1
WRITE HEADER OF LIST ONTO TAPE
GET NEXT ITEM NUMBER FROM LIST NAMED AFTER THAT SPECIFIC SEGMENT
FAIL
P2
WRITE ITEM ONTO TAPE

B Routine


ENTER
SPECIFIC SEGMENT ON ENVIR-GENERIC-SEGMENT LIST ?
YES
NO
B1.1
QUEUE IT TO THAT LIST
CREATE ENVIR-SPECIFIC-SEGMENT LIST
B1.2
QUEUE ITEM NUMBER TO ENVIR-SPECIFIC-SEGMENT LIST
RETURN