Kansas Working Papers in Linguistics  
Volume 8, number 1, 1983

CONTENTS

<table>
<thead>
<tr>
<th>Author</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maria Alzira Nobre</td>
<td>Oral Vowel Reduction in Brazilian Portuguese</td>
<td>1</td>
</tr>
<tr>
<td>Frances Ingemann</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patricia J. Hamel</td>
<td>Brazilian Portuguese Stressed Vowels: A Durational Study</td>
<td>31</td>
</tr>
<tr>
<td>Andrea G. Osburne</td>
<td>Comma Intonation in a Tone Language</td>
<td>47</td>
</tr>
<tr>
<td>Philip Hubbard</td>
<td>Albanian Reflexives: Violations of Proposed Universals</td>
<td>63</td>
</tr>
<tr>
<td>Mark J. Elson</td>
<td>The Generative Relationship</td>
<td>73</td>
</tr>
<tr>
<td>Hiroshi Nara</td>
<td>On Michael Dumett's Notion of Decidability</td>
<td>87</td>
</tr>
<tr>
<td>Kenneth L. Miner</td>
<td>Computerized Permutation of Pikean Field Metrics</td>
<td>97</td>
</tr>
<tr>
<td>Barbara Lynn Taghva</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anne M. Ediger</td>
<td>An Analysis of Sex Stereotypes in the Japanese Language</td>
<td>107</td>
</tr>
<tr>
<td>Contents of Previous Volumes</td>
<td></td>
<td>129</td>
</tr>
</tbody>
</table>
Present-day computer technology ought to encourage re-examination of pre-computer linguistic concepts, especially tools for analysis, since some earlier ideas, difficult to apply or work with when offered, may turn out to be useful if computerized. Here we give one example of such an investigation, that of Pike’s field matrix permutation.

Pike 1962 explored the concept of grammatical systems, or more properly sub-systems, as matrices, with categories (syntactic, morphological, phonological, semantic) as parameters and formatives as intersections of parameters, or cells. Pike 1963 put forward the idea of matrix permutation as a tool in linguistic analysis. This technique, like many, perhaps all, good ones, was of course always implicit in linguistic work, and Pike’s contribution was to show how it might be systematized. We will have nothing to say here about the implications of matrix permutation for Pike’s theory of language, or indeed for any theory of language, but will regard it merely as a tool for organizing data and revealing generalities.

In our discussion we will use, as data, Tano verbal prefixes as described in Treger 1956.1 One first sets up a matrix of the familiar sort, with rows and columns determined by an arbitrary arrangement of categories, as in Table 1.

Table 1. Initial Tano Matrix

<table>
<thead>
<tr>
<th>OBJ</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>1 sg</th>
<th>1 du</th>
<th>1 pl</th>
<th>2 sg</th>
<th>2 du</th>
<th>2 pl</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 sg</td>
<td>o</td>
<td>ku</td>
<td>i</td>
<td>may</td>
<td>may</td>
<td>may</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3 sg</td>
<td>ə</td>
<td>u</td>
<td>i</td>
<td>o</td>
<td>gn</td>
<td>i</td>
<td>ə</td>
<td>məŋ</td>
<td>məŋ</td>
</tr>
<tr>
<td>1 du</td>
<td>qa</td>
<td>kəŋ</td>
<td>qaŋ</td>
<td>qäŋ</td>
<td>X</td>
<td>X</td>
<td>ə</td>
<td>məŋ</td>
<td>məŋ</td>
</tr>
<tr>
<td>3 du</td>
<td>qa</td>
<td>qa</td>
<td>qaŋ</td>
<td>o</td>
<td>gn</td>
<td>i</td>
<td>ə</td>
<td>məŋ</td>
<td>məŋ</td>
</tr>
<tr>
<td>2 du</td>
<td>məŋ</td>
<td>məŋ</td>
<td>məŋ</td>
<td>may</td>
<td>may</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2 pl</td>
<td>məŋ</td>
<td>məŋ</td>
<td>məŋ</td>
<td>may</td>
<td>may</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1 pl</td>
<td>i</td>
<td>kəw</td>
<td>i</td>
<td>i</td>
<td>X</td>
<td>X</td>
<td>ə</td>
<td>məŋ</td>
<td>məŋ</td>
</tr>
<tr>
<td>3 pl</td>
<td>i</td>
<td>lw</td>
<td>i</td>
<td>i</td>
<td>o</td>
<td>gn</td>
<td>i</td>
<td>ə</td>
<td>məŋ</td>
</tr>
<tr>
<td>1 sg</td>
<td>ti</td>
<td>o</td>
<td>pi</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>ə</td>
<td>məŋ</td>
<td>məŋ</td>
</tr>
</tbody>
</table>

Kansas Working Papers in Linguistics, 1953, volume 8/1, Pages 97-106
Then successive permutations of rows and columns are made in order to bring similar formatives together. The goal is to obtain blocks (and other patterns of distribution; see below) so that generalisations can be made. For the data in the initial matrix of Table 1, a possible final matrix is given in Table 2.

<table>
<thead>
<tr>
<th>OBJ</th>
<th>2 ag</th>
<th>1 pl</th>
<th>1 sg</th>
<th>1 du</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>2 pl</th>
<th>2 du</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 sg</td>
<td>q</td>
<td>i</td>
<td>o</td>
<td>qn</td>
<td>q</td>
<td>u</td>
<td>i</td>
<td>ma</td>
<td>ma</td>
</tr>
<tr>
<td>3 pl</td>
<td>q</td>
<td>i</td>
<td>o</td>
<td>qn</td>
<td>1</td>
<td>iw</td>
<td>ipi</td>
<td>ma</td>
<td>ma</td>
</tr>
<tr>
<td>3 du</td>
<td>q</td>
<td>i</td>
<td>o</td>
<td>qn</td>
<td>qn</td>
<td>qn</td>
<td>qoppn</td>
<td>ma</td>
<td>ma</td>
</tr>
<tr>
<td>1 du</td>
<td>q</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>qn</td>
<td>kqn</td>
<td>qoppn</td>
<td>ma</td>
<td>ma</td>
</tr>
<tr>
<td>1 pl</td>
<td>q</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>kiw</td>
<td>ipi</td>
<td>ma</td>
<td>ma</td>
</tr>
<tr>
<td>1 sg</td>
<td>q</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>ti</td>
<td>o</td>
<td>pi</td>
<td>ma</td>
<td>ma</td>
</tr>
<tr>
<td>2 ag</td>
<td>X</td>
<td>may</td>
<td>may</td>
<td>may</td>
<td>o</td>
<td>ku</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2 pl</td>
<td>X</td>
<td>may</td>
<td>may</td>
<td>may</td>
<td>ma</td>
<td>mgw</td>
<td>mpi</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2 du</td>
<td>X</td>
<td>may</td>
<td>may</td>
<td>may</td>
<td>ma</td>
<td>ma</td>
<td>ma</td>
<td>ma</td>
<td>ma</td>
</tr>
</tbody>
</table>

It is revealing to line in the occurring patterns of distribution, as has been done here.

At least five generalisations can be tentatively made from Table 2:

i. q- marks 2 ag OBJ
ii. ma- marks involvement of 2nd person
iii. -n- marks dual number
iv. -pi-, with alternant -qi- before -n, marks 1 acting on 2
v. ma- marks 2 acting on 1, and takes precedence over, or
in Pike's term "outlines", ii. and iii.

Pike himself would no doubt carry the analysis further, but we wish here only to illustrate the method. Note that in the "final" matrix there is a residue involving i- and o-. A new matrix can sometimes be set up to examine, by the same method, such a remainder.

Although the exact pattern of distribution of formatives depends partly on how the parameters are set up, certain patterns tend to recur in a significant way:
1. A linear pattern, such as that for g-, tends to reflect marking of one parameter on one axis, in this case, 2nd sg object.

2. A block pattern, such as that for may-, tends to reflect marking of a combination of parameters, one on each axis, in this case, 2nd person acting on 1st person.

3. An L-shaped pattern, such as that for -n-, tends to reflect marking of one parameter on both axes ("involvement" marking), as in this case, involvement of dual, either as SUBJ or OBJ.²

4. A stairway pattern, such as that for the Koryak prefix ne- vs. its absence (Comrie 1980) shows in Table 3, tends to reflect hierarchical marking, as in this case, where the hierarchy is 1p x 2p x 3 sg x 3 pl: ne- occurs when the subject is lower on the (animacy) hierarchy than the object. For Algonquian examples of this see Morgan 1966.

Table 3. Koryak prefix ne-

<table>
<thead>
<tr>
<th>SUBJ</th>
<th>1p</th>
<th>2p</th>
<th>3p sg</th>
<th>3p pl</th>
</tr>
</thead>
<tbody>
<tr>
<td>1p</td>
<td>ne-</td>
<td>ne-</td>
<td>ne-</td>
<td></td>
</tr>
<tr>
<td>2p</td>
<td></td>
<td></td>
<td>ne-</td>
<td></td>
</tr>
<tr>
<td>3 sg</td>
<td></td>
<td></td>
<td>ne-</td>
<td>ne-</td>
</tr>
<tr>
<td>3 pl</td>
<td></td>
<td></td>
<td></td>
<td>ne-</td>
</tr>
</tbody>
</table>

(adapted from Comrie 1980: Table 3, p. 63)

The matrix permutation technique was applied to several languages in Erikson 1966 and 1965 and Pike and Erikson 1964. Hockett's negative response to the Potawatomi applications (Hockett 1966), which became quite well-known especially among Indianists, may be one reason for the subsequent lack of interest in the technique, at least as far as the literature reveals (but for a recent re-statement see Pike 1975). This criticism, however, had much more to do with the handling of the Potawatomi data than with the technique of analysis.

Another reason for subsequent neglect of the technique may well be that it is quite laborious. There seems to be no alternative to drawing a
new matrix after each permutation or two, increasing the likelihood of error each time. It should be obvious that, in principle, very complex data could be sorted out quite easily with a computer program and a CRT display.

Our program is written in FORTRAN, which, although it was not designed for character manipulation or list processing, seemed best for the initial experiment. Among its initial advantages are its portability (it can be run on many types of equipment) and its usability by untrained investigators. More important, the type of formatting required for matrix permutation turns out to be difficult to handle with other presently available programming languages.

An actual run of our program follows, except that since thirteen matrices are involved (our program will only carry out one permutation at a time) we will spare the reader all but the first and last; and we will spare him/her these also, since they are only slightly coded versions of Tables 1 and 2 respectively.

After the run, the program itself is given.

Finally, we close with a brief discussion of problems and limitations.

#FTN -MERN / HOMEWORK/ MATRIX

NO. OF ROWS YOU NEED
=10

NO. OF COLUMNS YOU NEED
=10

TYPE IN ROW 1 SEPARATED BY COMMA'S
= A,B,C,lag,1di,ipi,2ag,2di,2pl

TYPE IN ROW 2 SEPARATED BY COMMA'S
=pag,0,ku,1, may,may,may,x,x,x

TYPE IN ROW 3 SEPARATED BY COMMA'S
=3pag,0,ui,0,An,i,4,man,5a

TYPE IN ROW 4 SEPARATED BY COMMA'S
=1pd1,An,kAn,0pi,x,x,x,A,mAp,mAp1

TYPE IN ROW 5 SEPARATED BY COMMA'S

TYPE IN ROW 6 SEPARATED BY COMMA'S
=2pd1,4A,man,mAp,man,may,may,may,x,x,x

TYPE IN ROW 7 SEPARATED BY COMMA'S
=5pd1,1A,Ap1,man,mA

TYPE IN ROW 8 SEPARATED BY COMMA'S
=1pi1,i,kiw,ipi,x,x,x,A,mAp,mAp1

TYPE IN ROW 9 SEPARATED BY COMMA'S
=3pi1,i,lu,ipi,o,An,i,4,man,5a

TYPE IN ROW 10 SEPARATED BY COMMA'S
=1pag,ti,o,pi,x,x,x,A,mAp,man1

YOUR MATRIX IS
(coded version of Table 1)

TYPE COMMAND
=C 10 5
YOUR MATRIX IS...

TYPE COMMAND
=C 10 6
YOUR MATRIX IS...

TYPE COMMAND
=R 2 8
YOUR MATRIX IS...

TYPE COMMAND
=R 5 10
YOUR MATRIX IS...

TYPE COMMAND
=R 5 9
YOUR MATRIX IS...

TYPE COMMAND
=R 7 3
YOUR MATRIX IS...

TYPE COMMAND
=R 4 5
YOUR MATRIX IS...

TYPE COMMAND
=R 7 8
YOUR MATRIX IS...

TYPE COMMAND
=O 10 2
YOUR MATRIX IS...

TYPE COMMAND
=C 10 3
YOUR MATRIX IS...

TYPE COMMAND
=C 10 4
YOUR MATRIX IS...

TYPE COMMAND
=C 10 4
YOUR MATRIX IS

(coded version of Table 2)
Matrix Permutation

This program will read in the rows of an n by m matrix (for 0 <= n <= 10 and 0 <= m <= 10). Each entry in the matrix can be a string of up to four characters. The entries should be separated by commas.

Three kinds of commands can be given to the program, as follows:

1. R i j
2. C i j
3. $ 0 0

The first command will move row i to row j, while the rest of the rows are displaced in a circular fashion.

The second command will move column i to column j, while the rest of the columns are displaced in a circular fashion.

The last command will stop the program.

The original matrix and all of the subsequent matrices will be displayed.

The error "Illegal input string---- ignored" is the only error message which will be displayed. This error indicates that the user is not following the correct format of the commands. In the case of error, the Error Routine will issue this message and ignore the command.

CHARACTER *1 CHAR
CHARACTER $4 A(10, 10)
INTEGER M, R, X, Y
PRINT, 'NO. OF ROWS YOU NEED'
READ, M
PRINT, 'NO. OF COLUMNS YOU NEED'
READ, N
DO 20 I = 1, M
WRITE (42, 10) I
10 FORMAT (1X, 'TYPE IN ROW ', I2, ' (SEPARATED BY COMMA)')
READ, (AII, J), J = 1, N
20 CONTINUE
CALL OUT (A, M, N)
30 PRINT, 'TYPE COMMAND'
READ, CHAR, X, Y
IF ((CHAR .NE. 'R') .AND. (CHAR .NE. 'C') .AND. (CHAR .NE. '$'))
& 00 TO 40
IF ((CHAR .EQ. '5') .AND. ((X .GT. M) .OR. (Y .GT. N))) GO TO 40
IF ((CHAR .EQ. '6') .AND. ((X .GT. N) .OR. (Y .GT. M))) GO TO 40
IF (CHAR .EQ. 'G') GO TO 50
CALL MOVE (A, M, N, CHAR, X, Y)
CALL OUT (A, M, N)
GO TO 30
GO TO 50
40 PRINT 'ILLEGAL INPUT STRING --- IGNORED'
GO TO 30
50 STOP
END
SUBROUTINE OUT (A, M, N)
INTEGER M, N, I, J
CHARACTER *3 COL(10), ROW(10)
DO 10 I = 1, 10
COL(I) = 'DOC'
ROW(I) = 'ROW'
10 CONTINUE
PRINT 'YOUR MATRIX IS'
PRINT ' '
WRITE (42, 20)((COL(I), I = 1, N))
20 FORMAT (' ', 6X, 10(A3, Z, 1X))
PRINT ' '
DO 40 I = 1, N
WRITE (42, 20) ROW(I), I, (A(I, J), J = 1, M)
40 FORMAT (' ', A3, Z, 1X, 10(A4, Z))
PRINT ' '
40 CONTINUE
RETURN
END
SUBROUTINE MOVE (A, M, N, CHAR, X, Y)
CHARACTER *4 A(10, 10), TEMP
CHARACTER *1 CHAR
INTEGER M, N, X, Y, I, J, L
IF (CHAR .EQ. 'C') GO TO 60
IF (X .LT. Y) GO TO 30
DO 20 K = 1, X-Y
  I = X-K
  DO 10 J = 1, N
    TEMP = A(I, J)
    A(I, J) = A(I+1, J)
    A(I+1, J) = TEMP
 10 CONTINUE
20 CONTINUE
GO TO 120
30 DO 50 I = X, Y-1
  DO 40 J = 1, Z
    TEMP = A(I, J)
    A(I, J) = A(I+1, J)
    A(I+1, J) = TEMP
 40 CONTINUE
50 CONTINUE
This program is limited in certain ways. Because of the size of the customary CRT field, the attainable matrix size is too small. If one allows four character spaces per cell and two spaces between columns, as we have done in the foregoing, we have a maximum of eleven columns. Thus it was necessary to adapt even this data, which was deliberately chosen to be manageable in the run, nppp represents nppp+. Apn represents npn+. The program can be modified in this regard by changing lines 2, 30, 32, 41, 45 and 51. For example, if a user decides to represent the formats with arbitrary symbols -- using, say, one character per cell -- the number of possible columns increases to 36. This would of course necessitate constant decoding between permutations. CRTs with larger capacity are however now beginning to become available.

Another limitation on this particular program is that the numbering of rows and columns remains constant, as illustrated below. Suppose the command is: move row one to row three. The change will be, e.g.,

```
| R1 | a b c | R1 | d e f |
| R2 | d e f | R2 | a b c |
| R3 | a b c | R3 | a h c |
```
rather than: 

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2</td>
<td>d</td>
<td>e</td>
<td>f</td>
</tr>
<tr>
<td>R3</td>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
<tr>
<td>R1</td>
<td>a</td>
<td>b</td>
<td>c</td>
</tr>
</tbody>
</table>

This could be a serious problem: the formatives in a given row or column remain constant while their arrangement changes, but if two rows or two columns happened to have the same formatives, one could lose the distinction between them during the procedure. This problem is solvable in FORTRAN, but the resulting extra storage needed would have drastically increased the cost of running the program. For this reason, row 1 and column 1 were used for labelling at the time of input.

There are two further limitations which are caused by the nature of the language FORTRAN. First, each time a user implements the program it is necessary to type in the data. That is, if I work on Taos three times I must type in the data each time. Second, it is impossible to obtain a copy of only the last, or final, matrix. In a different programming language, these problems would be solvable.

It is obviously desirable to find or develop a more suitable programming language for matrix permutation. In addition it is desirable to allow more than one permutation at a time, to use more than two parameters (perhaps), and to be able to store and retrieve a portion of a given matrix as a separate matrix. In spite of these initial difficulties, in principle Pike's technique can be rather easily computerised. No doubt there are other such techniques, perhaps some due to this same highly original linguist, which would lend themselves usefully to the computer.

FOOTNOTES

1 A, B, and C are special classes of 3rd person singular object. Abbreviations: sg = singular; pl = plural; du = dual; 1, 2, 3 = person indices; p = person; OBJ = object; SUBJ = subject.

2 These results are not intended as a contribution to the study of Tiwa, but as part of a demonstration. We emphasize this in the hope of avoiding a confrontation under some such heading as "What Taowa is really like!"
REFERENCES


