COMPUTER PROCESSING
OF
MUSICAL SCORES

BY

UDAY G. GUJAR
AND
CLAUDIA A. CRAWFORD

TR85-029, SEPTEMBER 1985

ABSTRACT

This paper proposes an encoding scheme for printed music. A computerized system is designed and implemented to read music as human-encoded symbols, decode and store it within the computer, and to print it out either in its original key or in another key specified by the user. The graphics output is obtained by translating musical symbols into positional information and plotting it on a high resolution digital plotter. The rules for transposition, including those for transposing accidentals, are formalized. These have been programmed and appear to work correctly. The graphic output obtained is of high quality and should be acceptable for several purposes. The usefulness of the system would be increased immensely by providing a graphical input phase. Several plotter outputs of examples of musical scores along with the required coding are included.

1.0 INTRODUCTION

Computers have been used in the field of music as composing devices since their inception. They have also enjoyed fairly common use as a performance instrument since that time. In the past decade or so computers are also being used for the storage and printing of musical scores.

Prerau suggests several practical reasons for developing a system to read, process and print musical scores [PRER75]. These include such things as obtaining a piano reduction of an orchestral score, producing separate parts for each instrument from an orchestral score, producing new printings of works in old editions, printing newly composed music without the hand engraving process, transposing the music to any desired key, transferring existing scores for computer storage and/or transmission, etc.

Accompanists are often required to transpose music so that it will be in a more comfortable range for singers or instrumentalists. For a musician with a minimum of training, or a trained musician without the ability to do this by ear, it can be a laborious and lengthy operation. A computer system could be used in this situation very effectively.

One of the first systems to appear with music printing as a goal is the DARMS system [ERIC75]. The project involved inventing an encoding language for printed music and producing a master plate from which copies could be made. The DARMS system is now one of the major music encoding/printing systems available.

Another major encoding language is MUSTRAN [WENK74]. This language is intended to be mnemonic and similar to music notation so that it can be easily understood and used by musicians. It succeeds fairly well in this regard except that in some cases there is no similarity between the mnemonics and the musical symbols; for example, the accent (>) is coded as X24 which is not a very good mnemonic for accent.

At the University of Indiana, an integrated computer music software system has been developed [BYRD74,BYRD77,BYRD80]. It combines several programs developed there and elsewhere, mostly written in FORTRAN, to produce a powerful system that prints and analyses musical scores. Using MUSTRAN as an input language, Byrd's program SMUT produces plotted output of high quality; however, only one musical line is printed per stave. Also under development at the University of Indiana is an encoding system which allows music to be input directly to the computer via a sonic digitizer or organ keyboard [WITT77].

Effort is also made to use pattern recognition techniques for inputting the scores [PRER75]. Here, the music is read using a flying spot scanner, each symbol is isolated and a computer useable code is produced for the symbol once it is identified.

This paper describes a computerized music system [CRAW82] which reads music as human-encoded symbols, decodes and stores it and prints it out either in its original key or in any other specified key. An encoding scheme for this purpose is proposed and programmed.

2.0 OVERVIEW

The development of the musical system is broken into the following three stages:

(a) Input
(b) Coding
(c) Output

A simple user oriented coding scheme is proposed and implemented in the first stage. It is believed that the usefulness of the system would be enhanced if graphics input is available; however, this was not implemented because of lack of equipment, though the design has been proposed[CRAW82].

In the second stage, the encoded data is read and translated into the positional information appropriate for the musical score. At this point it is possible to apply a transformation to transpose the music to any desired key.

In the third stage, appropriate subroutines are called to plot the music using the plotting package developed at the University of New Brunswick [GUJA72, GUJA76]. This output looks like a standard musical score.

Music input to the system may be in any meter or key. Notes of any pitch from C_{-3} to C^4 can be input; the notes having a pitch outside this range are input using a directive such as 8 va. Duration of notes and rests may range from a dotted whole to a 64^{th} note or rest. All accidentals as well as common articulations are available. Character strings may also be input to write verses, labels, etc. The music must be written on two simultaneous staves using treble and bass clefs. Any number of vertically aligned parts are accommodated.

Once the music is properly encoded, it is translated into the X-Y coordinates so that it can be output, in any desired key, on a variety of plotting devices. The overall size of the score as well as the size of the notes can be controlled.

3.0 CODING

Figure 1 contains a short example of music to be encoded and the coding required. The first line of coding gives information concerning the desired size of the output plot. The second line gives some overall information about the piece including its key and meter. The third line inputs the title; the fourth and fifth lines code the two voices (soprano and bass), the pitch and duration of each note and the occurrences of bar lines. A more detailed explanation of the coding scheme follows.

There are two main categories of input variables, namely initialization variables and musical score variables (Figure 2). These are first described.

3.1 Initialization Variables

These are applicable to the entire piece of music and include such things as desired size of output, key and meter of the piece, title and composer, etc. (see Figure 3).

Plotting initialization variables are input on the first record. These comprise the desired height of each single staff (YLL), width of each page of music (WIDTH), length of each page of music (LENGTH), output device (DEV), number of verses of text to be printed (NV) and flags for extension of barlines through the centre of the grand staff and for a brace bracket in front of the grand staff (NVFLG). The value of YLL determines the size of all the musical symbols and characters. Therefore, changing the size of the symbols requires only a change in YLL. Appropriate default values are chosen if a value of zero is specified for any of these variables. The distance in the middle of each pair of staves depends upon the value of NV.

The second input record contains the program initialization variables and contains such information as key, number of sharps and flats in the key signature, time signature, smallest duration of a note in the music and parameters for transposition to a new key, if desired.

This could be followed by the optional input records containing title, composer's name, his dates and an instruction for how the piece is played. The type of these records is identified by the codes TL, CM, DT and IS respectively. These codes are followed by a character string giving the desired information which is placed at the appropriate locations on the music by the program.

3.2 Musical Score Variables

The second major category of input variables is the coded musical score variables. This may be divided into several categories including voice, notes, rests, character strings, other symbols, phrasing and ties, bar lines and end of input.

A summary of the encoding scheme for all these variables appears in Figure 4.

(a) Voice:

Voice is specified as S (for soprano), A (for alto), T (for tenor) or B (for bass) followed by a comma or a blank. The entire voice is then input from beginning to end before another voice is started.

(b) Note:

A note is indicated by the letter N and has four aspects, namely pitch, duration, accidentals and articulation or other symbols accompanying it.

Following "N" is the letter name of the note and the number of the octave it is found in with respect to middle C. Thus, middle C is coded as NC0 in both clefs. Notes in the treble clef in the first octave above middle C are coded with ones, in the next octave with twos, and so on up to four. Similarly, the bass clef notes in the first octave below middle C are coded with ones, in the next with twos and the next with threes. Notes higher or lower than these limits are obtained using the direction "8va" (coded as character data) as in published music. This directive "8va" stands for playing the note an octave higher (or lower) than written. An asterisk after the octave number indicates that a note is to be slightly displaced to avoid overlapping another note directly above.

Duration is specified as "U" and is followed by a character specifying the note type W (whole), H (half), Q (quarter), E (8th), S (16th), T (32nd) or Y (64th). The rarely occurring 128th note is not supported. Note type is followed by a number 0 to 3 indicating the note is not dotted (0), followed by one dot (1), followed by two dots (2) or is part of a triplet (3). Two more digits are specified for notes which are often grouped together by a beam; see Figure 5 for examples and [CRAW82] for more details.

Accidentals are coded with a "C" followed by the type as S (sharp), F (flat), N (natural), B (double flat), D (double sharp), NN (double natural), NF (natural-flat) and NS (natural-sharp).

Articulation and other symbols are coded with "Z" followed by U (for under) or O (for over) followed by the code S (for stoccato), T (for tenuto), A or H (for accents) and C (for staccatissimo). Normal mordent is coded as ZOR while inverted mordent is coded as ZUR. The code ZOU plots the turn over the note while ZUU plots the turn above and directly after the note.

(c) Rest:

Rests are encoded in a manner similar to notes using "R" instead of "N". They have two aspects, namely pitch and duration. The pitch portion is used to control the placement of the rest while the duration is coded as for notes.

(d) Character Strings:

Character strings are encoded with a "W" (for word) followed by a two character pitch code as in notes for placement purposes followed by the text.

(e) Miscellaneous Symbols:

Various miscellaneous symbols such as pause (PA), repeat sign (RP), begin crescendo (C*), end crescendo (C&), begin decrescendo (D*) end decrescendo (D&), coda (CO), sign (SI) are coded as "D" followed by a two character pitch code as in notes for placement purposes followed by the two character type code given above in brackets. At this stage, it is possible to specify the type as any single character which is outputed as a text.

(f) Phrasing and Ties:

Phrasing and ties are coded by a "P" followed by a digit indicating the nesting level, up to three. This is followed by an "O" (for over) or "U" (for under). Optionally, up to two more characters may follow indicating ending of the phrase, improper nesting or continuation from one line to another.

(g) Barlines:

Barlines may be single ("Q") or double ("V"); the last voice is coded with "X" (single) or "Y" (double) for the barlines. Pause, repeat dots, coda sign and section repeat signs associated with barlines are coded as YP, YR, YC, YS (etc.) respectively.

(h) End:

End of input is signified by "9" which causes termination and output of one musical piece.

4.0 INTERNAL REPRESENTATION

The information held in the code must be translated into the positional X-Y coordinates for processing and plotting. This section briefly explains how this is achieved.

4.1 Pitch

The pitch controls the placement (of a note, rest, etc.) on, above or below the grand staff and hence is used to determine the Y coordinate. Figure 6 shows the arrangement of lines and spaces and the name of each note. It can be seen that only seven different letters are used to represent all the notes in an octave. Two additional letters (W and V) are used for the placement of text.

A three dimensional array PITCH (I,J,K), with $1 \leq I \leq 9$, $1 \leq J \leq 2$, $1 \leq K \leq 5$, is initialized to contain all Y coordinates associated with pitches. The entries are referenced by I - an integer code for note name, J - an integer code for clef type and K - an integer code for octave number. For example, PITCH (3,1,2) refers to note E (I=3) in the treble clef (J=1) in the first octave (K=2) above middle C. From Figure 6, it can be seen that PITCH (3,1,2) will have a value 91 which represents E1 in treble clef; this represents the Y coordinate of the centre of the note head.

4.2 Duration

Duration determines the X coordinate (of a note, rest, etc.). A spacing of 20 units is assigned to the note type of smallest duration which is supplied as an input data. This is the minimum distance which allows two accidentals to be placed in front of a note. A space of multiples of 20 units is assigned to notes of longer duration. For example, if an eight note is the smallest duration note, then a quarter note is assigned 40 units, a half note 80 units and a full note 160 units. These spacings remain constant throughout the musical piece being processed. Of course, this spacing is adjusted appropriately for dotted notes and members of a triplet grouping.

4.3 Stems and Beams

Stems and beams are drawn, if required, after the note head is drawn; their position is determined from the X-Y coordinates of the notehead.

If the grouping to be beamed contains more than two notes, a least square routine [GERA78] is used to find the line of best slope through the note heads. The Y intercept of the line is then adjusted to allow a minimum of 15 units of stem length to connect each note to the beam. Stems are lengthened, if necessary, so that they will extend to the beam. Complex beaming figures require a slightly more complicated procedure. Complete details and the beaming algorithm itself are given in [CRAW82].

4.4 Other Musical Entities

Accidentals, ornaments, articulations, character strings and other symbols are drawn using the X-Y coordinates of the notehead.

Variable length phrases and ties are drawn as half-ellipses having variable major and minor axes. A least squares line is computed through the centres of the noteheads of the notes under the phrase. This is taken as the diameter of the major axis of the ellipse. The minor axis is computed as 15% of the length of the major axis, to a maximum of 20 units. A check is made to ensure that the ellipse does not begin or end within a notehead; in case it does, the position of the entire ellipse is shifted. This shifting is also employed to avoid collision with stems (though not always completely successfully).

5.0 TRANSPOSITION

Transposition is defined as changing the key in which a piece of music is performed, thus, changing the set of notes it uses. This process is broken into two phases. The first (and the simplest) is transposition of notes and the second (and fairly complex) is transposition of accidentals.

5.1 Transposition of Notes

A numerical code for the name of each key is found from the number of sharps or flats in the key signatures of the old key (in which the piece is written in) and the new key (in which the piece is to be transposed to). These two members are subtracted; the result, TRANSN, is the number of consecutive differently named notes between the two keys. If TRANSN is negative, the transposition is to a lower pitch, otherwise to a higher pitch. Every time a note is encountered in the input, it is adjusted by this number TRANSN so that it is transposed to the new key.

Transposition is always done to the closest new key. For example, to transpose from C major to F major, all notes are raised by four notes rather than lowering by five. The key names found are always major key names even if the keys are minor, since TRANSN is the same in both cases.

5.2 Transposition of Accidentals

Since no rules are available, as far as the authors could determine, to transpose the accidentals, the more musically knowledgeable author formalized the rules for this purpose and then programmed them. These rules are fairly complex so the subroutine that implements them is fairly involved. See [CRAW82] for more details.

6.0 GRAPHICS OUTPUT

The computer plotting package developed at the University of New Brunswick [GUJA72, GUJA76] is used to draw the musical symbols. It is a completely device independent package which supports a variety of output devices.

The following routines from the package are used:

AREA	-	defines physical area of the plot
SETPLT	-	defines user units on the X and Y axes
NOWPLT	-	draws a straight line or a point
CIRCLE	-	draws a circular arc
ELLIPS	-	draws an elliptical arc
CHARS	-	draws text

Each notehead is an ellipse with a major axis of 9 units in diameter and a minor axis of 6 units in diameter. Black noteheads are filled in by plotting successively smaller ellipses with the same centre. The number of ellipses depends upon the size of the notehead. A notehead of standard size (i.e. 0.06 inch) requires five ellipses to fill it.

Stems and beams are drawn by a routine which draws a straight line; beams are thickened by five successive calls to this routine with a slight shift in between.

Whole, half and quarter rests consist entirely of straight line segments and are drawn by NOWPLT. Rests of shorter duration have circular segments referred to as bulbs drawn by calls to CIRCLE. These bulbs are filled in by successive calls to CIRCLE.

Dots after the notes and the rests are drawn by CIRCLE and filled in in a similar fashion.

All accidentals consist of straight line segments only and are drawn by NOWPLT. This is also true for tenuto sign, accents, mordents, section repeat signs, bar lines, ledger lines and the staff. Staccato dots are drawn by CIRCLE. Pause consists of one large semicircle and a dot - all generated by calls to CIRCLE. Coda sign consists of an ellipse and two straight lines. Phrases and ties are elliptical arcs drawn by calls to ELLIPS. Turns, signs and clefs are drawn by several calls to NOWPLT and CIRCLE.

All character strings including the title, composer, text and time signature are drawn by calls to CHARS.

7.0 EXAMPLES

Several examples of outputs of music along with the coding required are given in this section.

7.1 Amen

The "Twofold Amen" shown in Figure 7, b, d and e is similar to a hymn. Figure 7a contains the published version, and Figure 7b contains output obtained from the plotter. The coding required for this output can be found in Figure 7c.

Figure 7b shows many aspects of plotted music. Aside from the appearance of various musical symbols such as clefs, key signature, time signature, and notes, it shows four vertically aligned voices. Also demonstrated are phrasing, ties, and text placement.

Figures 7d and 7e show the same piece of music transposed to two different keys, G^b major and D major respectively. The latter is also enlarged. The coding changes required are also shown.

7.2 Canons

Figures 8 and 9 show examples of two Canons composed by Carleton Elliott [ELLI80].

The published version of "Canon 1" is shown in Figure 8a, while the output obtained from the plotter is shown in Figure 8b. The coding required for this output is given in Figure 8c.

This example demonstrates computer drawn half and quarter rests as well as whole, half, quarter, and eighth notes. The latter occur both as single notes and as notes beamed together in groups. Staccato notes as well as the longer phrases are also illustrated.

"Canon 4" is shown in Figures 9a and b, with the published version appearing in the former figure and plotter output in the latter. This example illustrates more complex beaming, and shows phrasing which is broken by line endings. The coding required for the plotter output is given in Figure 9c.

7.3 Further Examples

This section contains two examples written to illustrate several features which are not shown in the previous figures. (Note that neither of these examples is intended as a composition to be performed).

The "Two Part Song" shown in Figure 10a is output from the plotter and illustrates three additional rhythmic figures, those being the triplet and various groupings. Also demonstrated are an accent, a tenuto sign, a pause, a crescendo sign, dynamic and speed indications as text, an eighth rest and further examples of phrasing and ties. The coding required for this example is found in Figure 10b.

The "Example 1" shown in Figure 11a is also output from the plotter and presents all types of basic note and rest durations available in the system. Also illustrated are a mordent, a turn, a D.S. sign, a coda sign, repeat signs and accidentals. Coding required for this example is found in Figure 11b.

8.0 CONCLUDING REMARKS

An encoding scheme for inputting musical scores is proposed. A user oriented system to store these encoded musical symbols and plot a score in any desired key has been designed and implemented. The code is easy for the novice musician to use as an extensive knowledge of music theory is not required. As well, the user does not have to be an experienced computer programmer. The code is mnemonic, so that anyone who knows the names and duration of notes and rests and the names of musical symbols should not have to spend a great deal of time learning it, and can input the data easily and rapidly. If an input error of large enough magnitude to stop the program is made, a message is printed informing the user exactly where the error lies.

It is believed that the system is most useful in the area of transposition, which is a dull and laborious task well suited for computer. Unfortunately, present copyright laws prohibit its general usage for this purpose.

The system should also be of use to composers as a quick and relatively inexpensive way to see their work in print. Performers of new works who have had to read from hand printed scores would be prompt to point out this advantage of computerized music transposition and printing.

The graphics output generated by the system is of good quality, and is comparable to published musical scores.

The system can be enhanced considerably by adding a graphics input device so that the user is not required to encode manually. A design for a graphics input phase is given in [CRAW82]; implementation work is contemplated.

Various other enhancements such as design of many more commonly occurring musical symbols, dynamic adjustment of horizontal spacing, inclusion of homophonic music [ELSO09], changing signatures, a facility to accept both single and multiple staff music (as is found in orchestral or organ music) etc. are also possible.

ACKNOWLEDGEMENTS

The authors would like to thank Prof. Thomas A. Austin for valuable help rendered during the design and development phases. His contributions to ensure musical correctness is appreciated. Thanks are also due to Mr. Anuj Gujar and Miss Anaya Gujar for suggesting enhancements to Figure 6 and pointing out an error in the process and to Mrs. Arlene Stocek and Mrs. Janice Price for typing the paper.

REFERENCES

[BYRD74] Byrd, D. "A System for Music Printing by Computer". Computers and the Humanities, Vol. 8, No. 3, pp. 161-172, 1974.

[BYRD77] Byrd, D. "An Integrated Computer Music Software System". Computer Music Journal, Vol. 1, No. 2, pp. 55-60, April, 1977.

[BYRD80] Byrd, D. "Human Engineering in a Portable Music Notation System". Private communication.

[CRAW82] Crawford, C.A. "Computer Encoding, Storage and Transposition of Musical Scores", M.Sc.(C.S.) Thesis, School of Computer Science, University of New Brunswick, Fredericton, N.B., Canada, 117 pages, 1982.

[ELLI80] Elliott, C. "Seventeen Canons for the Early Grades". Waterloo Music Company Limited, Waterloo, Ontario, 1980.

[ELSO09] Elson, L.C. "Elson's Pocket Music Dictionary". Oliver Ditson Company, Pennsylvania, 1909.

[ERIC75] Erickson, R.F. "The DARMS Project: A Status Report". Computers and the Humanities, Vol. 9, pp. 291-298, 1975.

[GERA78] Gerald, C.F. "Applied Numerical Analysis". Addison Wesley Publishing Company, Inc., 1978.

[GUJA72] Gujar, U.G. "Computer Plotting". University of New Brunswick Computing Centre, 212 pages, Oct. 1972, 10th printing, Oct. 1984.

[GUJA76] Gujar, U.G. "A Device-Independent Computer Plotting System". Proceedings ACM Symposium on Graphics Languages, Computer Graphics, Vol. 10, No. 1, pp. 85-100, April, 1976.

[PRER75] Prerau, D. "DO-RE-MI: A Program That Recognizes Music Notation". Computers and the Humanities, Vol. 9, pp. 25-29, 1975.

[WENK74] Wenker, J. "Mustran II: A Foundation for Computational Musicology". Computers in the Humanities, ed. L. Mitchell, Edinburgh, 1974.

[WITT77] Wittlich, G., Byrd, B., Nerheim, R. "A System for Interactive Encoding of Music Scores Under Computer Control". Computers and the Humanities, Vol. 11, 1977.

Coding: 0.3 0.0 0.0 PLOEB

4 4 Q

TLEXAMPLE$E

S,NF 1UQ0,NG 1UQ0,NA 1UQ0,NB 1UQ0,NC 1U W0,V

B,NG 1U W0,X,NC 1U W0,Y,9

Figure 1: An Example of a Musical Score and Coding

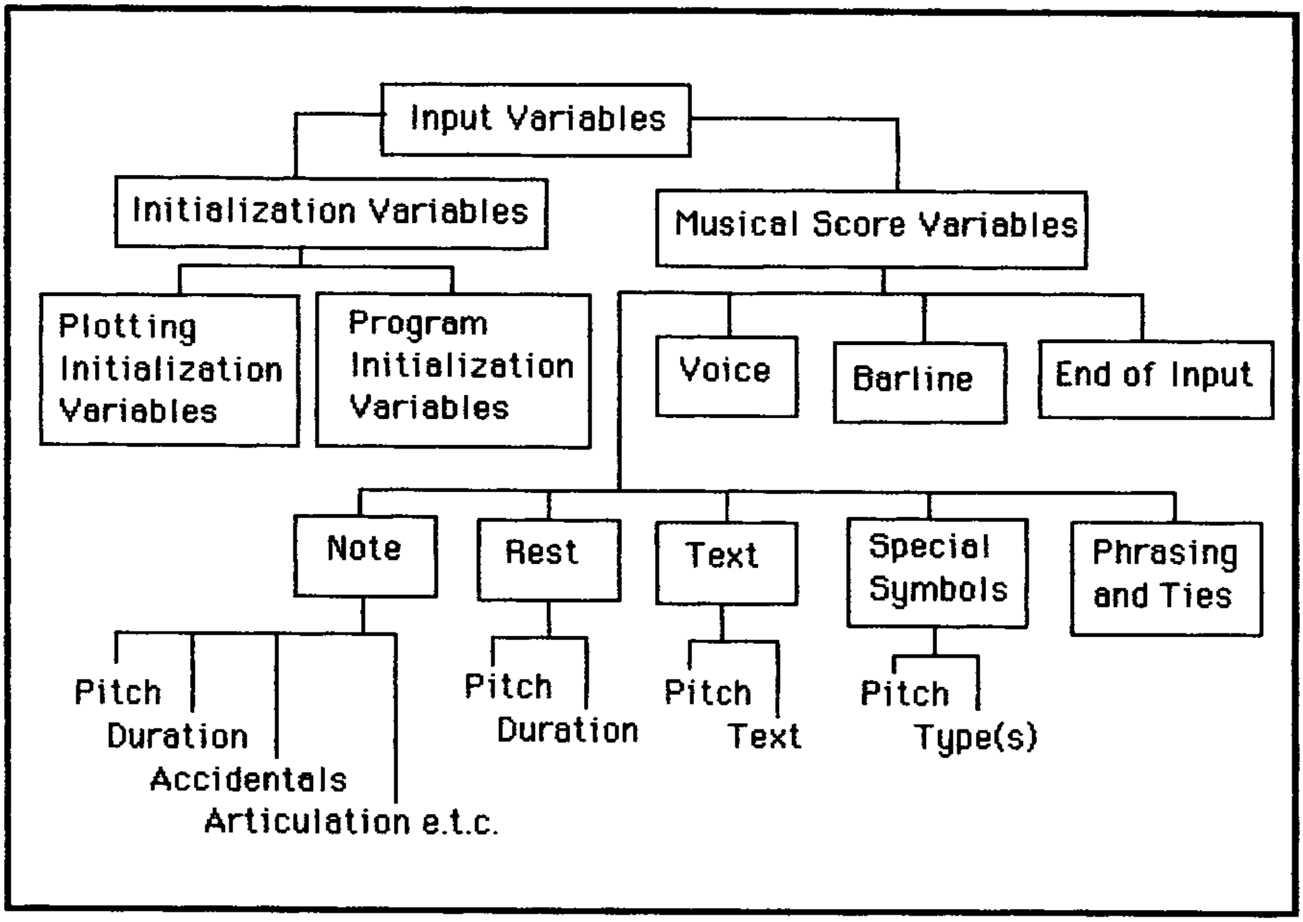


Figure 2: Overview of Input Variables

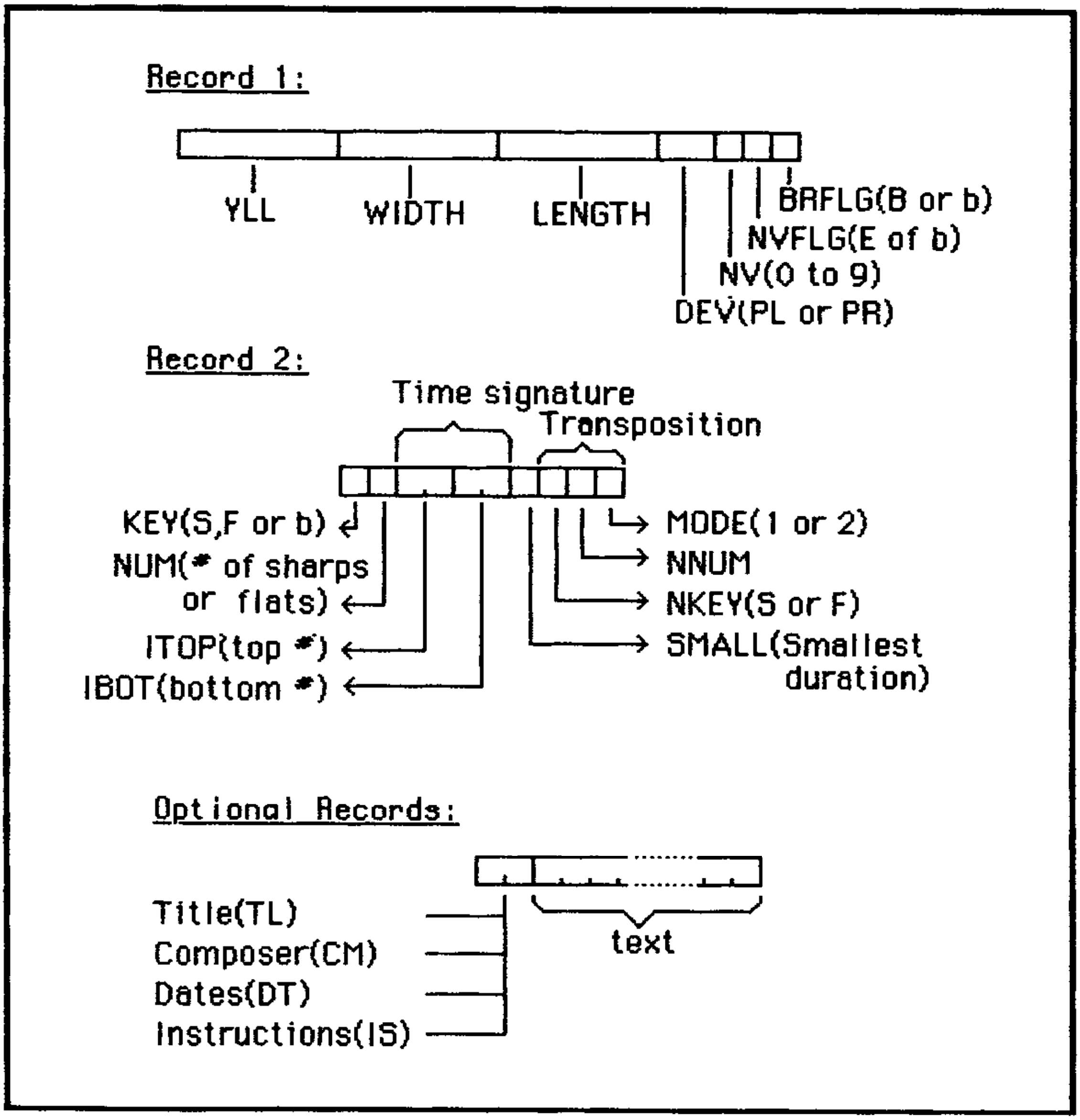


Figure 3: Initialization Variables

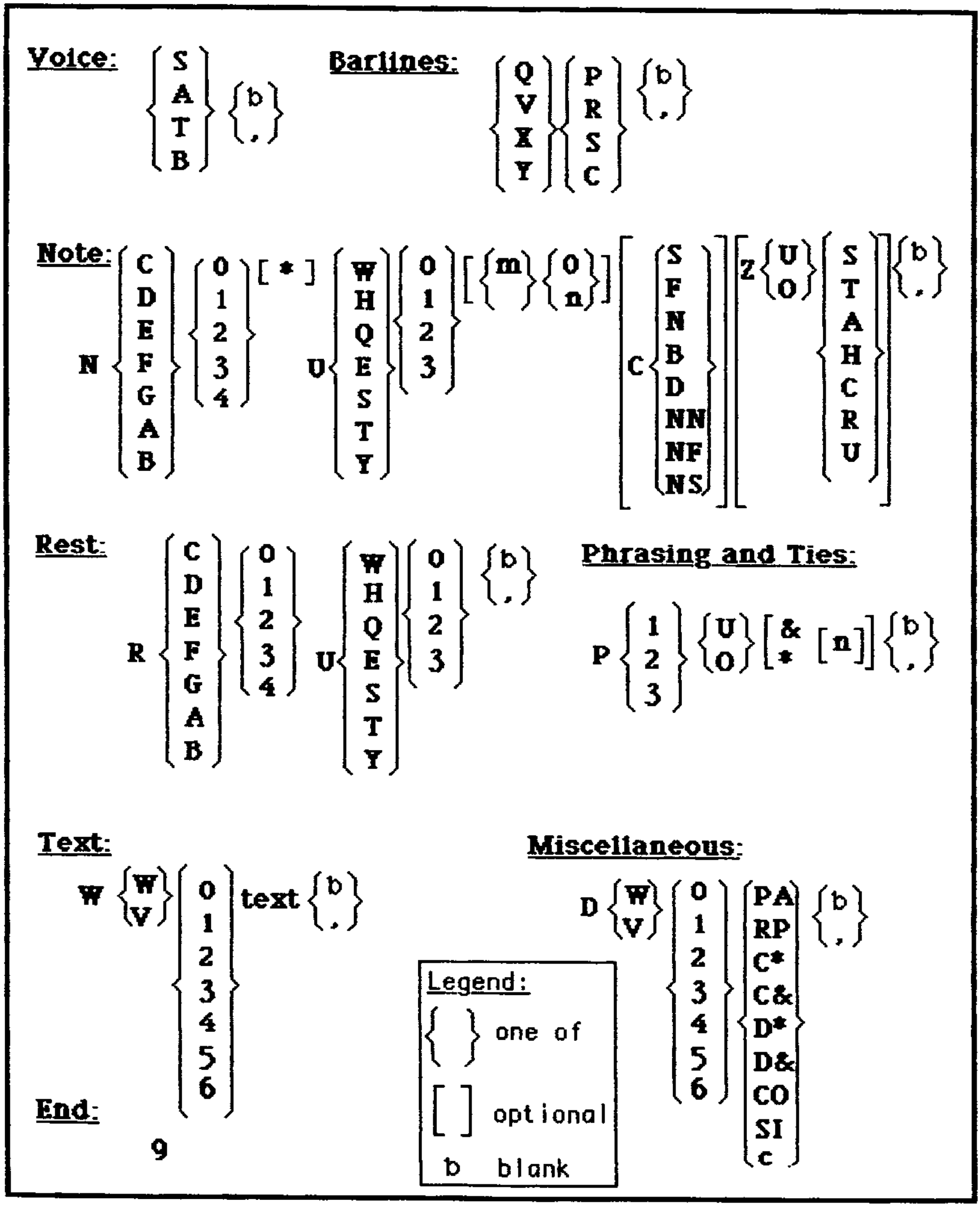


Figure 4: Encoding of Musical Score Variables

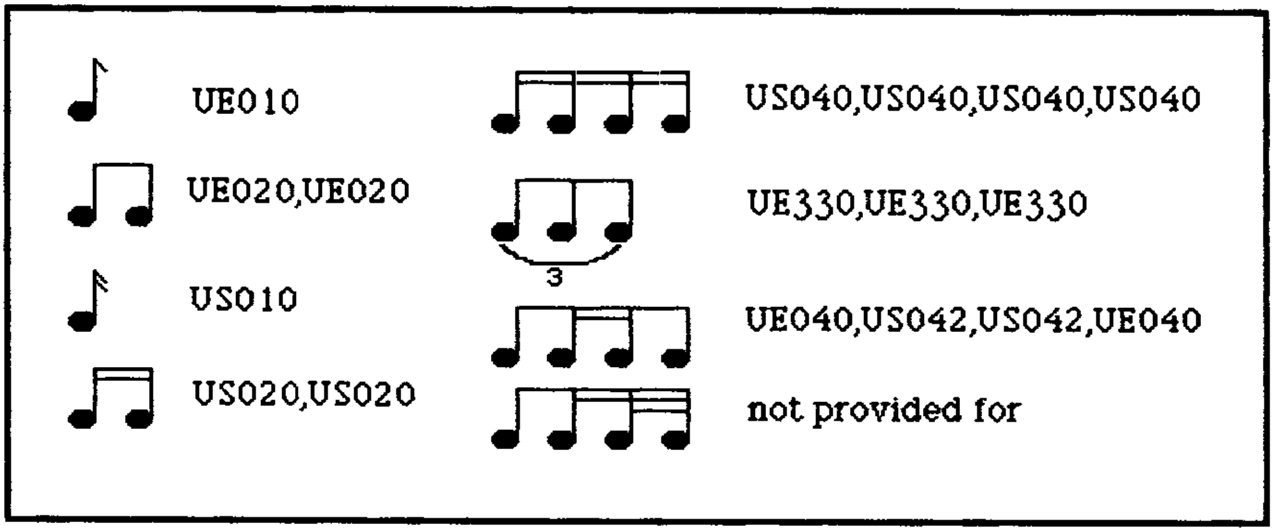


Figure 5: Some Examples of Beaming

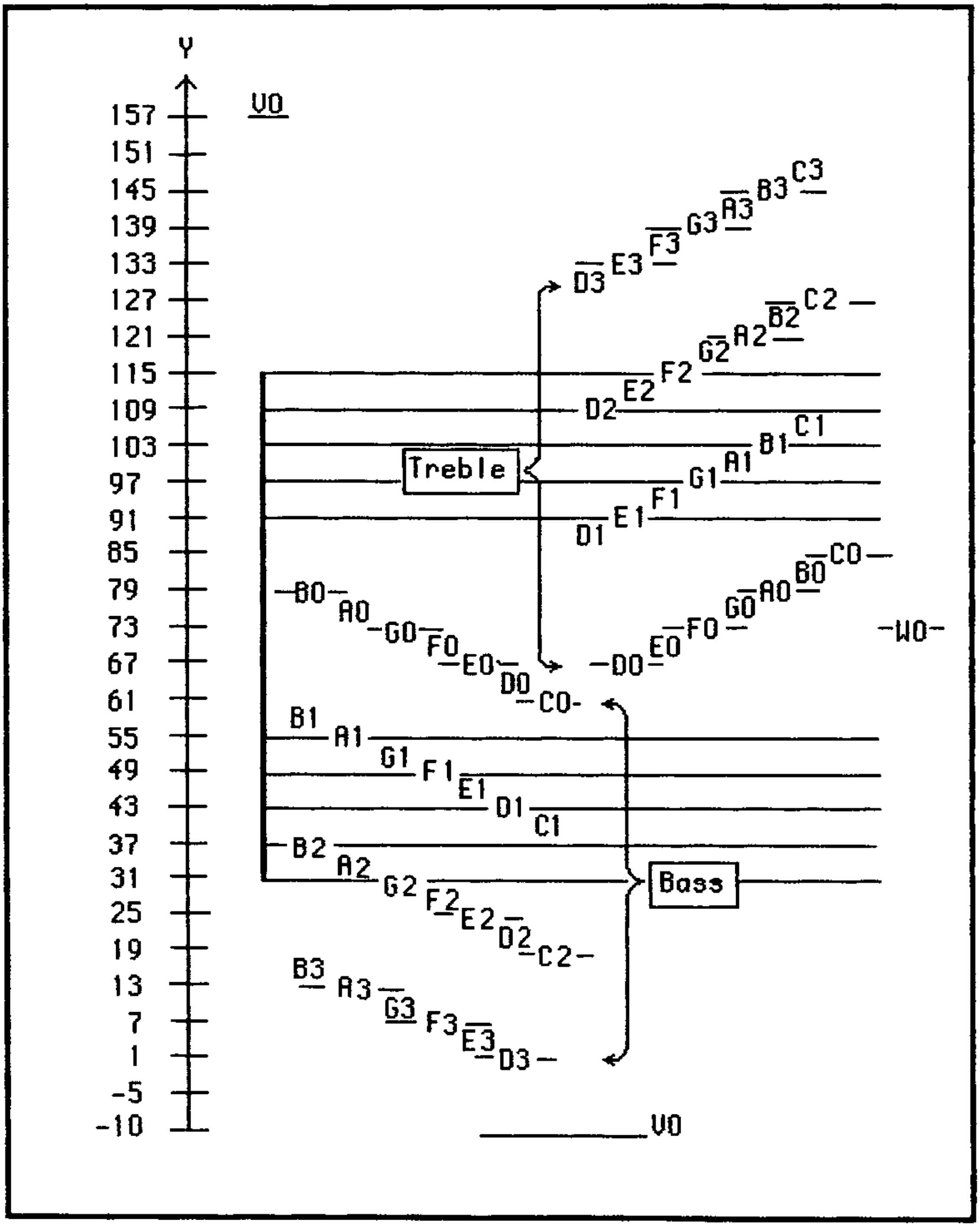


Figure 6: Internal Representation of Pitch

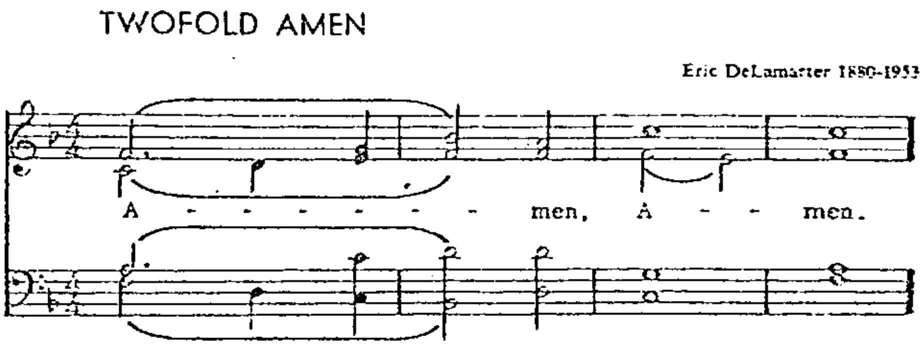


Figure 7a: Amen - Published version

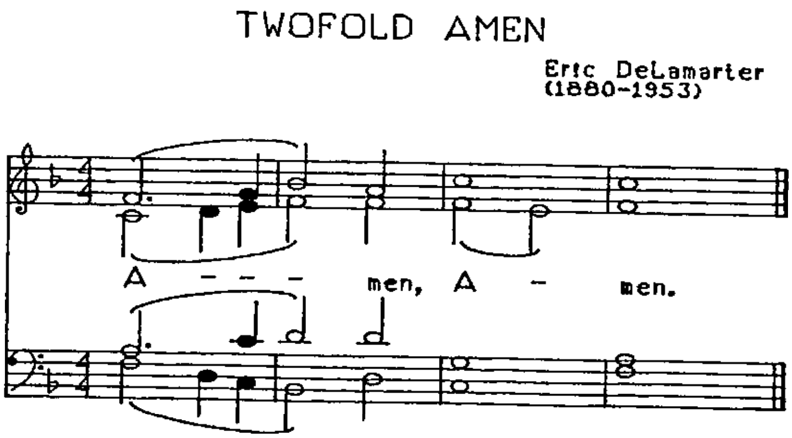


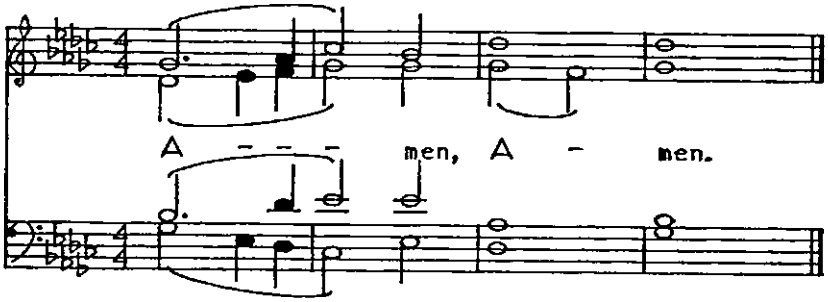
Figure 7b: Amen - Plotter output

```
Ø.Ø          Ø.Ø          Ø.Ø          PL1
F1 4 4Q
TLTWOFOLD AMEN$E
CME$LRJC $UD$LE$UL$LAMARTER$E
DT(188Ø-1953)$E
S,NF1UH1,P1O,NG1UQØ,Q,NB1UHØ,P1O&,NA1UHØ,Q,NC1UWØ,Q,NC1UWØ,
V,A,NCØUHØ,P1U,DW1A,ND1UQØ,DW1-,NE1UQØ,DW1-,Q,NF1UHØ,P1U&,
DW1-,NF1UHØ,WW1
 $LMEN,
Q,NF1UHØ,P1U,DW1A,NE1UHØ,P1U&,DW1-,Q,NF1UWØ,WW1
 $LMEN.
V,T,NA1UH1,P1O,NCØUQØ,Q,NDØUHØ,P1O&,NDØUHØ,Q,NG1UWØQ,
,NA1UWØ,V,B,NF1UHØ,P1U,ND1UQØ,NC1UQØ,
X,NB2UHØ,P1U&,ND1UHØ,X,NC1UWØ,X,NF1UWØ,Y,9
```

Figure 7c: Amen - Coding required for Figure 7b

TWOFOLD AMEN

Eric DeLamarter
(1880-1953)

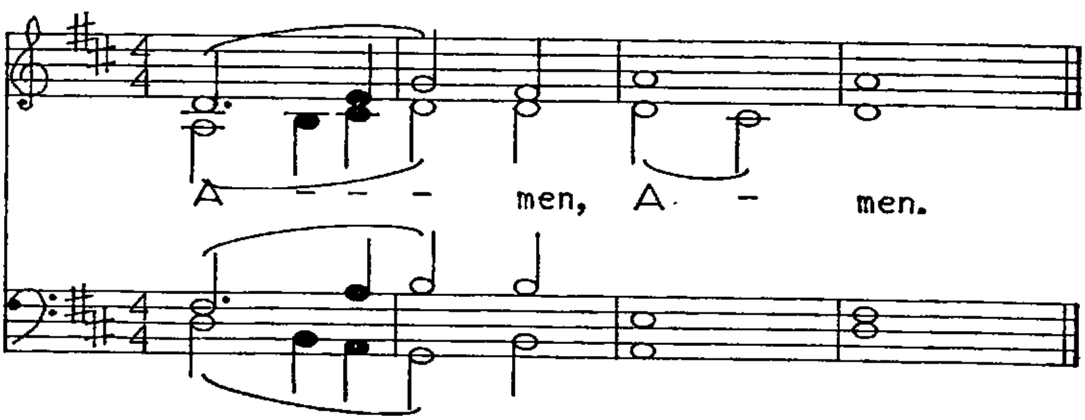


Coding change: 2nd data record: F1 4 .4QF6

Figure 7d: Amen - Transposed up to G flat major

TWOFOLD AMEN

Eric DeLamarter
(1880-1953)



Coding change: first two records: 0.3
F1 4 4QS2

Figure 7e: Amen - Enlarged and transposed down to D major

Eight Canons for Grade Two

1.

Used by permission

Figure 8a: Canon 1 - Published version

CANON NUMBER 1

Carleton Elliott

Figure 8b: Canon 1 - Plotter output

```
Ø.          1Ø.         Ø.          PL1EB
SØ 4 4E
TLCANON NUMBER 1$E
CMC$LARLETON $UE$LLLIOTT$E
ISQ=1Ø8-112$E
S,NA1UEØ4Ø,P1O,NG1UEØ4Ø,NA1UEØEØ,NB1UEØ4Ø,Q,NC1UQØZUS,P1O&,NG1UQØZUS,
NA1UQØ,P1U,NG1UEØ2Ø,NF1UEØ2Ø,Q,NE1UHØ,P1U&,RB1UHØ,Q,NG1UQØ,P1U,
NF1UEØ2Ø,NE1UEØ2Ø,ND1UEØ4Ø,NCØUEØ4Ø,ND1UEØ4Ø,NE1UEØ4Ø,Q,ND1UHØ,
NG1UHØ,P1U&,Q,RF2UQØ,NG1UQØZUS,NA1UQØZUS,NB1UQØZUS,Q,NC1UQØ,P1O,
NB1UEØ2Ø,NA1UEØ2Ø,NG1UQØ,NF1UQØ,Q,NE1UQ1,NF1UEØ1Ø,NG1UQØ,P1O&,NA1UEØ2Ø,
P1O,NB1UEØ2Ø,Q,NC1UQØ,NB1UEØ2Ø,NA1UEØ2Ø,NG1UEØ4Ø,NF1UEØ4Ø,NE1UEØ4Ø,
ND1UEØ4Ø,Q,NCØUWØ,P2U,Q,NCØUHØ,P2U&,P1O&,V,B,RD1UHØ,X,RD1UHØ,NA1UEØ4Ø,
P1O,NG1UEØ4Ø,NA1UEØ4Ø,NB1UEØ4Ø,X,NCØUQØZOS,P1O&,NG1UQØZOS,NA1UQØ,P1O,
NG1UEØ2Ø,NF1UEØ2Ø,X,NE1UHØ,P1O&,RD1UHØ,X,NG1UQØ,P1O,NF1UEØ2Ø,NE1UEØ2Ø,
ND1UEØ4Ø,NC1UEØ4Ø,ND1UEØ4Ø,NE1UEØ4Ø,X,ND1UHØ,NG1UHØ,P1O&,X,RA1UQØ,
NG1UQØZOS,NA1UQØZOS,NB1UQØZOS,X,NCØUQØ,P1O,NB1UEØ2Ø,NA1UEØ2Ø,NG1UQØ,
NF1UQØ,X,NE1UQ1,NF1UEØ1Ø,NG1UQØ,P1O&,NA1UEØ2Ø,P1O,NB1UEØ2Ø,X,NCØUQØ,
NB1UEØ2Ø,NA1UEØ2Ø,NG1UEØ4Ø,NF1UEØ4Ø,NE1UEØ4Ø,ND1UEØ4Ø,X,NC1UHØ,P1O&,Y,9
```

Figure 8c: Canon 1 - Coding required

4.

Used by permission

Figure 9a: Canon 4 - Published version

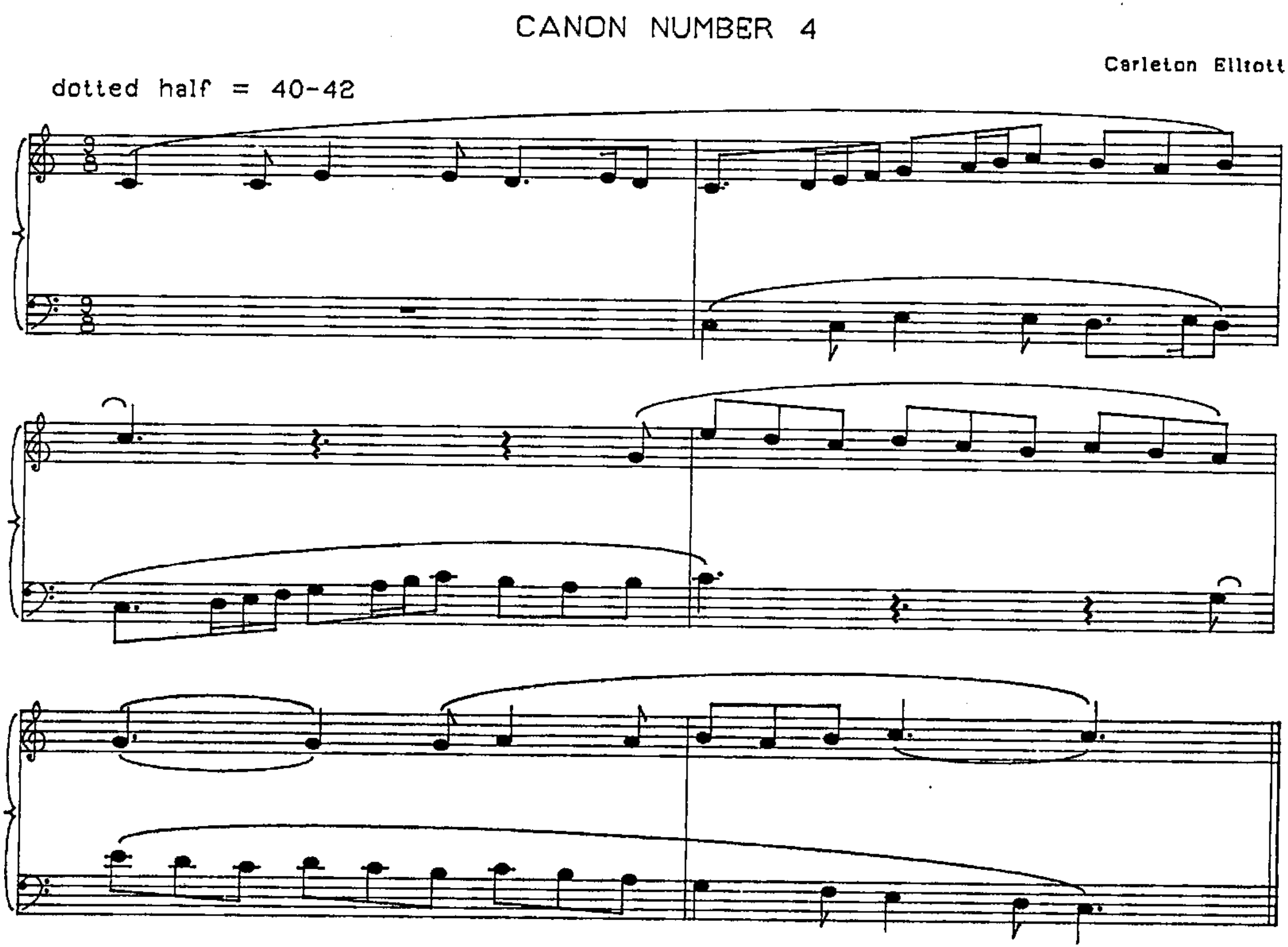


Figure 9b: Canon 4 - Plotter output

```
Ø.         Ø.         Ø.         PLIEB
SØ 9 8S
TLCANON NUMBER 4$E
CMC$LARLETON $UE$LLLIOTT$E
IS$LDOTTED HALF = 40-42$E
S,NCØUQØ,P1O,NCØUEØ1Ø,NE1UQØ,NE1UEØ1Ø,ND1UE13Ø,NE1USØ31,ND1UEØ3Ø,Q,
NCØUE14Ø,ND1USØ43,NE1USØ43,NF1USØ43,NG1UEØ4Ø,NA1USØ42,NB1USØ42,NC1UEØ4Ø,
NB1UEØ3Ø,NA1UEØ3Ø,NB1UEØ3Ø,P1O&,Q,NC1UQ1,P1O2,RF2UQ1,RF2UQØ,NG1UEØ1Ø,
P1O,Q,NE2UEØ3Ø,ND2UEØ3Ø,NC1UEØ3Ø,ND2UEØ3Ø,NC1UEØ3Ø,NB1UEØ3Ø,NC1UEØ3Ø,
NB1UEØ3Ø,NA1UEØ3Ø,P1O&,Q,NG1UQ1,P1O,P2U,NG1UQØ,P2U&,P1O&,NG1UEØ1Ø,P1O,
NA1UQØ,NA1UEØ1Ø,Q,NB1UEØ3Ø,NA1UEØ3Ø,NB1UEØ3Ø,NC1UQ1,P2U,NC1UQ1,P2U&,P1O&,
V,B,RF1UWØ,X,NC1UQØ,P1O,NC1UEØ1Ø,NE1UQØ,NE1UEØ1Ø,ND1UE13Ø,NE1USØ31,
ND1UEØ3Ø,P1O&,X,P1O,NC1UE14Ø,ND1USØ43,NE1USØ43,NF1USØ43,NG1UEØ4Ø,
NA1USØ42,NB1USØ42,NCØUEØ4Ø,NB1UEØ3Ø,NA1UEØ3Ø,NB1UEØ3Ø,X,NCØUQ1,P1Ø&,
RA1UQ1,RA1UQØ,NG1UEØ1Ø,P1O1,X,NEØUEØ3Ø,P1O,NDØUEØ3Ø,NCØUEØ3Ø,NDØUEØ3Ø,
NCØUEØ3Ø,NB1UEØ3Ø,NCØUEØ3Ø,NB1UEØ3Ø,NA1UEØ3Ø,X,NG1UQØ,NF1UEØ1Ø,
NE1UQØ,ND1UEØ1Ø,NC1UQ1,P1O&,Y,9
```

Figure 9c: Canon 4 - Coding required

Figure 10a: Two Part Song - Plotter output

```
Ø.Ø          Ø.Ø          Ø.Ø          PL1EB
FØ 3 4S
TLTWO PART SONG$E
CMC$LRAWFORD, $UC.$E
DT(1956-     )$E
IS$LALLEGRO$E
S,WDØ
 $LMF
NG1USØ4ØZUA,DW1C*,NA1USØ4Ø,NB1USØ4Ø,NC1USØ4Ø,NC1UEØ2ØZUS,
NE2UEØ2ØZUS,NG1UEØ2ØZUS,NB1UEØ2ØZUS,DW1C&,Q,NC1UEØ3ØZUS,WW1
 $LF
NG1USØ32,NF1USØ32,NE1USØ4Ø,NC1USØ4Ø,ND1USØ4Ø,NB1USØ4Ø,NC1USØ4Ø
NG1USØ4Ø,NE1USØ4Ø,NG1USØ4Ø,Q,NCØUH1,Q,ND1UE33Ø,P1U,WW1
 $LRIT. E DIM.
NE1UE33Ø,NF1UE33Ø,P1U&,NG1UE33Ø,P1U,NA1UE33Ø,NB1UE33Ø,P1U&,
NC1UQØ,V,B,RD1UHØ,NG1USØ4Ø,NF1USØ4Ø,NE1USØ4Ø,ND1USØ4Ø,
X,NC1UEØ1ØZOT,RE1UEØ,NG1UEØ2ØZOS,NG2UEØ2ØZOS,NC1UQØ,P1U,P2O,X,
NC1USØ4Ø,P2O&,NG1USØ4Ø,NE1USØ4Ø,NG1USØ4Ø,NC1UHØ,P1U&,X
NG1UEØ2Ø,NF1UEØ2Ø,NE1UE12Ø,ND1USØ21,NC1UQØ,YP,9
```

Figure 10b: Two Part Song - Coding required

EXAMPLE 1

Figure 11a: An Example - Plotter output

```
Ø.          Ø.          Ø.          PLØE
S      Y
TLEXAMPLE 1$E
S,DWØRP,NB1UWØZOC,DWØCO,NB1UHØCS,NB1UQØCN,NB1UEØ1ØCD,
NB1USØ1ØCB,NB1UTØ1Ø,NB1UYØ1Ø,Q,V,
B,RF1UWØ,RD1UHØ,RA1UQØ,RE1UEØ,RE1USØ,RE1UTØ,RE1UY1,XS,
ND1UEØ2Ø,ND1UEØ2Ø,ND1USØ2ØZOU,ND1USØ2Ø,ND1UTØ2ØZOR
ND1UTØ2Ø,ND1UYØ2Ø,ND1UYØ2Ø,YR,9
```

Figure 11b: An Example - Coding required