

Rochester Institute of Technology

## RIT Digital Institutional Repository

---

### Theses

---

1991

## An Expert system to generate musical variations in the style of Telemann

Kathy Johnson Merck

Follow this and additional works at: <https://repository.rit.edu/theses>

---

### Recommended Citation

Merck, Kathy Johnson, "An Expert system to generate musical variations in the style of Telemann" (1991). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact [repository@rit.edu](mailto:repository@rit.edu).

Rochester Institute of Technology  
School of Computer Science and Technology

An Expert System to Generate Musical Variations  
in the Style of Telemann

Kathy Johnson Merck

A thesis, submitted to  
The Faculty of the School of Computer Science and Technology  
in partial fulfillment of the requirements for the degree of  
Master of Science in Computer Science

Approved by:

Professor John A. Biles

Professor Walter A. Wolf

Professor Peter G. Anderson

July 25, 1991

## ABSTRACT

Musical performance practice during the Baroque period (approximately 1600 - 1750) required that a solo performer be adept at adding extemporaneous variations on the composer's original melody during performance. Atypically, George Phillip Telemann, in his "Methodical Sonatas", provided, along with the original melody line, a written-out variation of that line. Using knowledge of the Baroque style in general and that gleaned from an analysis of Telemann's variations in particular, this expert system models a Baroque musician's (expert's) process of creating such variations.

The variation process (programmed in Prolog) involves transformations to a list structure which represents the piece. The system "chooses" these transformations based on general rules, rhythmic and melodic vocabularies, and the harmonic framework of the specific piece. In addition, global musical considerations such as the duplication of musical sequences found in the original and the creation of overall musical continuity are addressed during the generation process through a multiple viewpoints approach.

To test the system, two variations were generated for each of ten musical movements. The expert has judged that syntactically the system works as anticipated, and semantically is generally successful; there remain possibilities, however, for refinement in replicating a more complete spectrum of human musical thought.

## TABLE OF CONTENTS

1. Introduction.....	1
2. Musical Background.....	2
2.1 Baroque Music.....	2
2.2 Variations as a Musical Style.....	3
2.3 George Phillip Telemann.....	4
3. Previous Work.....	6
3.1 Music and Computers.....	6
3.1.1 Analysis.....	6
3.1.2 Composition.....	9
3.1.3 Score Processing.....	12
3.2 Coding Schemes.....	13
3.2.1 Input Codes.....	14
3.2.2 Internal Representations.....	18
3.3 Application of Linguistic Theory to Music.....	24
3.3.1 Formal Grammars and Music.....	25
3.3.2 Musical Grammar Examples.....	28
3.4 Generative Modelling Systems in Music.....	32
4. Method and Implementation/Project Description.....	42
4.1 Introduction.....	42
4.1.1 The List as the Internal Representation.....	42
4.1.2 Transformational Grammar Approach.....	43
4.2 Input Codes.....	47
4.3 Method of Variation Development.....	51
4.3.1 Introduction.....	51
4.3.2 Labelling Melodic Sequences.....	54
4.3.3 Melodic Variation.....	57

4.3.3.1 Mirroring Sequences.....	57
4.3.3.2 Melodic Vocabulary Possibilities.....	58
4.3.3.3 Frequencies/Choosing Between Valid Melodic Vocabulary Entries.....	58
4.3.4 Rhythmic Variation.....	63
4.3.5 Checking Accidentals.....	67
4.4 The Database.....	70
4.4.1 Vocabulary Facts.....	70
4.4.2 Frequency Facts.....	71
4.4.3 Change Accidental Facts.....	72
4.4.4 Basic Musical Interval Facts.....	72
5. Results/Conclusions.....	74
5.1 Introduction: Syntax vs. Semantics.....	74
5.2 Results.....	75
5.3 Problems Encountered.....	80
5.3.1 Random Choice Efficiency.....	80
5.3.2 Multiple Viewpoints.....	83
5.4 Shortcomings and Future Directions.....	86
6. Appendix 1: Expert's Figures.....	91
7. Appendix 2: Movements Used For Variation Generation.....	96
8. Appendix 3: Program Listing.....	186
9. Bibliography.....	220

## 1. INTRODUCTION

Musical performance practice during the Baroque period (approximately 1600 - 1750) required that a solo performer be adept at adding variations on the composer's original melody during performance. Telemann, in his "Methodical Sonatas", provided, along with the original melody line of a solo instrument over a figured bass (see section 2.1), a written-out variation of that line. This offers an unusual opportunity both for students wishing to develop this skill and for musicologists studying the ornamentation style of the period.

This thesis describes an attempt to design an expert system that can model the human process of creating variations, using knowledge of the Baroque style in general and that gleaned from Telemann's variations in particular, to produce variations appropriate to the original melody line provided as input. This knowledge includes general common "rules" and rhythmic and melodic vocabularies as well as the harmonic framework of the specific piece. The expert whose behavior the system models is Ed Merck, whose thesis for his Master's Degree in Historical Musicology at Wesleyan University, Extemporaneous Variation in the Baroque: A Study of Telemann's Methodical Sonatas [Me2], involved learning from Telemann's sonatas the skills that allowed the creation of appropriate musical variations, both for the Methodical Sonatas analyzed and for other pieces of the period similar in style.

## 2. MUSICAL BACKGROUND

### 2.1 Baroque Music

The period of music history generally referred to as the Baroque era is typically dated 1600 to 1750 [Ap, p. 82]. During this period there was great emphasis on instrumental music, with such musical forms as the suite, the sonata, the concerto, and theme and variations being developed. A piece in sonata form was typically written in several movements, often for one solo instrument with basso continuo. The basso continuo part, or "figured bass", is a bass line which consists of the notes of the bass line and figures (numbers and accidental symbols). This line is "realized" by the keyboard accompanist, who fills in the harmonic aspects of the performance by adding chords according to the figured bass given (see Appendix 1).

As the use of the figured bass illustrates, the performer played a larger role in the Baroque era compositional process than in many other historical periods. As with the accompanist, the performer playing the melodic line above the figured bass was not provided with a score containing exact notes to be played. Ornamentation was not notated but rather it was expected that an accomplished musician would know how to add appropriate ornamentation to a written score.

"During this period [the 17th and 18th centuries] singers probably never executed a solo part as it was written. Corelli, Handel, Tartini, and their contemporaries made their written parts of

sonatas for strings mere sketches of what the player should do. ... Between 1650 and 1750 the practice of writing ornaments in notes was frowned on as detrimental to the visual clarity of the melodic line" [Ap, p. 630].

## 2.2 Variations as a Musical Style

According to the Harvard Dictionary of Music [Ap] there are five basic kinds of musical variations: 1) the harmony of the theme is preserved, while the melody is modified by adding ornamentation or other changes, but leaves its basic character intact; 2) the harmony is preserved while the new melody has no strict ties to the original; 3) the harmony changes as well, but the overall structure (number of sections, phrases, measures, cadences) remains the same; 4) even the structure differs from the original; 5) the melody remains unaltered while the harmonies change. Type 1 is typical of Baroque music (approximately 1600 - 1750), 2 of music of the Classical period (approximately 1750 - 1825), 3 of Romantic music (approximately 1825 - 1900) and 4 and 5 of contemporary musical variations.

Johann Quantz, in his Baroque treatise On Playing the Flute [Qu] discusses musical performance practice of the Baroque period. Reilly, in his examination of the writings of Quantz, explains the Baroque variation ideal:

"The principles involved in the addition of variations are quite simple. The essential notes of the melody must not be obscured, the notes that form the variation must be within the limits of acceptable embellishment



of the harmony of the principal note, and the variations themselves must reinforce the expressive character of the movement" [Re, p. 113].

These variations usually occurred in slow movements of pieces and mirrored the emphasis on sequence and imitation found in themes of the period.

Baroque variations, in their ornamented style, followed the practice discussed in the previous section. They were often not written out, but rather the performer would "improvise" on the melody while playing, much as a jazz musician improvises a solo based on a melodic, rhythmic and harmonic framework.

### 2.3 Georg Philipp Telemann

Georg Philipp Telemann (1681 - 1767) composed in the late Baroque German mixed style (common from 1700 to 1750). This mixed style combined the highly regular French style, which emphasized "essential" ornaments, and the Italian style, which encouraged more freedom in the variation (described as "optional modifications of principal intervals") [Me2]. His variations in the Methodical Sonatas can be labelled "extemporaneous variation" to suggest

"the freedom involved in the creation of these variations, capturing the contrast with the more rigid form of French ornamentation and the more general concept of improvisation involving the introduction of new material. Telemann's variations are best described as elaborations of pre-existing melodic designs" [Me2, p. 4].

These twelve "Methodical Sonatas" or Zwölf Methodische Sonaten were published in two sets, one in 1728 and the second in 1732, and were written for violin or transverse flute solo with keyboard accompaniment.

The sonatas were fairly typical examples of the solo sonata of the time but were unusual because Telemann notated example variations as well as the original melodies, perhaps as a learning aid to the amateur player attempting to refine his skills [Me2]. Though Telemann did not supply any kind of rules to produce these variations in addition to his examples, Quantz [Qu] does provide a general Baroque melodic vocabulary: a table, as he calls it, containing "the most common kinds of intervals, together with the basses appropriate to them and above the basses ... the harmony for them, so that the variations stemming naturally from these numbers or figures will be seen clearly in the following tables..." [Qu, p. 137]. The system described in this thesis uses the expert's attempt at building a Telemann vocabulary, using both the Telemann examples and the Quantz vocabulary as a model to help in generating variations.

### 3. PREVIOUS WORK

#### 3.1 Music and Computers

Computing has affected various areas of the field of music. In addition to the use of computers for sound production, they have been used extensively in music theory, both in analysis and composition, and in score processing and printing. The application of artificial intelligence concepts and techniques has spanned these areas both in allowing/providing for "intelligent" musical behavior by computers and for the analysis and development of theories of human musical skills and perceptions of both the composer and the listener.

##### 3.1.1 Analysis

In musical analysis, computer applications range from cataloging collections of a particular time period or composer to statistical analysis of the properties of a particular genre of pieces, to producing a grammar to describe a certain style of music. Examples of cataloging include work by Bernstein and Olive [BO], who designed a thematic concordance (cataloguing motives from throughout a piece) of the 16th century chanson. An initial attempt to build a thematic index (that is, an index which "identifies a composition according to the opening musical material of each piece" [BO, p. 154]), posed several problems because there were cases where 1) pieces essentially the same started differently, 2) pieces essentially different started the same, or 3) pieces had borrowed middle

material. An index did not address these relationships. It was possible to test the concordance concept using computers to address these issues, starting with about 300 pieces. Other examples of cataloging studies classify Ecuadorean Urban Music [BR] and Javanese music [Li1].

Once cataloging was done in each of these projects, the resultant database was then used for analysis of certain musical features. In some studies, "similarity metrics have been applied to ... databases to attempt to find evidence for time and place of origin, or sometimes the authorship of pieces whose attributes have been lost" [A2, p. 148]. In the three cases described above and in general, however, such computer analysis programs attempt to identify formal characteristics of pieces; to provide

"interval counts; citations and descriptions of such components as particular rhythmic patterns, specified intervals, or uses of accidentals; statistical counts of chord roots, inversions, or variants; and correlations of text accents with metrical or agogical accents, to name but a few of the many attempts to use the computer effectively as a tool for research" [Li3, p. 93].

In addition to the projects mentioned above, such analysis was done in studies by Nakamura and Inokuchi [NI] in determining the scale type of Japanese folk music, by Tsuboi and Ishizuka [TI] in analyzing these scales and their relationships to each other within pieces (through modulations), by Harris and Brinkman [HB] in analyzing set classes/ordered pitch collections of contemporary

music, by O'Maidin [Om] in finding melodic relations between two segments of Irish and Scottish jigs by inspecting melodic contour, metrical accentuation, durations and transpositions, by Stech [St1] in analyzing the pattern content of melodic lines, and by Longuet-Higgins and Steedman [LS] in developing programs to determine the time signature and key signature of the subject of a Bach fugue.

In determining key and time signatures of Bach "Well-tempered Clavier" pieces, Longuet-Higgins and Steedman use two sets of rules, one for meter and one for key. Given an input stream of a fugue subject, the rules narrow down and focus on the possible meter/key, eliminating possibilities that don't meet rule constraints and successively refining the "answer". In this way, they claim that they model the listener of such Bach pieces (rather than modelling the composer):

"We have, however, attempted to make our programs mirror the progressive character of musical comprehension - by which we mean that as a fugue subject proceeds the listener's ideas about its metre and key become more and more definite, and may indeed crystalize well before the end of a long subject. The progressive nature of the listener's comprehension is made explicit in an assumption about the permitted order of musical events in an acceptable melody. This assumption we call the 'rule of congruence', and it is fundamental to the operation of both our harmonic and our metrical rules" [LS, p. 223].

### 3.1.2 Composition

In contrast to this approach of using rules to model the listener's activities in perceiving performed music, many other researchers have used computer analysis to look at the composer's side of the process. Involvement of computers in the composition process has taken place on several levels. Determining grammars to describe a type of music and then using these grammars to allow computer generation of pieces in the analyzed style places much of the compositional process on the computer; this is the major topic of this thesis and will be examined in later sections. Included in the "computer-composed" category are the early systems such as the 1957 one which produced the "Illiac Suite for String Quartet" [HI], partially based on random generation with rules to approve/disapprove of the random selections. At the other end of the compositional spectrum measuring cooperation between human and computer are those systems being used solely for text processing (the human composer writes his piece on paper, then "types" it into the computer for its text processing functions).

Recently, interactive systems have been developed that provide more balance between the human and computer composers. One such system, the POD System of Interactive Composition Programs [Tr] allows the user to set the structure of a piece and to modify it in conjunction with computer feedback (sound synthesis).

"The interactive nature of the program lies in the user exercising control over various parts of the program to obtain successive modifications of the synthesized structure. That is, the user works on both the sonic and syntactic levels within the program, the principal task being to establish the relationship between the two. ... The semantic level of operation is that of the user evaluating interim results and modifying the strategy for obtaining a satisfactory goal structure" [Tr, p. 30].

By allowing the human composer to set parameters on the production of "pitch, timbre, vertical density, amplitude, and spatial distribution" [Pe2, p. 284] and to receive auditory feedback, POD helps the composer in an immediate way.

Another such tool, "the intelligent composer's assistant" [Ro3], has a balanced interactive mode as its primary goal, yet approaches it differently:

"The overall goal of an interactive composer's assistant is to support the composer in highly creative phases of composition. This includes the creation of the plan and architecture of a composition, and the encoding of musical material into the working score" [Ro3 p. 170].

Under this theory, the musical score is viewed as a series of objects (including procedures, data, or both) containing such information as pitches, durations, ornaments, phrases, etc., as well as other more abstract structural objects used for analysis. These objects may be manipulated by the composer through "hooks", either generated by the composer in set-up requests or by the system, using its knowledge of a concept and inference to apply that concept to applicable objects.

Roads has implemented a specialized composer's assistant system, Ios, for interactive orchestration of a score. Rules are given that specify some musical feature; the Ios system searches the score for instances of this feature and can then associate this feature with a certain instrument in the orchestrated score. For example, the flute may be assigned to the highest pitches of a phrase by a rule, or ornamentation may be added through the rules. A menu system is also provided as part of the user interface.

Cope has developed EMI (Experiments in Music Intelligence), a system in which the human composer regulates and directs the work of the computer, allowing it various amounts of power in the compositional process. Based on the use of identifiers - antecedent, consequent, ornament, and statement, and levels at which these identifiers are valid - foreground, middleground, or background, the composer sets up rules about which kinds of identifiers can follow which others. A dictionary listing acceptable intervals to "fill in" these identifiers is also available to the computer; by designing this dictionary with values specific to a particular style, the composer can lead the computer into keeping its generated compositions within that musical context.

The composer creates an initial motif (or instructs the computer to do this). "This becomes the principal germ of the work's phrase structure. Using a function that parses longer units from this central icon, the program then produces dictionary-correct variants" [Co1, p. 33]. In addition, throughout the



process the composer may use or ignore machine suggestions.

Flavors Band, a language by Fry [Fr2], helps the composer to create jazz and pop music pieces. Using menus as well as a text editor, the composer manipulates the tree-based musical structure representation (entire score, sections, parts, phrase processors) at the phrase processor level, which can effect low-level changes and, interconnected with other phrase processors, high-level modifications. A network of phrase processors makes up a score; all three types of phrase processors - note modifiers, control flow modifiers and accessors to event arrays (storage of the score produced) - can be used to build that score. In addition, at least theoretically, "event arrays can be used to store themes to be read by another phrase processor in order to generate a variation" [Fr2, p. 29]. Outside of this creation process, the composer can use the text editor and menus to modify the score, assign instrument parts, etc. without recomputing the whole score. Documentation, source code, and examples of phrase processor uses are also available through the menu system.

### 3.1.3 Score Processing

With the advent of more complex graphics capabilities, the processing of music text has seen major developments. Research has been done to provide better user interface tools for publishing/editing software. The HyperScore Toolkit [Po] contains programs that supply classes for score processing and other

user-interface tools (in the Smalltalk-80 Programming System environment) to

"provide comprehensive note- score-, and sound-processing functions and tools for the rapid development of new graphical or command-oriented user interfaces to music-processing software packages built in an extensible programming language and environment" [Po, p. 6].

Specific processing systems have included Professional Composer for the Apple Macintosh [Go] and Mockingbird [Go], where scores are entered at a Yamaha keyboard, the program gives corresponding noteheads, and the composer adds durations, barlines, etc., using a mouse for editing. The Finale system [To] by Coda Music Software, for home computer users, packages many music publishing possibilities into one software tool and is labelled by its designers as incorporating intelligent composer's assistant features as well. Manuscripts can be entered in several different modes to allow for flexibility, and a light pen and mouse make use easier. In addition to this textual processing, Finale has a hyperscribe function which notates music played on an attached MIDI keyboard. The transcription may be less than accurate for complex pieces. By giving the computer additional parameters such as the key, meter, where barlines fall, etc., the composer can interact with and provide the program with what it needs to improve the accuracy of its transcription.

### 3.2 Coding Schemes

In order to make use of computers in the field of music, certain computer design concepts must be addressed as they relate to music, each of which is necessary as a part of processing the syntactic and capturing the semantic aspects of a musical composition. These include the development of a group of symbols (code) in which to input the musical score, programming languages to manipulate musical processing, and data structures and representations to be used within the computer.

### 3.2.1 Input Codes

Input codes, systems representing the musical score and its symbols to allow machine processing, range widely depending on intended applications and approaches. Some systems use primarily numeric representations [Li4]. One Fortran representation used by Baroni employs a six digit number to represent each note; each field represents one specific musical feature of the note. Another numeric representation is that of Jackson and Bernzott, who code pitches by labelling them as a number that specifies where they fall on the piano keyboard, and represent rhythmic durations of these notes by columns in the representation.

What seems to be the more common approach, however, is a combination alphanumeric code, which can be easily understood and learned (often using mnemonic symbols) as well as easily manipulated both for efficiency and accuracy [St2, p. 287]. One such popular code is DARMS, Digital Alternate

Representation of Musical Scores, originally designed by Stefan Bauer-Mengelberg in the early 1960s in the arena of automated music printing (and originally called the Ford-Columbia Music Representation). In this code, a note is represented in terms of its pitch (a number denoting its position on the staff), and rhythm (a letter symbol representing duration). There are also other control codes to represent the positions of notes in the score and how one symbol is located in respect to another. These include instrument codes, space codes (above/below the staff), pseudo-space codes (for symbols not needing specific line/space placement), delimiters of horizontal/vertical movement (to the next note), skip codes (to leave space between notes), as well as representations for fingerings and other such notations [Er2]. There have been many applications based on DARMS; the Ecuadorean Urban Music cataloging project mentioned earlier is one system that uses DARMS as an input language.

Two types of the DARMS code exist, User-DARMS (also called Input-DARMS or DARMS) and Canonical-DARMS which is a version more efficient for machine processing. Currently a program to translate any Input-DARMS form to a standard Canonical-DARMS representation is being researched by Anthony Wolff and Bruce McLean, and an Input-DARMS syntax checker is also under development [Er2]. Interfaces between DARMS and other music data-processing systems such as IML/MIR have been discussed [Er2].

IML/MIR, Intermediary Music Language/Music Information

Retrieval, is a Princeton system similar to DARMS in that it encodes pitches by a number representing the line or space on the staff on which it resides but differing in other ways, including the representation of duration as a numeric code. It is an analysis-oriented software system used for projects such as the Mendel/Lockwood Josquin project to identify suspensions in cadences [A1].

CLML, Chicago Linear Music Language, is a variation on DARMS, designed for the specific cataloging project of the 16th century chansons at the University of Chicago mentioned earlier in this paper. CLML is actually a subset of DARMS in that it includes only the symbols that would be found in 16th century chanson music. "This simplifies the process of encoding by providing the encoder with a markedly uncluttered vocabulary" [BO, p. 156]. It also approaches the input in a linear way to match its application. Since each vocal part of a 16th century chanson is recorded separately on paper, this is the way the chanson is encoded in CLML (voice by voice) and the computer then aligns the voices given this code.

Other codes provide input mechanisms specific to a certain application. ANTOC, Alpha-Numerischer-Ton-Code, is used by Steinbeck [St2] to enter data appropriate and necessary for the analysis of German folksongs. The code is designed to match the specific nature of the application, and is used to encode the design structure of the piece as well as the individual note information. Information is stored in three sections, an identification section (to identify the

particular segment of the melody), a specification section (which uses single symbols to indicate such characteristics of that segment as its key, meter, mode, etc.), and a notational section (which contains symbols for the individual notes of the segment, including its pitch, octave, duration and position in the measure).

Brinkman [BR2] has designed a code that allows mathematical manipulation of pitch codes within the computer (as is commonly done in analysis programs for transposition, interval inversion, etc.) and yet also provides for the differences between enharmonic notes and intervals. He feels this is often lost in the original "pitch class" representation, which only provides information on the note's position on the staff - two different notes such as A-flat and G-sharp have the same position. Here, he invents the "name class" system, giving the spelling of the note on the staff, not just its position. Using both the pitch class and name class in pairs to represent a note or interval allows a clearer description of the actual note. A third term can be added to represent octave information, or such information can be mapped onto the two existing terms using mathematical formulas. He also discusses devising a code where this pair can be represented as a single integer.

Biles [Bi] has written a program to identify the performer of a jazz improvisation based on previously "heard" solos by that performer and others using an adaptive pattern matching algorithm based on weighted difference polynomials. In addition, the system attempts to generate solos in the style of a

given artist. A BNF grammar gives the rules for the input code, which can represent a single melodic line and an accompanying harmonic progression. Notes are represented by an alphabetic pitch followed by a duration (given as a fraction of the total measure), and special code (optional) to represent inflection and ties. A pitch's register is assumed to be closest to that of the last note's; this may be overridden by placing the octave number in front of the pitch code. The harmonic portion of the code, designating the root pitch and the type of chord, is entered by placing each chord in the input stream directly in front of the first note against which it is played. Time signatures are representable, but dynamics and tempo markings are not.

Input codes, therefore, range in generality and specificity, with the ability to adapt more general codes such as DARMS to specific applications and the potential to create new codes and new approaches to cater to some more unique applications.

### 3.2.2 Internal Representations

Whereas input codes represent the notation of a score, the data of a program, they do not represent the data in any semantically meaningful way.

"The core of the technical problem in each application (composition, performance, theory, and signal processing) is the development of the proper internal representations for the musical domain under consideration." [Ro3, p. 169]

Several programs have been written to transfer input codes to some functional representation useful for processing the semantic content of a musical composition by the computer rather than just stopping at the syntactic code. The SCORE system, for instance, is a Fortran program that takes a file of input codes and produces a list representation of the data, which encompasses parameters "dealing with the details of the synthesis technique used by the instrument which will receive the score" [Lo2, p. 258]. Different arrays are set up for pitches, rhythms, dynamics, etc.

"SCORE takes fields sequentially from each set of parallel arrays, translating them into note statements. It does this by traversing them in parallel, with an appropriate interpreter for each array type, generating one or more note statements for each index into the parallel arrays" [Lo2, p. 258].

Somewhat similarly in terms of building a representation from an input code, Alexander Brinkman of the Eastman School of Music has written a program (in Pascal) to function as a DARMS Scanner:

"The primary purpose of this program is to interpret the meaning of graphic symbols encoded in DARMS. We have bypassed Canonical DARMS, opting instead for direct translation of each encoded graphic symbol into representations that are more directly accessible for musical analysis. ... DARMS, in both its User and Canonical forms, encodes the graphic symbols used to represent music, not their meaning. It 'knows' what symbols are in the score and where they are, but not what they represent. In that sense,



uninterpreted DARMS is about as useful to  
an analysis program as the printed score  
is to a person who cannot read music"  
[Br1, p. 226].

The output of the Scanner program is a table of information about the score where the context dependency of the input DARMS code is eliminated by explicitly specifying temporal information of all parts (start time, stop time, duration), and linking this to information such as dynamics, articulation, etc. This output can then be used to build a data structure of the score using a doubly-linked ring structure, consisting of many circular lists [Br1, p. 251]. A "spine" list serves as a time line; "each spine node contains temporal information and link fields to connect it to notes in the part. Fields in each note node contain all other attributes of the note." [Br1, p. 251] In addition, there are start-time and stop-time links which tie together events beginning/ending at the same time. All of the links exist together within the structure;

"together, they allow an analysis program to move about the score in any manner desired, looking back or ahead at will, combining horizontal and vertical motion in any manner required. This makes it possible to evaluate context to a degree that is difficult to achieve when dealing with one-dimensional representations such as strings." [Br1, p. 251]

In Biles' jazz improvisation analysis/synthesis work, the program is given a solo (melody line and accompanying harmonic progression) coded in the input grammar, and translates this code into a multiple array representation which is

then used by the rest of the program functions. The first array represents the melodic notes of the solo: its register, pitch within that register, duration, a pointer (offset) to the chord that is played at the same time, and "special information" such as ties and stress figures. The second array holds the chordal information about the piece: the measure number in which the chord occurred, the root pitch of the chord, the intervals describing the type of chord, and a pointer to the first note in the note array with which this chord is played. The third array holds measure pointers, pointing to the first note (in the note array) of each measure, while the fourth array holds information about potential key/tone centers: what the key is and at what chord in the chord array the key "window" begins.

Myers [My], (based on Smoliar [Sm2]), in representing transformations taken in a Schenkerian analysis system (described in more detail later), uses a tree structure to describe a musical event. An event can be a single note, a sequence of other events occurring in some specified order, or a simultaneity of events all beginning at the same time. This tree can also be represented as a parenthesized list; after applying one of a number of functions to the list (tree) structure, the resultant list is output by the computer. The program creates the original tree by reading an input ASCII file, and its write function outputs the tree in the list notation.

In addition to these specific "low-level" representation options, several

alternative design approaches provide possibilities to guide the design of a representation. Levitt [Le1, 2] has chosen to use constraint languages to implement his expert system to generate jazz improvisations. In constraint systems, instead of the conventional approach of writing algorithms with specific procedures to solve a new problem, each built-in constraint functions as a part of a network. This network of constraints forms a model that identifies the structure which is valid in a particular domain. Each constraint may have associated with it several procedures, which can be connected in different ways to model particular situations. New domain information is added by using links to allow built-in procedures to be used in new ways. Given a constraint model, the program can "fill in the blanks" from a partial description of an object [Ro3, p. 177]. Levitt's program

"negotiates mutual harmonic, melodic, and thematic constraints to produce a solo from a chord progression and melody. The constraint system I developed employed roughly 30 musical terms, including the most basic music theory concepts and the kinds of constraints I think I negotiate when I improvise on a keyboard" [Le2, p. 20].

In addition to constraint representations, several other design alternatives exist. Many knowledge-based systems track the semantics of a piece by allowing for multiple perspectives of the same piece or object. In his expert system CHORAL, which takes as input a melodic (soprano) line of a Bach chorale and generates an appropriate harmonization (that is, generates the other three voice

parts - alto, tenor, and bass, typically found in a chorale of that period), Ebcioglu [Eb1] uses multiple viewpoints, processing/analyzing a chorale in the chord skeleton, the fill-in, the melodic string, the merged melodic string, the time-slice, and the Schenkerian analysis views. These work in tandem to gradually, step-by-step, generate the chorale harmonization. Heuristics and constraints fit into each of these views to tailor selections made during the generation process.

Design alternatives such as semantic networks and frames also allow for multiple viewpoints or representations. Semantic networks consist of nodes whose relationships to each other are shown via labelled links such as is-a, consists-of, etc.

"In semantic networks, some forms of deduction are trivial; for example, with a score encoded as a semantic net, we can simply follow is-a and consists-of nodes to deduce the properties of notes and the contents of note groups" [Ro4, p. 27].

Levitt's jazz improvisation program takes two semantic networks as input; the first provides a general description of a piece while the second is a specific realization of the concept described by the general semantic network. The program draws relationships between the two. Then, when given a second description (network), the program generates a realization based on the first pair. This framework is used to take samples of the New Orleans jazz style and "learn" larger concepts to be applied in its own improvisations.

Frames also provide models for grouping information into larger concepts.

"Frames can be thought of as structured semantic nets which supply a default context for otherwise isolated facts and procedures." [Ro4, p. 28] Each frame has "slots" to hold these facts and procedures; each slot can have a default value as well as a range of acceptable values to take. Balaban, in her system to study Western Tonal Music [Ba1-6] defines a representation based on frames and logic called "The Generalized-Concept Model":

"The Generalized-Concept (G-C) formalism is a hybrid logic based formalism, that associates objects in the subject domain with their descriptions. The chunking of the knowledge gives all the good properties of frames systems, while retaining the simplicity and rigorousness of logic" [Ba3, p. 1].

Unlike more typical frame representations, Balaban's system does not have built-in inheritance; rather, inheritance is explicitly specified. The logic elements provide the inferencing mechanisms and assertational qualities, while the frame model provides a natural means of describing an information structure.

### 3.3 Application of Linguistic Theory to Music

Similarities between speech and music exist at many levels, including perception, analysis, and use of written symbolic representations. The application of linguistic theories to music, then, seems natural.

"Both in its written notation and in analytic listening, music involves the recognition and use of symbols, symbolic structures, representations, and transformations. Listening

to music involves aural perception, just as speech does. On the other hand, the higher level perceptual mechanisms of music and language are apparently quite different. ... A number of researchers have concluded that the subject of musical perception complements speech perception, and that an understanding of both is essential to a full understanding of human symbol-using behavior" [Ab2, p. 148].

Rather than elaborate on many of the possible similarities, however, we shall focus on the theory that both language and music have a clear structural basis and that computational models found worthwhile to model language structures may be applicable to tonal music, particularly those models called "grammars".

"The usual method of traditional music theory is to show how the most elementary of musical patterns, in which 'grammatical' relations are properly completed, can be extended by logical complications. We find that constituent structure grammars are adequate to embody the knowledge expressed by traditional musical theory, just as they are adequate to embody traditional knowledge about speech" [LG, p. 140].

### 3.3.1 Formal Grammars and Music

A formal grammar describes the structure of a language. It lists the terminals of the language (those "words" that are used in the language) as well as the non-terminals, which are used to describe valid combinations of the terminals in the form of rules ("productions"), where non-terminals are replaced by

terminals and/or other non-terminals.

Grammars have typically been classified into types, several of which are discussed by Roads [Ro2] in terms of their suitability for musical applications. A type 0 (free) grammar includes no restrictions on productions; therefore, a string can be lengthened or shortened through the application of a rule. Infinite strings and null productions are also allowed. Musically these constructs are not applicable since practically there can be neither null or infinite compositions.

Type 1 (context-sensitive) grammars have productions such as  $AXB \rightarrow AYB$ , where strings are allowed on both sides of the rule, but Y must be longer than X (that is strings will be made longer by applying this type of production). These seem to be appropriate to describe musical structures but the fact that they allow ambiguity, are difficult to trace, and would possibly take a huge number of productions to describe all possible contexts of musical situations make them more appropriate in restricted forms. One example of such a form would be using only context-free rules but adding control procedures to simulate context-sensitivity by regulating which of these rules is applicable at which stage of the analysis.

Type 2 (context-free) grammars further restrict the rule format, allowing only one left-hand-side non-terminal to be expanded at a time.

"The power of type 2 grammars for music lies in their ability to represent multi-leveled syntactic formations, since any non-terminal (representing a macrostructure category

like motive, phrase, sentence, section, or movement, and the like) may generate a string of tokens at a lower level" [Ro2, p. 50].

A rule such as  $A \rightarrow XAB$  allows nesting or "self-embedding" of material, a particularly common and natural structure in music.

Type 3 (finite-state) grammars allow no more than one non-terminal on each side of a production. Rules such as  $A \rightarrow a$ ,  $A \rightarrow aB$  (where capital letters are terminals and lowercase are non-terminals) are valid. This kind of production does not allow nested motives or phrases but rather leads to a more linear sequence of processes/development.

A regulated grammar takes a type 2 context-free grammar and adds control procedures to determine when certain rules should be applied (such as resolving ambiguity, controlling recursion, etc.). Alternatively, a control language may be used to regulate the grammar by labelling each production; its expressions use these labels to choose various rules at different stages of parsing.

Lastly, transformational grammars have been designed specifically to handle the structure of natural languages. There are two parts of the grammar: a "phrase-structure grammar", which generates abstract sentences, and a set of rules, which transform the abstract sentences to sentences in the natural language (English). In addition, in the case of spoken languages, morphonetic rules are used to map the natural language sentences to utterances used in spoken sentences. As the two-phase concept is applied to a musical grammar, one can follow the same path of taking a general "phrase-structure"-like grammar to



describe/generate the framework of a piece without specifics being given. A set of rules then adds the specifics, transforming the abstract to an actual "fleshed-out" piece.

### 3.3.2 Musical Grammar Examples

Some of the above grammar types have been used for specific musical applications. The early computer composition work of Xenakis is based on the use of rules of stochastic composition, using Markov chains "to generate strings of sound textures where each texture is in turn described by probability distributions" [Ho2, p. 51] . These Markov chain processes relate to the type 3 (finite-state) grammar model, where each rule changes one non-terminal and so moves in a step-like manner. Unlike strictly random processes, however, a Markov chain "takes into account the context of an event in a sequence making the probability of its occurrence depend on the event that preceeded it" [Jo, p. 46].

Biles [Bi] also uses Markov chains in the generation part of his jazz improvisation analysis/synthesis program. Existing solos of a particular jazz soloist are analyzed, and Markov chains are built for the notes of the piece. Using tonal windows (viewing a section of the piece as being based on a particular chordal group) to provide the "deep structure" of the piece, notes are selected from the Markov chains and transposed into a particular tonal center,

providing the "surface structure" of the improvisation.

After discussing the limitations of type 2 and 3 grammars, Moorer [Mo] extends the Markov chain approach by adding heuristics that he claims allow for more context-sensitivity. The structure of the piece is set by the operator, who chooses the number of "major" groups, the number of "minor" groups within these, and the number of beats per minor group, as well as setting probabilities to guide the system in the next steps. Rhythms are added randomly; chords are chosen using probabilities and are then passed through a heuristically based filter which eliminates dull and/or long, "unmusical" chord groups. Finally, the melody notes are chosen using probabilities, which dictate whether certain melodic figures will be copied or new, and direct the derivation of new figures.

Winograd [Wi] draws on the regulated grammar approach in his program to analyze the harmonic structure of pieces. Using what he terms a "systemic grammar" he ties syntax to semantics by using heuristics to guide which syntactic parsings have the most semantic meaning at any given point.

"In Winograd's Lisp program, semantic procedures were used to guide the parsing of the music. This enabled the analysis program to avoid many ungrammatical parsing paths; it also eliminated ambiguous paths that might be grammatical but not very meaningful in terms of an overall analysis of a piece" [Ro1, p. 16].

Also trying to deal at the semantic level for analysis, Narmour developed an "implication-realization model", an expectation-based system using basic

semantic primitives from which all musical structures are built. Based on the natural language processing theory of Conceptual Dependency by Schank, this approach uses rules to model the natural relationships between primitives, goals, and contexts in a musical setting rather than rules that focus on syntax.

The transformational grammar approach views the derivational process as a relationship between a general/deep level (the musical structure) and a more detailed/surface one (the musical piece). This has similarities to transformational linguistic approaches:

"a parallelism exists between the Ursatz [defined below] in tonal music and deep structure in linguistics. Likewise one finds an analogy between the transformational derivation of the surface structure of a natural language from the base and the generation of a musical composition through successive prolongations of the Ursatz" [FR, p. 22].

This quote introduces us to terms used in the most well-known of the transformational musical approaches, that of Heinrich Schenker. His system, used in analysis of Western music, identifies three structural levels in a composition, the foreground (surface structure), the middleground, and the background (Ursatz). Groups of notes at one level can be reduced to single notes at a higher level moving toward the Ursatz, which is a two-voice motive based on some version of the tonic triad [Fo]. A set of functions can provide the means for moving in the other direction, from the Ursatz to a composition, thus approaching a transformational grammar.

"The foundation of Schenker's theories was based on the postulation of the existence of musical proto-structures which could expand into tonal compositions through a well-defined set of rewriting rules. ... According to Schenker's theories, an entire tonal composition could arise from a series of such elaborations (which he called diminutions), compounded ultimately upon a single note - the "tone" of the composition's tonality" [Sm2, p. 42].

Smoliar [Sm2] has formalized and Myers [My] has implemented a system based on Schenkerian analysis in which a theorist can interactively generate musical strings from a base (like Schenker's *Ursatz*) by using various commands that apply rewrite rules to produce a new level structure (surface) from the old. As discussed earlier, the system uses a tree to represent the musical structure, starting at the root and adding/deleting branches, filling in and rearranging notes.

Lerdahl and Jackendoff [LJ2] have devised a transformational grammar approach with similarities to Schenker's in its melodic formalisms, but they also incorporate rhythmic transformations. The system uses four interconnected hierarchies, each with particular structures and possibilities/constraints for elaboration and yet all working together to form one musical piece. Two of the hierarchies address rhythmic qualities (grouping, meter) while the other two focus on pitches (time span, prolongational reduction).

"The close dependence of prolongational reduction on time-span reduction constitutes a major claim of our theory. It asserts that the perceived patterns of tension and relaxation in pitch structure depend

crucially in the hierarchy of structurally important events within time-spans as defined by meter and grouping. In other words, the listener's understanding of pitch connections in a piece is a function of how he segments its surface. This claim entails the unification of pitch and rhythm within one overarching theory" [Ch].

There are two kinds of rules used by the system to analyze the structure of these components in a specific piece of music: well-formedness rules define constraints on relationships between parts of the hierarchies, and preference rules choose appropriate well-formedness rules for each hierarchy from those possible. Here one sees aspects of other grammar types imposed upon the transformational model.

### 3.4 Generative Modelling Systems in Music

A large number of composition applications using transformational grammars build on the grammar as an analysis tool. To model some compositional style (of a certain period, composer, etc.), analysis is done to define a grammar from existing works of that style and then used to generate new works in that same style.

Rader [Ra1] has written a program to generate rounds that first generates a harmonic framework, which it then uses to create a melody that will function successfully as a round. In each of these generations, productions used to choose the next chord/note are controlled by applicability rules, which define when

productions may or may not be used, and weight rules, which assign weights to applicable productions to decide which production will actually be applied.

In proposing a generative grammar of Eskimo songs, Pelinski [Pe1] analyzed syntactic structures, both those identified by the performers themselves and those determined by observation, and builds a grammar based on the proper organization of these sections, which he later uses to produce Eskimo songs of that particular style.

Other system developers concentrate on the style of a certain time period rather than on a folk style like rounds or Eskimo songs. Lidov and Gabura [LG] designed a generative program based on a grammar of common practice period art music (around the time of Haydn - late 1700's). They limit their work to an eight-bar phrase in two-four time, and generate pieces within this framework. First a "rhythmic base" is generated by a Rhythm Grammar, giving a string of durations labelled according to tonal stress, using lower numbers for more stressed notes. Using this duration/stress string, a Pitch Grammar assigns actual pitches to these notes. This is done starting with the most stressed notes (at the root of the tree) and continuing to scan the string until all levels/notes are processed. This grammar shows similarities to the Schenkerian transformational model in its hierarchical tree with different structural levels. In addition, once the string is produced, a filter is used as an analysis tool to evaluate the melody for melodic contours.

Rothgeb [Ro5] has approached the Baroque period and its use of the figured and unfigured bass in keyboard accompaniment. He has written a system that realizes an unfigured bass following two general steps, inferring the appropriate figured bass from the unfigured one and then realizing that figured bass. His program uses constants and variables to represent intervals and individual bass notes, and uses predicates and functions to process these terms. Predicates take individual bass notes (in the DARMS code) and return truth values determining applicability of rules due to the bass progression and figures already assigned; the functions are used to move back and forth within the bass line, taking bass notes as arguments and returning other bass notes.

Several other systems have been developed that explore Baroque style; MELOS2 [BB] examines the structure of Bach's Lutheran chorales using two kinds of rules, macroformal, which deal with the entire chorale, and microformal, which approach the individual phrase level. The overall approach parallels Schenkerian analysis:

"The main hypothesis is that every phrase of a chorale is an ordered entity based on several levels of a hierarchy; certain notes, or rather certain features of these notes, are seen as the structural backbone, while others are seen as secondary elements and decorations. The phrase is therefore thought of as an organism that derives from an initial kernel and enriches itself by expansion in successive structural phrases"  
[BB, p. 206].

The macroformal rules determine a general framework, organizing the individual phrases into a whole, unified piece by using periods to group phrases and analogy rules to ensure continuity between phrases. Within the limits imposed by the macroformal rules, the microformal rules produce an individual phrase, first by taking a kernel (like Schenker's *Ursatz*) and developing a primitive phrase by roughly filling in intervals within it using a set of development figures. The program then transforms the primitive phrase to the final melody by adding actual accents, durations, pitches, tonality, and other embellishments.

The HARMONY program [BB] looks at the lower three voices (not the melody) of the first phrase of Bach chorales. They designed a harmonic framework for a chorale phrase by identifying a hierarchy in the chordal structure. The harmonic kernel is the first and last chords, usually tonic triads. The next (second) level adds those chords with a tonic function that are not in the first/last positions but add to the kernel framework. Next the chords that connect the tonic chords are added, then those that connect the tonic chords of levels one and two to the broad connectors of level three. Finally, the remaining filling chords are added. The MELOS2 grammar can be used in conjunction with this grid to generate a soprano line and, when modified slightly, to generate a bass line to fit into the harmonic structure.

In further work on Baroque style and a grammar for the melody line of



Bach chorales, Baroni and Jacobini began research [BJ2] on the first two phrases of major mode chorales in four-four time. For analysis, they separate the examples into two phrases, and divide the structure of each phrase into several fields: the beginning, central body, and cadence. Various rules govern the possible transitions in these fields, the general configuration of the phrase, and the structural relationships between the phrases. The program generates melodic phrases in the following way:

```
/* generate acceptable piece (phrases and connections between phrases) */
repeat
  /* provide overall structure */
  generate durations of the phrases

  /* generate all phrases */
  repeat
    /* generate one acceptable phrase */
    repeat
      choose the first note
      continue note by note through each field
        following transition rules
      choose a cadence
    until phrase meets global constraint rules

  until there are no more phrases to generate

until pass as acceptable through filter governing connections
  between phrases
```

In a revision of this first program [BJ3], Baroni and Jacobini analyze the entire chorale (not just the first two phrases), and make a substantial change in

methodology. Instead of using the local and global rules separately as in the old version, now the local and global rules act in tandem,

"leaving aside a sequential structure in favor of a transformational structure of grammar able to pick out, right from the beginning of generation, the essential elements which make an organic whole of the phrase and able to gradually develop these elements, keeping the sense of overall unity of the phrase" [BJ3, p. 2].

Given the number of periods (phrases in which the final note is the tonic), levels (phrases that make up periods, in which the final note is non-tonic), and durations of phrases, an analogy tree is built, which will provide the framework of the chorale as a whole and address how the phrases will relate to each other. The tree has several levels (based on the MELOS2 analysis): the kernel as the root, then the primitive phrase, modified phrase, adapted phrase and finally, at the leaves of the tree, the surface phrase.

Relationships are formed between nodes of the tree in the following way: starting at the surface phrase level, groups of "sister" nodes (phrases of the same length) are randomly chosen and linked to a parent node on the next highest tree level (the adapted level in this case). Nodes not in a sister group are matched one-on-one with unmatched parent nodes. Next, sister groups of adapted phrases are chosen and linked to a parent modified phrase node, etc. Once this analogy tree is finished, the actual phrases represented in the tree can be generated.

As in MELOS2, the primitive phrase is generated by filling in the kernel

scale between notes of the kernel. Then, following "rules of insertion" a modified phrase is generated; these rules govern such generations by inserting repetitions, appoggiatura, neighbor tones, and skips within certain applicable contexts. The modified phrase is converted to the adapted phrase through rules of duration, which take strong and weak beats yielded from the insertion steps and map them to actual rhythms. A surface phrase is achieved by "transformations", which assign actual tonal features to notes through modification (cutting, inversion), adaptation (giving precise pitches/tonality, checking cadences), and fioritura (adding ornamentation) to the adapted phrase. Relationships between phrases are ensured by the analogy tree because phrases with the same ancestors have the same characteristics to a certain point in the tree, guaranteeing continuity, but then branch at some point to add diversity.

In the CHORAL expert system [Eb1] discussed earlier, which harmonizes a Bach chorale using multiple viewpoints, each process (viewpoint) has its own array to hold notes chosen for the chorale (one note per array element). During execution, all processes are blocked except one, which updates its own array as much as possible using information from its own array, input, and the arrays of other viewpoints. It then schedules the next process; the chord skeleton view acts as a clock, which executes one step when it is scheduled and moves all other processes on in the generation process. Each viewpoint has three cooperating parts to use in executing its generate-and-test strategy. Production rules generate

possible notes to be assigned to the current solution array element, constraints reject certain potential assignments, and heuristics lead the program to try some assignments before others. If no acceptable candidates to be added to the array are found, the program backtracks (using a stack) to a previous stage and continues from there.

In addition to generating the harmonization, Ebcioglu's program outputs a Schenkerian-style analysis for the melody and bass lines. The hierarchical tree is built by a parsing algorithm, which reduces the melodic line to the Schenkerian descending linear progressions based on the *Ursatz*. This is done using a stack; if the current input pitch does not continue linearly from the previous pitch (that is, it jumps), the parser pushes down the current state on the stack and enters a new state. Later, when that same progression continues in linear fashion (after having been stored on the stack), the most recent notes are output as nodes of the parse tree, groupings (slurs) are made to show any closure of a progression, and the stack state is restored.

In addressing a situation somewhat similar to improvising Baroque variations, systems have been written to generate jazz improvisations. In addition to the Biles program (using Markov chains) described earlier [Bi], Ulrich has done work in analysis and synthesis (not based on a grammatical approach) of a set of jazz variations on a melody. The system described [Ul] analyzes the melodic material and its relationship to the harmonic background by identifying

chords likely to accompany melodic notes, grouping these notes/chords together to find key centers, assigning functional descriptions to these chords within these keys, and finally determining appropriate scales to describe how the melodic theme fits into the harmonic background. From here, the system uses the framework that it has set up for itself to place melodic material based on motives and scales that were analyzed into the new variation, within the given harmonic structure.

Levitt, like Ulrich, has designed a system based on two components, analysis and synthesis, and is not grammar-based [Le1]. Unlike Ulrich's "key" system, however, Levitt's analysis program infers a more flexible framework of modes. Levitt's variations also have less strict ties to the original melodic line. Using a system of inherited patterns and defaults, themes from the original melody are examined for a variety of melodic features. Original phrases are ranked to encourage use of those with the most interesting material. A variation of that phrase section is generated according to the feature identified. For example, a leap of greater than a fifth in the input phrase will be modelled/inherited by the new variation but may leap in the opposite direction. Once this phrase is completed, the next part of the variation may be based on the next most interesting phrase (as ranked during the analysis section) or on the variation phrase just generated.

Once these patterns have given some direction to the new phrases (along

with the boundaries provided by harmonic and interval description constraints), default information is used to narrow down the set of possible actual pitches that satisfy the constraints to a single one.

"This procedure whittles away the options, simultaneously avoiding large leaps, non-modal tones, and local repetitions - as long as this is consistent with the given descriptions" [Le1, p. 30].

Attention is also given to movement toward a harmonically important pitch within sub-sections of the piece using "trajectories", which point to notes to "aim for", adding another level of musical structure to the variation.

## 4. METHOD AND IMPLEMENTATION/ PROJECT DESCRIPTION

### 4.1 Introduction

#### 4.1.1 The List As the Internal Representation

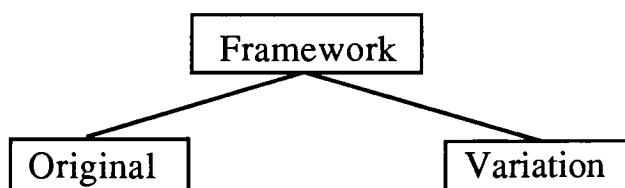
The basic representation used by this program in generating a variation on a given musical piece is the list structure. At all stages of the variation development process the musical piece is represented as a nested list. Each melodic note of the original piece is represented as a list of three lists (the first holds the chord supporting that note, the second the rhythmic value of the note, and the third the pitch), to be discussed in more detail in section 4.2. The whole piece, then, consists of a list of these sublists. As the original list goes through the transformation process, various configurations of sublists are used to represent the piece.

The basic approach used to analyze the current list and build a new list that is farther along in the transformation process is recursion, as supported by the Prolog language and the package, Advanced AI Systems' Prolog, version M-2.0. Each time the list is "changed" to a new list, it is done by matching the first (head) element (a single note or some grouping of notes depending on the stage in the development) to part of a goal. Using this goal, this head element (either unchanged or transformed) is placed on the new list and the rest of the original list is then processed, using the same goal matching sequence (recursion). Not until the end of the processing of a list (piece) is the new list really formed (as we come out of the recursive calls).

This list/sublist internal representation differs from the approaches taken by Brinkman and Biles, in which the different characteristics of notes were represented separately and linked together through a central structure. Whereas the use of these structures (Brinkman's spine list and Biles' multiple array representations) makes the viewing of the piece (traversing these structures) a different kind of activity from altering the piece, the list approach allows for both the viewing and manipulation of the structure to be done by a matching/substitution process, supporting a more transformational approach.

#### 4.1.2 Transformational Grammar Approach

As discussed in section 3.3.2 the transformational music approaches of Schenker, Lerdahl and Jackendoff, and Baroni and Jacobini are based on the notion that a musical piece can be analyzed at several levels. Like the work of Smoliar and Myers, the underlying model used in this analysis system is the tree structure. The transformational nature of the process of converting the original and framework representations (lists) into a variation list would fit on a transformational tree as follows:





The "original" input list contains every note of the piece, while in the framework passing and other augmentative notes are eliminated; what remains is a piece which represents an underlying musical structure of the original. One can see this in the musical notation of the example which will be used throughout this chapter. It is taken from the opening section of the composition "Sonata in d<sup>m</sup> for Recorder and Continuo", Opus 1 #2 adagio movement, by Loeillet (Loeillet\_d<sup>m</sup>). Original and framework versions (transposed to C as is done for input to the program) are notated as follows:

Original *am* *g#dim* *am* F *d#dim* E *am*

Framework

*dm* *G7* *C* F *Bdim* E

*am* E *am*

In other words, given the original, the framework would be one level up (toward the Ursatz in the Schenkerian model or the kernel in the Baroni Jacobini model). Using this higher base level we can generate a variation, another possible surface level of the tree. Going up more levels from the framework and generating from there would allow more diversity in the variation. As is done by the expert at times, more abstract choices of voicing possibilities (chordal tones other than those derived in the framework level from the original) could be chosen from such a higher level. Currently this program makes changes in the framework voicing possibilities only if the melodic vocabulary makes a change during its substitution. It only works at the most "leafy" layers of the tree and does not address the deepest levels.

In the Schenkerian model, transformations of the framework to a variation would not directly see the original leaf list. In this multiple viewpoints version (see section 4.3.3), however, the original is "peeked at" for guidance while the transformation of the framework is occurring. Since the framework is not generated by the program but is input along with the original, some analysis of the original during the "phrase to phrase" melodic (and rhythmic) transformational process is used. In addition, unlike the strict Schenkerian systems, rhythmic substitutions are also explored, though they are strictly leaf to leaf substitutions. This is similar to but less involved than the work of Lerdahl and Jackendoff or Lidov and Gabura. Lerdahl and Jackendoff identify the

underlying rhythmic structure in the transformation process, giving equal importance to the rhythmic and melodic aspects of the transformation. Lidov and Gabura actually begin their generation process with a rhythmic structure and then add pitches to the notes. It should be noted that Lidov and Gabura's work did not deal with a generation from an existing structure - variation - but rather a generation from "scratch" in a certain style.

#### 4.2 Input Codes

As explained above, the list representation of a piece consists of a sublist for each note in the score. For each note there are three sublists, one each to hold the chord played with that note, the rhythmic value of the note, and the note's pitch. As in the DARMS input code, pitches are represented by their letter name and octave. Rhythmic information about each note is separated from the pitch information to support this representation, however. Like the Biles input system chordal information is entered in the input stream, associated with the notes supported by that chord.

Pitch letter names must be lowercase, sharps add an 's' and flats an 'f' to the pitch lettername. For example: c, cs, d, bf, b. Octave information follows the DARMS practice of labelling the notes on the piano keyboard the notes between middle C and the b above that are octave 0, the next c to b make up

octave 1 and so on. Octaves below middle C would be -1, -2 and so on. Pitches in octave 0 need not notate the 0 as part of their pitch code; other octaves are represented by adjoining the octave number to the pitch letter. For example: c, d, b, c1, g1, bf1, c2, fs2, g2. Trills are notated using the symbol "tr" as a separate pitch, added to the note group after the pitch which should be trilled.

The melodic pitches found in the original melodies lie between middle c (c) and g2. All original melodies have been transposed to the key of C.

Similar to the Biles' notational system, chords are represented in lower case letters (g, c, fs), major chords by the letter alone and minor chords with an 'm' following the root letter (dm, am), diminished chords with 'dim' following the root letter (bdim, fsdim), seventh chords with a '7' after the letters (g7, dm7), and major 7th chords with a 'maj7' after letters (cmaj7).

Rhythmic codes are as follows:

w	whole note
h	half note
q	quarter note
e	eighth note
s	sixteenth note
th	thirty-second note

sf	sixty-fourth note
t	eighth triplet (3 eighth triplets take 1 beat)
st	sixteenth triplet (3 = 1/2 beat)

Prefacing these codes with a 'd' signifies a dotted note value (dh = dotted half note); a dotted note takes the number of beats of its base rhythm and adds on 1/2 of that amount. A 't' preface means that the rhythm is tied to the previous rhythm.

Input code for the Original and Framework versions of the Loeillet example which was examined earlier are given below:

<u>Original</u>		<u>Framework</u>
	↓	
[[[am],[q],[e1]], [[am],[e],[c1]], [[am],[e],[a]],		[[[am],[q],[e1]], [[am],[e],[c1]], [[am],[e],[a]],
[[gsdim],[q],[f1]], [[gsdim],[te],[f1]], [[gsdim],[s],[e1]],		[[[gsdim],[q],[f1]], [[gsdim],[te],[f1]],
[[gsdim],[s],[d1]],		[[gsdim],[e],[d1]],
[[am],[e],[c1]], [[am],[e],[e1]], [[am],[e],[a]], [[am],[e],[b]],		[[am],[q],[c1]], [[am],[e],[a]], [[am],[e],[b]],
[[f],[e],[c1]], [[f],[e],[e1]], [[dsdim],[q],[ds1]],		[[f],[q],[c1]], [[dsdim],[q],[ds1]],
[[e],[dq],[e1]], [[e],[e],[e1]],		[[e],[dq],[e1]], [[e],[e],[e1]],
[[am],[e],[c1]], [[am],[e],[e1]], [[am],[e],[a]], [[am],[e],[c1]],		[[am],[q],[c1]], [[am],[q],[a]],
[[dm],[q],[f1]], [[dm],[ts],[f1]], [[dm],[s],[e1]], [[dm],[s],[d1]],		[[dm],[q],[f1]], [[dm],[te],[f1]], [[dm],[e],[d1]],
[[dm],[s],[c1]],		
[[g7],[e],[b]], [[g7],[e],[d1]], [[g7],[e],[g]], [[g7],[e],[b]],		[[g7],[q],[b]], [[g7],[q],[g]],
[[c],[q],[e1]], [[c],[ts],[e1]], [[c],[s],[d1]], [[c],[s],[c1]],		[[c],[q],[e1]], [[c],[te],[e1]], [[c],[e],[c1]],
[[c],[s],[b]],		
[[f],[e],[a]], [[f],[e],[c1]], [[f],[e],[f]], [[f],[e],[a]],		[[f],[q],[a]], [[f],[q],[f]],
[[bdim],[q],[d1]], [[bdim],[ts],[d1]], [[bdim],[s],[c1]],		[[bdim],[q],[d1]], [[bdim],[te],[d1]], [[bdim],[e],[b]],
[[bdim],[s],[b]], [[bdim],[s],[a]],		
[[e],[e],[gs]], [[e],[e],[b]], [[e],[e],[e]], [[e],[e],[e1]],		[[e],[q],[gs]], [[e],[q],[e]],
[[am],[q],[c1]], [[am],[te],[c1]], [[am],[s],[b]], [[am],[s],[a]],		[[am],[q],[c1]], [[am],[te],[c1]], [[am],[s],[b]], [[am],[s],[a]],
[[e],[e],[gs]], [[e],[s],[a]], [[e],[s],[b]],		[[e],[e],[gs]], [[e],[s],[a]], [[e],[s],[b]],
[[e],[de],[b]], [[e],[s],[a]],		[[e],[de],[b]], [[e],[s],[a]],
[[am],[dq],[a]].		[[am],[dq],[a]].

## 4.3 Method of Variation Development

### 4.3.1 Introduction

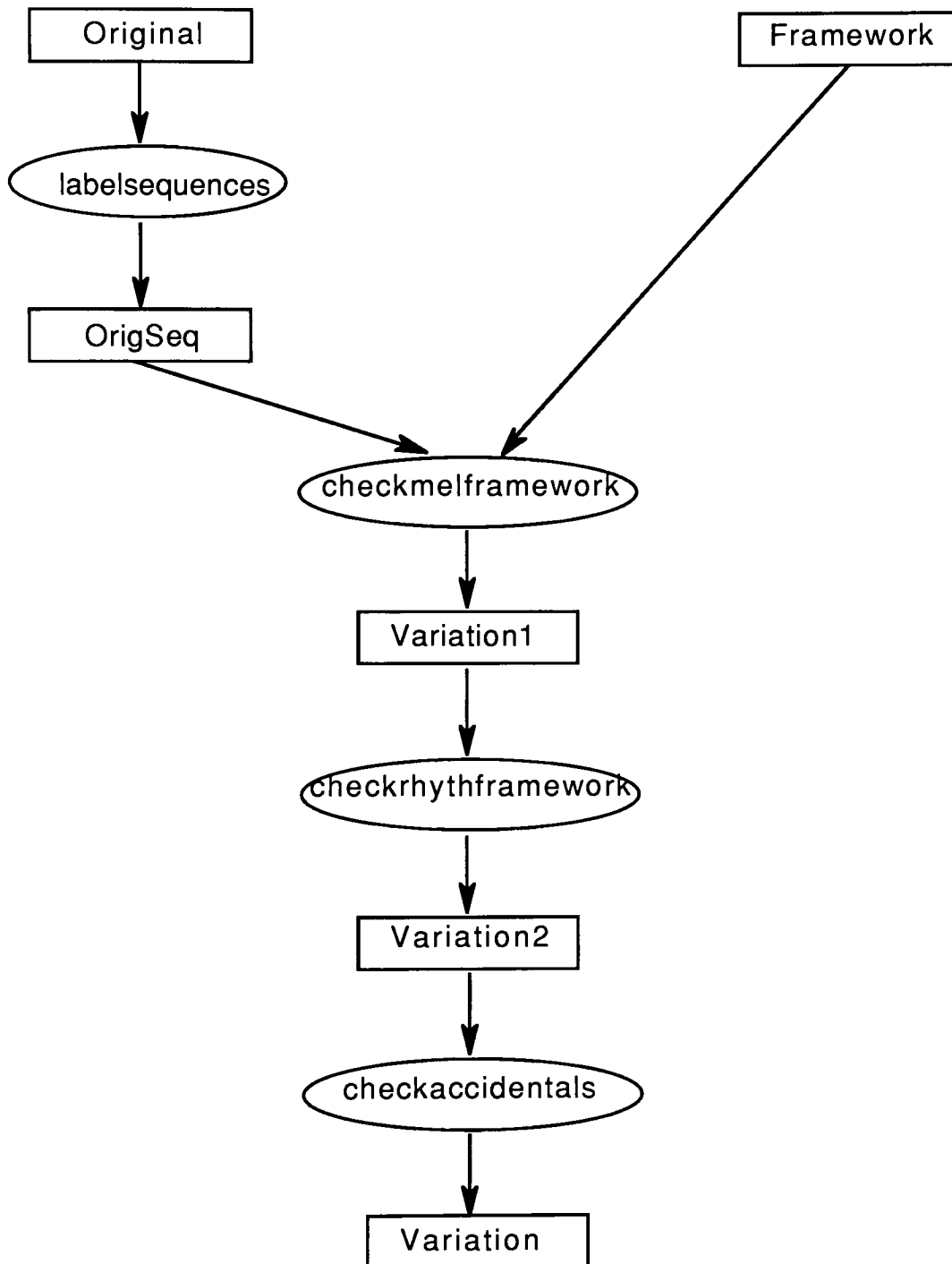
After using the analysis of Baroque style rules and Telemann variations to build rhythmic and melodic vocabularies, transposing the pieces to be played, and assimilating the style of the original and Telemann variations by playing them, the expert seems to do the following in the generation of a variation:

- 1) for a given phrase, look at the figured bass and determine  
the voicing possibilities and which of them was used in the  
original (see Appendix 1)
- 2) categorize the melody of the original, find "minimized" class  
(Schenker)
- 3) categorize the rhythm of the original, find its class
- 4) generate variation on above: usually by changing the  
original melody and/or rhythm according to classes  
and vocabularies as above or occasionally by using  
different voicing possibilities to create contrast
- 5) keep in mind more global musical issues such as sequences,  
movement toward the end of the phrase and piece and  
continuity within the piece
- 6) loop back to step 1



To accomplish these steps, the basic structure of the Prolog implementation is as follows - data (the lists) are enclosed in boxes and rules (the transformation "procedures") are circled:

## VARIATION DEVELOPMENT METHOD



### 4.3.2 Labelling Melodic Sequences

A melodic sequence begins with a group of notes between which are certain intervals. If this interval pattern is repeated several times with each repetition beginning on a different pitch, a sequence is formed.

In this implementation, a sequential motif may contain four to nine notes, and this motif is repeated two times (three occurrences total), immediately following the original. In the more general sense there is no limit on the length of the sequence, neither in terms of number of notes per grouping nor the amount of space between repetitions. The implementation could be changed to reflect different choices for length; the numbers chosen above were sufficient to describe the sequences found in the examples used.

The "Original" input melody is analyzed for sequences and a new list "OrigSeq" is produced which contains the original melody with sequences bracketed and labelled. A sequence (of four to nine notes) is identified by pattern matching adjacent notes and their intervals, and if found, replacing the sequence in the new list by grouping the notes of each section of the sequence and also of the sequence as a whole into a bracketed list structure. If no sequence is found beginning with the current note, that note is added to the new labelled list unchanged and the same searching/matching procedure continues with the rest of the original piece.

Following the example (Loeillet\_dm), the list changes look as follows:

<u>Original</u>		<u>OrigSeq</u>
	↓	
[[[am],[q],[e1]], [[am],[e],[c1]], [[am],[e],[a]],		[[[am],[q],[e1]], [[am],[e],[c1]], [[am],[e],[a]],
[[gsdim],[q],[f1]], [[gsdim],[te],[f1]], [[gsdim],[s],[e1]],		[[gsdim],[q],[f1]], [[gsdim],[te],[f1]], [[gsdim],[s],[e1]],
[[gsdim],[s],[d1]],		[[gsdim],[s],[d1]],
[[am],[e],[c1]], [[am],[e],[e1]], [[am],[e],[a]], [[am],[e],[b]],		[[am],[e],[c1]], [[am],[e],[e1]], [[am],[e],[a]], [[am],[e],[b]],
[[f],[e],[c1]], [[f],[e],[e1]], [[dsdim],[q],[ds1]],		[[f],[e],[c1]], [[f],[e],[e1]], [[dsdim],[q],[ds1]],
[[e],[dq],[e1]], [[e],[e],[e1]],		[[e],[dq],[e1]], [[e],[e],[e1]],
[[am],[e],[c1]], [[am],[e],[e1]], [[am],[e],[a]],		[melseq,[[seq1,[[[am],[e],[c1]],[[am],[e],[e1]],
[[am],[e],[c1]],		[[am],[e],[a]], [[am],[e],[c1]],
[[dm],[q],[f1]], [[dm],[ts],[f1]], [[dm],[s],[e1]],		[[dm],[q],[f1]], [[dm],[ts],[f1]], [[dm],[s],[e1]],
[[dm],[s],[d1]], [[dm],[s],[c1]],		[[dm],[s],[d1]], [[dm],[s],[c1]]],
[[g7],[e],[b]], [[g7],[e],[d1]], [[g7],[e],[g]],		[seq2,[[[g7],[e],[b]], [[g7],[e],[d1]], [[g7],[e],[g]],
[[g7],[e],[b]],		[[g7],[e],[b]],
[[c],[q],[e1]], [[c],[ts],[e1]], [[c],[s],[d1]], [[c],[s],[c1]],		[[c],[q],[e1]], [[c],[ts],[e1]], [[c],[s],[d1]], [[c],[s],[c1]],
[[c],[s],[b]],		[[c],[s],[b]]],
[[f],[e],[a]], [[f],[e],[c1]], [[f],[e],[f]],		[seq3,[[[f],[e],[a]], [[f],[e],[c1]], [[f],[e],[f]],
[[f],[e],[a]],		[[f],[e],[a]],
[[bdim],[q],[d1]], [[bdim],[ts],[d1]], [[bdim],[s],[c1]],		[[bdim],[q],[d1]], [[bdim],[ts],[d1]], [[bdim],[s],[c1]],
[[bdim],[s],[b]], [[bdim],[s],[a]],		[[bdim],[s],[b]], [[bdim],[s],[a]]],
[[e],[e],[gs]], [[e],[e],[b]], [[e],[e],[e]], [[e],[e],[e1]],		[[e],[e],[gs]], [[e],[e],[b]], [[e],[e],[e]], [[e],[e],[e1]],

(Original continued)

| (OrigSeq continued)

\_\_\_\_\_

[[am],[q],[c1]], [[am],[te],[c1]], [[am],[s],[b]], [[am],[s],[a]],

[[e],[e],[gs]], [[e],[s],[a]], [[e],[s],[b]],

[[e],[de],[b]], [[e],[s],[a]],

[[am],[dq],[a]]].

[[e],[de],[b]], [[e],[s],[a]],

[[am],[dq],[a]]].

### 4.3.3 Melodic Variation

The group of clauses which performs this phase of the transformation steps through the OrigSeq and Framework lists in parallel and creates a new list, NewLs, a list with melodic motif variations substituted for original motives. There may be many Original notes for each Framework note. In this implementation up to five extra Original notes may exist between two framework notes (this covers all examples), but this would not be difficult to change. Melodic motif pattern substitutions are developed in what can be viewed as a multiple viewpoints loop similar to that used by Ebcioglu in his CHORAL expert system. Specifically, in this implementation, for each note in the framework substitutions are based on 1) sequences within the Original, 2) the available and valid melodic vocabulary possibilities, and 3) what changes have been made thus far within the variation (frequencies). The loop progresses through each of these three viewpoints for each note or note group analyzed before moving on to the next note (group) to be processed.

#### 4.3.3.1 Mirroring Sequences

If the Original melody has a sequence beginning at the current note, that sequence will be substituted/developed for the variation in such a way as to ensure that each part of the sequence is developed using the same melodic vocabulary entry, thus mirroring the sequence in the original. The first occurrence of the

sequential pattern will be developed in the same manner as a nonsequential occurrence (see below), but facts are added to the database describing which motives were used so that in developing subsequent occurrences the same motif can be used by checking these database facts.

#### 4.3.3.2 Melodic Vocabulary Possibilities

If no sequence is found starting at the current framework note's corresponding Original position, patterns of notes starting from the current framework pitch (from 5 to 2 notes) are checked to see if there is a motif in the database that matches that pattern. That is, the 5 note pattern beginning at the current note is checked to see if there is a valid melodic pattern match, then 4 notes and so on to 2. If no pattern match is found, the single note is placed unchanged in the new list and the process continues with the rest of the piece (list).

#### 4.3.3.3 Frequencies/Choosing Between Valid Melodic Vocabulary Entries

If there are several valid substitutions for a motif, the system randomly chooses between these valid possibilities. An added constraint keeps track of the number of times a particular possibility has been chosen in the past and chooses according to the following priority list (which could easily be modified): first choose one that has been used once, if none are available then choose one that has

been used three times, then never, then twice. This is done in an effort to model the expert's attempt to keep symmetry within the piece but to also avoid too much repetition when possible.

Changes after this phase (using the same example as has been followed in the previous discussions) and a trace of the melodic vocabulary substitutions appear on the next several pages:



## Framework

### Variation1

			[[[am,am,am],[q,e,e],[e1,f1,g1,e1,c1,d1,e1,c1,a]],
			[[gsdim],[q],[f1]],
			[[gsdim,gsdim],[te,e],[f1,f1,e1,d1,c1,d1]],
			[[am,am],[q,e],[c1,c1,b,a,g,a]], [[am,f],[e,q],[b,trill,c1]],
			[[dsdim,e],[q,dq],[ds1,trill,e1]],
			[[e],[e],[e1]],
			[[[am,am],[q,q],[c1,g,a]],
			[[dm],[q],[f1]], [[dm,dm],[te,e],[f1,c1,d1]]],
			[[[g7,g7],[q,q],[b,f,g]],
			[[c],[q],[e1]], [[c,c],[te,e],[e1,b,c1]]],
			[[[f,f],[q,q],[a,e,f]],
			[[bdim],[q],[d1]], [[bdim,bdim],[te,e],[d1,a,b]]],
			[[e,e],[q,q],[gs,e,d,e]],
			[[am],[q],[c1]], [[am],[te],[c1]], [[am],[s],[b]], [[am],[s],[a]],
			[[e,e],[e,s],[gs,trill,a]], [[e],[s],[b]],
			[[e],[de],[b]], [[e],[s],[a]],
			[[am],[dq],[a]]].

## MELODIC VOCABULARY ENTRIES TRACE

- e1, c1, a --> e1, f1, g1, e1, c1, d1, e1, c1, a  
    melvocab(dtriad, 1, [A,B,C], [A,X,Y,A,B,Z,A,B,C]) :-  
        desc\_third(A,B),  
        desc\_third(B,C),  
        asc\_second(A,X),  
        asc\_second(X,Y),  
        asc\_second(B,Z).
- f1 passes through
- f1, d1 --> f1, f1, e1, d1, c1, d1  
    melvocab(d3, 4, [A,B], [A,A,X,B,Y,B]) :-  
        desc\_third(A,B),  
        desc\_second(A,X),  
        desc\_second(X,B),  
        desc\_second(B,Y).
- c1, a --> c1, c1, b, a, g, a  
    melvocab(d3, 4, [A,B], [A,A,X,B,Y,B]) as above
- b, c1 --> b, trill, c1  
    melvocab(am2, 1, [A,B], [A,trill,B]) :-  
        asc\_min\_second(A,B).
- ds1, e1 --> ds1, trill, e1  
    melvocab(am2, 1, [A,B], [A,trill,B]) as above
- e1 passes through (does not get grouped because next note  
    begins sequence

## VOCABULARY TRACE CONTINUED

(sequence)

(occurrence 1)

- c1, a --> c1, g, a  
    melvocab(d3, 1, [A,B], [A,X,B]) :-  
        desc\_third(A,B),  
        desc\_fourth(A,X),  
        asc\_second(X,B).

f1 passes through

- f1, d1 --> f1, c1, d1  
    melvocab(d3, 1, [A,B], [A,X,B]) as above

(occurrence 2)

- b, g --> b, f, g  
    e1 passes through   - e1, c1 --> e1, b, c1

(occurrence 3)

- a, f --> a, e, f  
- d1 passes through  
- d1, b --> d1, a, b

(end of sequence)

- gs, e --> gs, e, d, e  
    melvocab(d3, 3, [A,B], [A,B,X,B]) :-  
        desc\_third(A,B),  
        desc\_second(B,X).

-rest pass through except trill

As is discussed in the MELOS2 system, we can determine that the rules governing these three viewpoints can be identified as macroformal or microformal. The rules regarding sequences and determination of valid vocabulary options can be viewed as microformal, in that they are specific to a particular phrase and are not influenced by or particularly concerned with the overall piece. The constraints used through the frequency counts, however, are more macroformal in nature and are aimed at giving cohesion to the piece as a whole.

The use of constraints in this implementation also parallels the nature of the constraints described in the Lerdahl and Jackendoff system, in which choices within viewpoints are made and are then checked against constraints which monitor the relationships between the pieces of the whole. Similarly the Ebcioglu system lets each viewpoint go as far as it can on its own in making decisions (choosing possible note values) and uses constraints to reject certain potential assignments.

#### 4.3.4 Rhythmic Variation

The variation's rhythm is then developed using a time slice view, substituting by rhythmic group (which were made depending on the melodic groupings made). The rhythmic variation development section determines whether a sequence has been involved or not and modifies either a sublist which

is non-sequential or a sublist which does include a sequence (extra bracketing that had been used to notate the sequence is removed at this point). The modification occurs by first finding the number of notes in the new melodic motif which has been substituted. Second, rhythmic vocabulary facts are checked in the database, which supply substitutions given the original rhythmic motif and the number of notes in the new motif.

If several rhythmic substitutions apply a random choice is made in a way similar to that in the melodic development process. Unlike the melodic approach to assuring that sequences carry over into the variation, however, it is only necessary here to use the frequency constraints to assure rhythmic continuity through the sequence rather than additionally adding temporary facts to the database.

The following several pages show the changes made to the list during this phase and a trace of the rhythmic vocabulary entries used in the transformation process.

## Variation1

[[[am,am,am],[q,e,e],[e1,f1,g1,e1,c1,d1,e1,c1,a]],  
[[gsdim],[q],[f1]],  
[[gsdim,gsdim],[te,e],[f1,f1,e1,d1,c1,d1]],  
[[am,am],[q,e],[c1,c1,b,a,g,a]],  
[[am,f],[e,q],[b,trill,c1]],  
[[dsdim,e],[q,dq],[ds1,trill,e1]],  
[[e],[e],[e1]],  
[[[am,am],[q,q],[c1,g,a]],  
[[dm],[q],[f1]], [[dm,dm],[te,e],[f1,c1,d1]]],  
[[g7,g7],[q,q],[b,f,g]],  
[[c],[q],[e1]], [[c,c],[te,e],[e1,b,c1]],  
[[f,f],[q,q],[a,e,f]],  
[[bdim],[q],[d1]], [[bdim,bdim],[te,e],[d1,a,b]]],  
[[e,e],[q,q],[gs,e,d,e]],  
[[am],[q],[c1]], [[am],[te],[c1]], [[am],[s],[b]], [[am],[s],[a]],  
[[e,e],[e,s],[gs,trill,a]], [[e],[s],[b]],  
[[e],[de],[b]], [[e],[s],[a]],  
[[am],[dq],[a]]].

## Variation2

[[[am,am,am],[s,s,s,s,s,th,th,s],  
[e1,f1,g1,e1,c1,d1,e1,c1,a]],  
[[gsdim],[q],[f1]],  
[[gsdim,gsdim],[tth,th,th,s,s],[f1,f1,e1,d1,c1,d1]],  
[[am,am],[s,s,s,s,s],[c1,c1,b,a,g,a]],  
[[am,f],[s,s,q],[b,trill,c1]],  
[[dsdim,e],[q,dq],[ds1,trill,e1]],  
[[e],[e],[e1]],  
[[am,am],[e,e,q],[c1,g,a]],  
[[dm],[q],[f1]], [[dm,dm],[ts,s,e],[f1,c1,d1]],  
[[g7,g7],[e,e,q],[b,f,g]],  
[[c],[q],[e1]], [[c,c],[ts,s,e],[e1,b,c1]],  
[[f,f],[e,e,q],[a,e,f]],  
[[bdim],[q],[d1]], [[bdim,bdim],[ts,s,e],[d1,a,b]],  
[[e,e],[t,t,t,q],[gs,e,d,e]],  
[[am],[q],[c1]], [[am],[te],[c1]], [[am],[s],[b]], [[am],[s],[a]],  
[[e,e],[s,s,s],[gs,trill,a]], [[e],[s],[b]],  
[[e],[de],[b]], [[e],[s],[a]],  
[[am],[dq],[a]]].

## RHYTHMIC VOCABULARY ENTRIES TRACE

- rhythvocab([q,e,e], 9, 1, [s,s,s,s,s,th,th,s]).
- rhythvocab([te,e], 6, 1, [tth,th,th,th,s,s]).
- rhythvocab([q,e], 6, 1, [s,s,s,s,s,s]).
- rhythvocab([e,q], 3, 1, [s,s,q]). (problem with trill)
- rhythvocab([q,dq], 3, 1, [q,dq]). (no vocab entry)
- pass through e
- rhythvocab([q,q], 3, 1, [e,e,q]).
- pass through q
- rhythvocab([te,e], 3, 1, [ts,s,e]).
- rhythvocab([q,q], 3, 1, [e,e,q]).
- pass through q
- rhythvocab([te,e], 3, 1, [ts,s,e]).
- rhythvocab([q,q], 3, 1, [e,e,q]).
- pass through q
- rhythvocab([te,e], 3, 1, [ts,s,e]).
- rhythvocab([q,q], 4, 1, [t,t,t,q]).
- same except trill (e,s --> s,s,s)

#### 4.3.5 Checking Accidentals

This phase steps through the list structure examining one note at a time to ensure that it fits within the chordal structure that underlies it. Instances where incorrect accidentals exist in the piece arise because of the nature of the melodic variation process. In order to use the same melodic vocabulary entries for any group of pitches which has the same characteristic interval or intervals (a3, d4, dtriad, etc.) the pitch substitutions were not done on an absolute pitch basis. That is, there are not separate entries for a descending major third beginning on an e and a descending major third beginning on an f. Rather, the descending third transformations are based on some combination of variables which are instantiated by the use of the interval facts in the database relative to the particular absolute notes being transformed. In this "non-absolute" process of variable instantiation some accidentals chosen from the database will not necessarily be appropriate with the chords with which these notes are associated. The correction of this problem is done by the "changeaccidental" clauses.

Since the notes are grouped by the melodic and rhythmic substitutions that were made to the original and framework, checking each note consists of checking each note within each sublist (group) of notes against all the chords of that group. The database contains facts which state which accidentals should be changed for each chord. If a match (of a "bad" note) is found, it is replaced in the list by the given accidental (the same pitch name but with or without a sharp



or flat). Otherwise the note is acceptable and is left as is in the new list.

In the Loeillet\_dm example traced in the previous sections, no accidental changes are made. Thus the final version of the variation is that listed after the rhythmic transformations (section 4.3.4). Staff notation of this output variation (with the original melody included for comparison purposes) follows:

Variation

original am g#dim am F #dim G am

Variation

dm original G7 C F Bdim E

Variation

original am E am

## 4.4 The Database

Entries in the database can be divided up into four basic sections: 1) melodic and rhythmic vocabulary facts, 2) melodic and rhythmic frequency facts, 3) changeaccidental facts, and 4) basic musical interval facts.

### 4.4.1 Vocabulary Facts

Both melodic and rhythmic vocabulary facts match a given pattern with a valid substitution for that pattern.

Melodic facts look like:

`melvocab(a3,1,[A,B],[A,X,X,B]) :-`

`asc_third(A,B), asc_fourth(A,X), desc_second(X,B).`

The first argument labels the interval or motif (Ex: `dtriad`, `a5run`) represented by the given pattern. The second argument is an identification number used to distinguish between different substitutions for the same interval/motif. The third argument represents the original notes (list) to be changed, and the fourth gives a valid substitution given the subgoals, which fill in this last argument.

Rhythmic vocabulary facts look like:

`rhythvocab([q,e,e],6,1,[s,e,s,s,s,e]).`

Here the first argument is the old group of rhythms, the second is the number of notes in the new melody grouping for which a rhythm is needed, the

third is the identification number for that original rhythm, and the last is a valid substitution. All rhythmic substitutions keep the same total number of beats between old and new versions.

Melodic and rhythmic facts were garnered from the expert's rhythmic and melodic vocabularies (see Appendix 1). These vocabularies were built by the expert after analysis of the Telemann variations. When an original melodic or rhythmic motif appeared for which there was not a direct entry in the expert's vocabulary "tables", Telemann variations were investigated to try to find such a transformation or, if necessary, entries that were available to fit actual versions found in the examples were adapted.

#### 4.4.2 Frequency Facts

For each melodic and rhythmic vocabulary entry there is a corresponding fact in this section of the database which keeps track of the number of times that option (substitution) has been used to date in the variation. All are initialized to 0.

Melodic frequency facts

freq(a3,1,0)

state the interval, melodic identification number for that interval, and frequency of use for that option.

Rhythmic frequency facts

freq([q,e,e],6,1,0)

list the original rhythm, number of notes in the new melodic pattern for that same rhythmic duration, identification number for that old rhythm/number of notes combination, and frequency of use for that option.

#### 4.4.3 Change Accidental Facts

Example: changeaccidental(gs, c, g).

Change accidental facts list the original pitch, the chord against which that pitch is being checked, and the new pitch to substitute if the first two conditions are met.

#### 4.4.4 Basic Musical Interval Facts

These rules and facts include the definitions of musical intervals based on facts relating to the interval of the second (two adjacent letter name notes). No distinction is made between major and minor intervals but in listing the facts of seconds both major and minor seconds are listed.

Examples:

asc\_fifth(X, Y) :- asc\_third(X,Z), asc\_third(Z,Y).

asc\_third(X,Y) :- asc\_second(X,Z), asc\_second(Z,Y).

asc\_second(c, d).

`asc_second(c, ds).`

Unison facts are also listed and are used in checking if two pairs of notes have the same interval between them (for instance, when checking intervals within potential sequences).

## 5. RESULTS/CONCLUSIONS

### 5.1 Introduction: Syntax vs. Semantics

Musical compositions are difficult to analyze using quantifying measurements; often the success of a piece is based on the "feeling that it sounds right". Many musicians argue that syntactically correct music is not necessarily good "music" at all; there are elements of a "musical" piece that are beyond the mere observance of rules. Here we face a fine line of distinction, and, making things even more difficult, where that line is drawn is not clear.

In addition, we may ask where this system's variation output fits on this spectrum. Clearly all of its transformations are based on strict rules, all of which are programmed before any musical generation takes place. Does this, therefore, systematically eliminate the possibility that the output is musical or even musically acceptable?

Given the emotional aspect of music listening and evaluation it is clearly difficult to scientifically "measure" the performance of a computer program that generates such compositions. Fortunately in the present case we can identify whether the basic syntactic musical goals of the system were achieved and can rely on a less scientific semantic analysis based on the musical judgment and comments of the expert.

## 5.2 Results

For each of the ten movements in Appendix 2 two variations were generated. In Chapter 4, in explaining the approach of the system, its goals were also described. Those goals were met almost entirely as described, and the system "works" in a fundamental sense: variations are generated given an original melody with chordal structure and a framework of that original. This was done modelling the expert, using melodic and rhythmic vocabularies and paying attention to more global issues such as sequences and continuity within the piece (through the use of frequencies).

Of these syntactic goals, the only one that occasionally has problems meeting the specifications as described is the checking of accidentals at the end of the transformation process. As described in chapter 4, during this last pass through the list, each pitch in a group's pitch list is checked against each of the chords in the chord list. Herein lies the problem. If two chords in the chord list support different accidentals of the same pitch name, the system changes the accidental to fit each one. Therefore, the last chord in the list in effect takes precedence over other chords since each note is checked against that one last. For example, in Telemann\_gm variation 2 measure 7 beat 1, the fs1 should be an f1 since it is supported by a dm chord. The grouping developed from the framework was the f1 from beat 1 and the d1 from beat 2. This was substituted by g1,f1,e1,d1 with the chordal grouping dm,D7. When checking accidentals,



each pitch, when checked against the dm chord, was left alone. Since each pitch was checked against the D7 chord second, however, the f1 was changed to and left as an fs1. This problem occurs fairly infrequently but should be solved (suggestions are discussed in section 5.4).

Another syntactic issue that is not an "error" but rather an awkward notation should be mentioned. Trill notation, as described in chapter 4, uses the symbol "tr" as a separate pitch added to a pitch list between two notes where the trill is to take place. Because the "tr" is listed as a separate pitch, in the rhythmic development of the note group the resulting rhythmic motif chosen for the variation contains a rhythmic duration for the "tr" symbol. When transcribing the variations to musical notation this rhythmic duration is added to the note preceeding the "tr" notation (the note on which the trill is performed). For example, in Telemann\_BF variation 1 measure 1 beat 2, the gs1 trill to a1 is notated in the output as `[[am,am],[s,s,e],[gs1,trill,a1]]`. The gs1 is musically notated as an eighth note trill (the two sixteenth note notations combined) and the a1 retains its eighth note designation.

Finally in the syntactic realm, there will be cases in the system's output where a rhythmic vocabulary entry will not exist for a particular melodic transformation that has been made. Most of the cases in these examples are covered, but one such situation was left in for the sake of example. In the Handel\_F variation 2 output, the following appears (this is circled in the Appendix 2

listing of the output): `[[g7,g7],[tdq,e],[f1,f1,e1,d1,c1,d1]]`. The rhythmic vocabulary entry to match this grouping would need to be `rhythvocab([tdq,e],6,1,[some 6 note 2 beat rhythmic substitution here])`. It does not exist in the database, but could be easily remedied by adding such a rhythmic vocabulary fact (and a corresponding frequency fact) to the database.

Having thus discussed the syntactic accomplishments of the system we turn to the more subjective comments of the expert for a semantic analysis. The expert's analysis suggests that the program did achieve success in creating musically satisfying pieces. He feels that the computer variations contain "ideas" that are more interesting and/or unusual in many cases than those that he generates himself. The variations tend to make more structural changes than those of the expert, and his seem to be more confined to ornamentation on the existing tune. This could be explained by the more "real-time" quality of the expert's generation. Rather than breaking up the transformation and making several passes through the piece, each with a different and specific functional focus, the expert must approach the process as he's playing and has less "memory" available to focus on the overall picture of the underlying structure. Especially in the more complex areas such as sequences, he was pleasantly surprised by the system's output.

There are less positive sides to the interesting combinations made by the system, however. Occasionally the variation becomes too "far-out" and the tune

is lost. Altering the melodic vocabulary so that only "safer" variations are generated would eliminate this, but would also eliminate the positive results of the use of these more unusual transformations, as described above. The expert did find several melodic and rhythmic vocabulary entries that had been adopted that he felt should be eliminated. For example, he thought that in Loeillet\_G variation 2 measure 6 beat 2, the three repeating sixteenth notes should be avoided and should be a single dotted eighth instead. This three repeating notes motif is part of the melodic vocabulary entry `melvocab(d5,1,[A,B],[A,X,B,B,B])` which is used in this particular case and in other variations.

Other than these few specific vocabulary changes, though, changes in the process could be made that would allow the interesting vocabulary combinations to remain but would address the majority of the problems leading to the loss of the tune of the original melody. When analyzing the results, we often found that the "bad" spots occurred when the framework had strayed too far from the original, leaving too many notes out from the original melody line, which allowed the variations to obscure the tune. For example, in Loeillet\_G variation 1 measure 5 beats 3 and 4, the complicated pattern strays too far from the original melody line. If the framework had included the fourth beat sixteenth notes from the original, upon variation the fourth beat would have stayed much closer to the tune. In some other cases, by putting too few notes from the original in the framework the resulting variation became boring. For example,

in Loeillet\_dm framework measure 2 beat 3, the e1 should have been put in the framework along with the c1. This would have transferred through to both variations and better reflected the original melody line.

Another change that would help to keep the prominence of important tune notes intact in the variations addresses the "movement" of the piece. Certain notes should be stressed because they are at a certain part of a phrase or a piece; the variation should "move" toward these notes. Different parts of the phrase or at least critical notes in these positions should be marked somehow and movement around these notes should be monitored.

Too little movement makes the piece stall. This was identified by the expert in Loeillet\_G variation 1 measure 4. Whereas in measure 3 there was a lot of movement, the transition to measure 4 drops off too dramatically and leaves a sense of loss of movement. Too much movement, on the other hand, can make the variation sound too obscure, such as in Loeillet\_G variation 1 measure 9, especially when the original was already quite dense. Here we also have the additional consideration of where we are in the piece. We have entered a secondary theme based on chords that are more "far-out" than the beginning and ending sections of the piece. In these situations, less variation must be done in order to keep the tune in focus. One example, which the expert identified as showing a good balance of movement, was Telemann\_C variation 2 measures 4 and 5. In measure 4 the movement is accelerated from the original, and yet this

is balanced in measure 5 by slowing down. It is this overall balance that is important.

### 5.3 Problems Encountered

Two problems that needed coding attention during the development process will be discussed here: the poor efficiency of the random choice of valid vocabulary entries and the difficulty of supporting the multiple viewpoints aspect of the generation process.

#### 5.3.1 Random Choice Efficiency

The random number generator provided by the AAIS Prolog package generates numbers within the approximate range -32000 to 32000. Using a random number to determine which of several vocabulary entries should be chosen was complicated because for different vocabulary categories (intervals) there were different numbers of valid options. Since in the current version of the program there are at most four valid substitutions for any motif, the random number range was divided into four equal segments. That is, any number returned by the random number generator less than -16000 would be assigned option #1, those between -16000 and 0 option #2 and so on. In order to implement the constraint system whereby those options with certain frequencies

would be chosen first, the choice of one of these random numbers and its corresponding option is not necessarily acceptable, however, since we have a priority of frequencies that must be followed. The program calls the randomchoice goal as many times as necessary with different frequency options until it finds one that successfully satisfies both vocabulary and frequency criteria.

The implementation of the system as outlined above originally caused inefficiency problems since often many random choices had to be made (with quite a high number of repeats given the wide ranges with only four options) in order to find an acceptable one. Several pieces of code were added to this area of the implementation to focus on improving its efficiency.

In the first, rather obvious (and necessary) improvement, before any searching and matching is done by the "changemel" clauses, the melodic vocabulary entries in the database are checked to assure that there exists at least one entry for the interval(s) in question. For example, in "change5bmel", the five notes that will potentially be substituted are first matched to the melodic vocabulary entries to see if an entry matching those five notes exists. If so, then the goal proceeds to call "rndmchoice". If not, no choosing is done; this first check allows us to pass directly through to the "change4bmel" clause.

Second, when in the random choice section of code, before any random number is generated, the frequency facts are checked to make sure that there is at

least one entry for the current interval which has the desired frequency. If not, the goal fails and backtracks to try the next most desirable frequency. If there is at least one entry, a random number is generated and its range (option number) determined. The option to which this range corresponds (1 through 4) is checked to see if it is one with the desired frequency.

Here we see the third efficiency improvement area. If the option is not one with the desired frequency, a fact "alreadytried(X)" where X is the option number, is asserted into the database. Using these alreadytried facts allows us to pass over the analysis of that option's suitability if it is randomly chosen again. If the option is one that does have the desired frequency, this is the option that will be used; the corresponding melodic vocabulary entry is used to get the new notes to be substituted for the old pattern. The frequency fact in the database is updated, and any other database entries that were made to mark options alreadytried during this process are removed.

Occasionally with this range implementation we get a message such as

"<Undefined Goal>:

||||| (12:168)alreadytried(4).:

in the middle of the variation development. This seems to occur in cases where there are not four entries for a given vocabulary interval (there is no option #4). By simply pressing the return key the program continues and successfully produces a variation.

### 5.3.2 Multiple Viewpoints

The second difficult development issue was encountered when implementing the multiple viewpoints aspect of the generation process. In order to support the multiple viewpoints it was necessary to be able to view both the Framework and OrigSeq lists within the same loop (starting at the same note). In other words, it was necessary to move along through the pieces in parallel. The problem is compounded by the fact that the notation (subgroupings) within OrigSeq differs from the "normal" notation when in a sequence for identification purposes.

There are basically two cases to be addressed. First, if there is no sequence in the original version of the piece starting at the current note during the "checkmelframework" portion of the transformation, we must move ahead in the original melody as well as in the framework so that after the transformation process the two versions will be lined up for the next recursive call. Second, if there is a sequence we must map the sequence from the original onto the framework so that the variation can be done as usual from the framework base.

The first, non-sequential case is handled in two steps. Using the "mapFrameworkintervaltoOrig" or "onenotemapFrameworkintervaltoOrig" (if only one Framework note will pass through) clauses, a group of framework notes is mapped to their corresponding notes in the original by recursively finding the



original notes for each framework interval represented in the framework group of notes being examined. These original notes corresponding to one framework interval could include situations where: the original has only the same two notes as the framework interval, there are up to 5 extra original notes between the two framework notes, either of the above but the last framework interval of the list is being examined, the last original note and the last framework note are all that remain, or the original has up to 4 extra notes before the next framework note. The corresponding original notes are grouped into a list called TempOriginal.

The pattern matching/substitution of the framework grouping is then done as discussed earlier, choosing the melodic transformations for the variation. The framework "pointer" of where we are in creating the variation is updated automatically through the unification process. The original version must also move past those notes which correspond to those developed in the framework. Since TempOriginal contains those notes which mirror the framework notes which have been used, we can assume that TempOriginal is an initial sublist of Original and that whatever else follows that sublist will be the rest of the original. The "updateOriginalpointer" clauses take the Original and TempOriginal lists and create this new list RestOriginal, which is then used in the recursive call to vary the rest of the piece.

The solution to the second case, where sequences are involved, approaches the mapping in the opposite direction. Here we need to map what is already

known in the Original (the sequence) to the Framework version so that we can use the framework in the variation process as usual and that in the development of the framework the original sequence will be mirrored. Because the notation in the Original has the sequence parts bracketed, we can use each of these separate sublists to find the appropriate sequence boundaries in the framework. The framework representation of the sequence is stored in three separate lists, Framework1 through Framework3. Each of these lists is derived by using an Original sequence occurrence list and pulling off the corresponding Framework notes using "findseqboundary" clauses.

After these temporary framework sequence lists are built, the melodic transformations are determined as discussed earlier. The first occurrence is varied as a non-sequential pattern would be (using the vocabulary and the random function with priorities). The second and third occurrences are then developed to model the first (using facts added to the database), thus leaving the sequence intact. In this case both the Original and the Framework "pointers" are already pointing to the next note in the piece to be checked. The framework was updated through the unification process during the melodic development (as in the non-sequential case) and the original was updated during the mapping to find the sequence boundaries in the framework.

## 5.4 Shortcomings and Future Directions

The following areas reflect some shortcomings of the current system. I have included suggestions for future work which could address these limitations.

1. Awkwardness of trill notation could possibly be ironed out by attaching the "tr" notation to the pitch it modifies rather than adding it to the melodic grouping as a separate pitch. This would eliminate the problems that occur in the checkrhythframework clauses whereby rhythms are found for a grouping which counts the note "tr" and therefore assigns the "tr" some rhythmic duration. No problems should arise in the checkaccidentals section because of this suggested approach; the two pitches forming the trill should never be changed by these clauses anyway since they are both directly from the framework and would fit correctly with the chord structure.
2. The problems arising from the current checkaccidentals implementation result from the grouping of pitches and chords during the checkmelframework portion of the transformation process. These groupings do not necessarily contain one chord for each pitch, but rather there may be two or three chords for as many as eight or nine pitches. When checking the accidentals, then, we don't know where

in the pitch progression the chord changes. Currently we check all pitches against all chords of the group. An improvement would check pitches only against the particular chord that supports them. This could perhaps be accomplished by associating a chord with each pitch. That is, when melodic vocabulary substitutions are done, as a pattern of pitches is replaced, a chord entry would be made for each pitch added. Often this would simply mean repeating a chord label as many times as there are pitches added. Sometimes, however, the pitches being replaced do not have the same chord, and then some decisions would have to be made about which notes of the new pattern should have a certain chord and where we should switch to the new chord. This would somehow have to be added to the vocabulary entries.

3. It would be more realistic if the handling of sequences were a bit more extensive. For instance, in the current implementation, if the same interval occurs two times in the first occurrence of a sequence and two different melodic vocabulary entries are chosen to develop those intervals (and corresponding facts are entered into the database), the second and third occurrences of the sequence will not know which entry to follow to mirror the first occurrence. More

globally, it would be nice to have sequence labelling cover nested and nonadjacent sequence occurrences, and perhaps to analyze the framework for any sequences that exist there and should be developed on this deeper level.

4. To solve the semantic issue of better addressing the differences in function of the parts of a phrase and the way phrases fit together, the Baroni and Jacobini approach could be examined. The use of rules to govern the transitions between phrases as well as within phrases could help to ensure that the piece has more coherence globally. Also somehow marking the phrase parts and having some heuristics/constraints about which development approaches are appropriate in which cases could be beneficial. As a small improvement, when any pitches are not changed from the framework to the variation, the corresponding original pitches could be passed through to the variation rather than passing through the unchanged framework note.

5. Generating the framework from the original should eventually be done by the computer rather than the user if possible. The work of Ebcioglu includes the building of a Schenkerian tree analysis from a

melody and bass line using a stack. Perhaps this could be refined so that after such a tree is built some interior level of the tree could be identified and used as the machine generated framework.

6. Once such a tree structure was in use it might also be possible to make more complicated use of the voicing possibilities identified by the expert in his approach. These could be added as another of the multiple viewpoints, used in steering the choice of vocabulary choices or perhaps taking an existing choice and modifying it to use the new voicing possibility as part of the choice's final pitch selection.
7. Most systems could adopt more user-friendly features. This system is certainly no exception. Particularly the input and output representations are difficult. With the new MIDI sound processing technology work could be done to receive input from a keyboard and produce audio output. Once this is achieved, the use of the system as an interactive program (like the systems of Cope and Fry) would be much more feasible. The system could generate its variation and then receive feedback from the expert about certain areas of the piece which it could then change, or the expert could add specific

variation elements.

## 6. APPENDIX 1: EXPERT'S FIGURES

Included in this appendix are the following figures from the expert's thesis:

- figured bass example
- determining voicing possibilities from a figured bass
- melodic vocabulary examples
- rhythmic vocabulary examples



# Figured Bass Example

## SONATE G-MOLL

Adagio.

figured bass

realization of figured bass

10

# Determining Voicing Possibilities from Figured Bass

**XLVIII** **XLIX**

Voicing Possibility

a.

Figured Bass

original voicing possibility

new choice.

d.

original + new to form variation

e.

# Melodic Vocabulary

①

Exercise 1 consists of three staves. The first staff has a treble clef and a key signature of one sharp (F#). It contains a quarter note G4, an eighth note A4, a quarter note B4, and a quarter note C5, followed by a whole rest. The second staff has a treble clef and a key signature of one sharp (F#). It contains a quarter note D5, an eighth note E5, a quarter note F#5, and a quarter note G5, followed by a whole rest. The third staff has a treble clef and a key signature of one sharp (F#). It contains a quarter note A5, an eighth note B5, a quarter note C6, and a quarter note D6, followed by a whole rest. The exercise concludes with a double bar line and the Roman numeral  $\text{XV}_2$ .

②

Exercise 2 consists of eight staves. The first staff has a treble clef and a key signature of one sharp (F#). It contains a quarter note G4, an eighth note A4, a quarter note B4, and a quarter note C5, followed by a whole rest. The second staff has a treble clef and a key signature of one sharp (F#). It contains a quarter note D5, an eighth note E5, a quarter note F#5, and a quarter note G5, followed by a whole rest. The third staff has a treble clef and a key signature of one sharp (F#). It contains a quarter note A5, an eighth note B5, a quarter note C6, and a quarter note D6, followed by a whole rest. The fourth staff has a treble clef and a key signature of one sharp (F#). It contains a quarter note E6, an eighth note F#6, a quarter note G6, and a quarter note A6, followed by a whole rest. The fifth staff has a treble clef and a key signature of one sharp (F#). It contains a quarter note B6, an eighth note C7, a quarter note D7, and a quarter note E7, followed by a whole rest. The sixth staff has a treble clef and a key signature of one sharp (F#). It contains a quarter note F#7, an eighth note G7, a quarter note A7, and a quarter note B7, followed by a whole rest. The seventh staff has a treble clef and a key signature of one sharp (F#). It contains a quarter note C8, an eighth note D8, a quarter note E8, and a quarter note F#8, followed by a whole rest. The eighth staff has a treble clef and a key signature of one sharp (F#). It contains a quarter note G8, an eighth note A8, a quarter note B8, and a quarter note C9, followed by a whole rest. The exercise concludes with a double bar line and the Roman numeral  $\text{XVII}_2$ .

# Rhythmic Vocabulary

①

②

Handwritten musical notation for rhythmic exercises. The page contains 11 staves of music in treble clef, 2/3 time signature. The exercises are numbered 1 and 2. Exercise 1 is marked with a circled 1, and Exercise 2 is marked with a circled 2. The notation includes various rhythmic patterns, rests, and accidentals (sharps, flats, naturals). Some exercises include Roman numerals (I, II, III, IV, V, VI, VII, VIII, IX, X, XI, XII) and letter combinations (H, B, C, D, E, F, G, A, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z) indicating specific rhythmic values or patterns.

## 7. APPENDIX 2: MOVEMENTS USED FOR VARIATION GENERATION

Following is a list of the movements used for variation generation by the system. Each is listed by the composer's name and the key of the piece and can be found on the ensuing pages. They are labelled according to the following codes:

M = one of Telemann's Methodical Sonatas

E = the expert has a written variation of this piece in his thesis

O = a piece of approximately the same style as those written by

Telemann

Handel\_F: Sonata in F Major for Recorder and Continuo, Opus 1 #11, larghetto, George Friedrich Handel (O)

Loeillet\_am: Sonata in a minor for Recorder and Continuo, Opus 1 #1, adagio, Jean Baptiste Loeillet de Gant (O)

Loeillet\_dm: Sonata in d minor for Recorder and Continuo, Opus 1 #2, adagio, Jean Baptiste Loeillet de Gant (O)

Loeillet\_G: Sonata in G Major for Recorder and Continuo, Opus 1 #3, largo, Jean Baptiste Loeillet de Gant (O)

Telemann\_BF: Sonata in B-flat Major for Recorder and Continuo, largo, George Phillip Telemann

Telemann\_C: Sonata in C for Recorder and Continuo from "Der getreue Musik-meister, George Phillip Telemann (E)

Telemann\_gm: Sonata in g minor, Methodical Sonata #1, adagio, George Phillip Telemann (M, E)

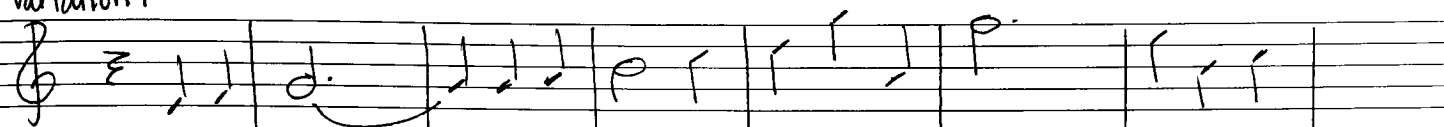
Telemann\_em: Sonata in e minor, Methodical Sonata #3, grave, George  
Phillip Telemann (M)

Telemann\_cm: Sonata in c minor, Methodical Sonata #8, adagio,  
George Phillip Telemann (M)

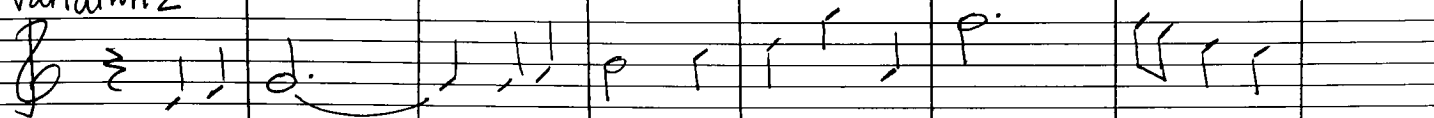
Telemann\_C2: Sonata in C Major, Methodical Sonata #12, andante,  
George Phillip Telemann (M)

# Handel - F

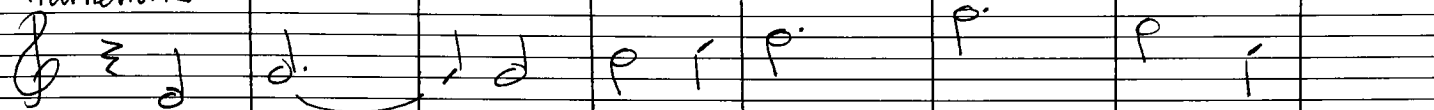
## Variation 1



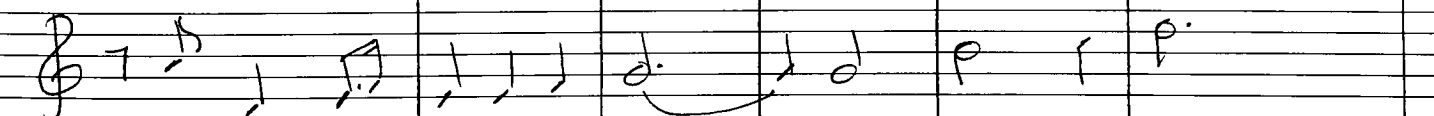
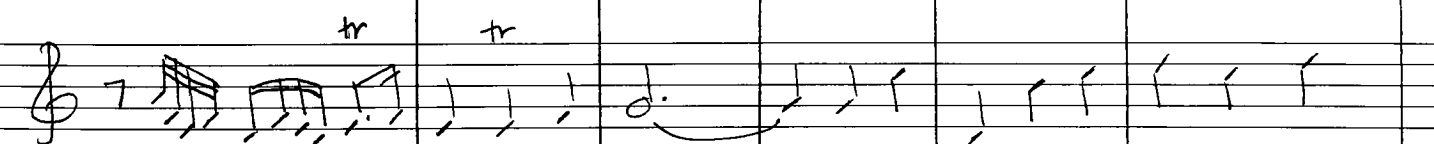
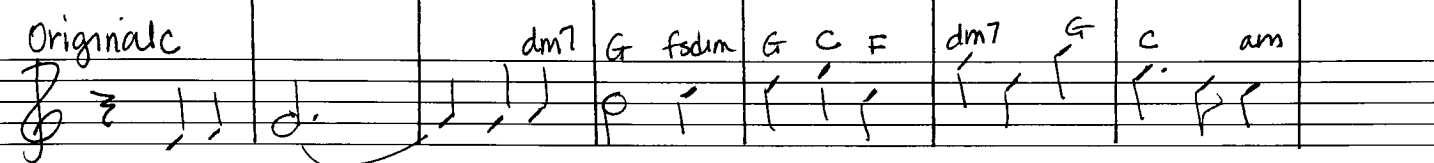
## Variation 2



## Framework



## Original c



Handwritten musical score for a 4-staff system. The first two staves are treble clef, and the last two are bass clef. The music features various notes, rests, and trills. Chord symbols are written above the bass staff.

Chord symbols: D, G, am, D, G, C, dm, G, C.

Handwritten musical score for a 4-staff system. The first two staves are treble clef, and the last two are bass clef. The music features various notes, rests, and trills. Chord symbols are written above the bass staff.

Chord symbols: dm, G7, C, D7, E, am, F, em, dm, am, dm, am, E.



Handwritten musical score for a 4-staff system. The first three staves are treble clef. The fourth staff contains chord symbols: *am*, *G7 C*, *F dm*, *G C bdm*, *C*, *dm*. The music includes triplets and various rhythmic patterns.

Handwritten musical score for a 4-staff system. The first three staves are treble clef. The fourth staff contains chord symbols: *G C*, *am*, *C F dm*, *G7*, *C*, *dm F*. The music includes slurs and various rhythmic patterns.

Handwritten musical score for a 4-staff piece. The first two staves are treble clef, the third is treble clef, and the fourth is bass clef. The music features eighth and sixteenth notes, triplets, and rests. Chord symbols are written below the fourth staff.

Chord symbols: C, am, C7, dm, C, bdim, C, G, C

Handwritten musical score for a 4-staff piece. The first three staves are treble clef, and the fourth is bass clef. The music features eighth and sixteenth notes, triplets, and rests. Chord symbols are written below the fourth staff.

Chord symbols: am, f#dim, G

```
[
[[c],[q],[r]],[[c],[q],[e]],[[c],[q],[f]],[[c],[dh],[g]],[[c],[tq],[g]],
[[c],[q],[g]],[[dm7],[q],[a]],[[g],[h],[b]],[[fsdim],[q],[cl]],
[[g],[q],[dl]],[[c],[q],[el]],[[f],[q],[cl]],[[dm7],[q],[fl]],
[[dm7],[q],[dl]],[[g],[q],[gl]],[[c],[dq],[el]],[[c],[e],[dl]],
[[am],[q],[cl]],[[dm7],[e],[r]],[[dm7],[e],[a]],[[g],[q],[d]],[[g7],[de],[e]],
[[g7],[s],[f]],[[c],[q],[e]],[[c],[q],[e]],[[c],[q],[f]],[[c],[dh],[g]],
[[c],[tq],[g]],[[c],[q],[g]],[[dm7],[q],[a]],[[g],[h],[b]],[[d7],[q],[cl]],
[[g],[q],[dl]],[[c],[q],[el]],[[am],[q],[cl]],[[d],[h],[a]],[[g],[q],[b]],
[[am],[q],[cl]],[[d],[dq],[a]],[[d],[e],[g]],[[g],[dh],[g]],
[[c],[q],[r]],[[c],[q],[el]],[[dm],[q],[dl]],[[g],[de],[b]],[[g],[s],[a]],
[[g],[q],[g]],[[g],[q],[r]],[[g],[q],[r]],[[g],[q],[gl]],[[c],[q],[el]],
[[dm],[dq],[fl]],[[dm],[e],[el]],[[g7],[e],[fl]],[[g7],[e],[dl]],
[[c],[dq],[el]],[[c],[e],[dl]],[[c],[e],[el]],[[c],[e],[cl]],[[d7],[dq],[dl]],
[[e],[e],[cl]],[[e],[e],[dl]],[[e],[e],[b]],[[am],[q],[cl]],[[f],[q],[fl]],
[[em],[q],[el]],[[dm],[q],[dl]],[[am],[q],[cl]],[[dm],[q],[b]],
[[am],[e],[cl]],[[am],[e],[dl]],[[am],[dq],[b]],[[e],[e],[a]],
[[am],[q],[a]],[[dm],[de],[fl]],[[dm],[s],[el]],[[dm],[q],[fl]],
[[g7],[q],[g]],[[c],[de],[el]],[[c],[s],[dl]],[[c],[q],[el]],
[[f],[e],[f]],[[f],[e],[el]],[[dm],[de],[dl]],[[dm],[s],[cl]],
[[dm],[de],[cl]],[[dm],[s],[b]],[[g],[q],[b]],[[c],[q],[g]],[[bdim],[q],[g]],
[[c],[q],[g]],[[c],[e],[a]],[[c],[e],[g]],[[dm],[e],[f]],[[dm],[e],[e]],
[[g],[q],[d]],[[c],[q],[el]],[[c],[q],[el]],[[am],[dh],[el]],[[c],[tde],[el]],
[[c],[s],[dl]],[[f],[q],[cl]],[[dm],[q],[fl]],[[g7],[tdq],[fl]],[[g7],[e],[dl]],
[[c],[q],[el]],[[dm],[e],[dl]],[[dm],[e],[cl]],[[f],[dq],[b]],
[[f],[e],[cl]],[[c],[q],[e]],[[c],[q],[e]],[[c],[q],[e]],[[am],[q],[cl]],
[[c7],[q],[e]],[[dm],[q],[a]],[[c],[dq],[g]],[[c],[e],[a]],[[bdim],[q],[f]],
[[c],[e],[e]],[[c],[e],[d]],[[g],[dq],[d]],[[g],[e],[c]],[[c],[q],[c]],
[[c],[q],[cl]],[[c],[de],[b]],[[c],[s],[a]],[[c],[dh],[g]],[[c],[q],[r]],
[[c],[q],[el]],[[c],[e],[dl]],[[c],[e],[g]],[[am],[e],[cl]],
[[am],[e],[e]],[[fsdim],[dq],[fs]],[[fsdim],[e],[g]],[[g],[dh],[g]]
].
```

```

[
[[c],[q],[r]],[[c],[h],[e]],[[c],[dh],[g]],[[c],[tq],[g]],
[[c],[h],[g]],[[g],[h],[b]],[[fsdim],[q],[c1]],
[[g],[dh],[d1]],[[dm7],[dh],[f1]],
[[c],[h],[e1]],[[am],[q],[c1]],[[dm7],[e],[r]],[[dm7],[e],[a]],
[[g],[q],[d]],[[g7],[de],[e]],[[g7],[s],[f]],
[[c],[q],[e]],[[c],[q],[e]],[[c],[q],[f]],[[c],[dh],[g]],
[[c],[tq],[g]],[[c],[h],[g]],[[g],[h],[b]],[[d7],[q],[c1]],
[[g],[dh],[d1]],[[d],[h],[a]],[[g],[q],[b]],
[[am],[q],[c1]],[[d],[dq],[a]],[[d],[e],[g]],[[g],[dh],[g]],
[[c],[q],[r]],[[c],[q],[e1]],[[dm],[q],[d1]],[[g],[q],[b]],
[[g],[q],[g]],[[g],[q],[r]],[[g],[q],[r]],[[g],[q],[g1]],[[g],[q],[e1]],
[[dm],[h],[f1]],[[g],[e],[f1]],[[g7],[e],[d1]],
[[c],[h],[e1]],[[c],[e],[e1]],[[c],[e],[c1]],[[d7],[h],[d1]],
[[e],[e],[d1]],[[e],[e],[b]],[[am],[q],[c1]],[[f],[q],[f1]],
[[em],[q],[e1]],[[dm],[q],[d1]],[[am],[q],[c1]],[[dm],[q],[b]],
[[am],[e],[c1]],[[am],[e],[d1]],[[am],[dq],[b]],[[e],[e],[a]],
[[am],[q],[a]],[[dm],[q],[f1]],[[dm],[q],[f1]],
[[g7],[q],[g]],[[c],[q],[e1]],[[c],[q],[e1]],
[[f],[q],[f]],[[dm],[q],[d1]],
[[dm],[q],[c1]],[[g],[q],[b]],[[c],[q],[g]],[[bdim],[q],[g]],
[[c],[q],[g]],[[c],[e],[a]],[[c],[e],[g]],[[dm],[e],[f]],[[dm],[e],[e]],
[[g],[q],[d]],[[c],[q],[e1]],[[c],[q],[e1]],[[am],[dh],[e1]],[[c],[tq],[e1]],
[[f],[q],[c1]],[[dm],[q],[f1]],[[g7],[tdq],[f1]],[[g7],[e],[d1]],
[[c],[q],[e1]],[[dm],[q],[d1]],[[f],[dq],[b]],
[[f],[e],[c1]],[[c],[q],[e]],[[c],[q],[e]],[[c],[q],[e]],[[am],[q],[c1]],
[[c7],[q],[e]],[[dm],[q],[a]],[[c],[h],[g]],[[bdim],[q],[f]],
[[c],[q],[e]],[[g],[dq],[d]],[[g],[e],[c]],[[c],[q],[c]],
[[c],[h],[c1]],[[c],[dh],[g]],[[c],[q],[r]],
[[c],[q],[e1]],[[c],[e],[d1]],[[c],[e],[g]],[[am],[e],[c1]],
[[am],[e],[e]],[[fsdim],[dq],[fs]],[[fsdim],[e],[g]],[[g],[dh],[g]]
].

```

?- v(Variation).

```
Variation = [[ [c], [q], [r] ], [ [c, c], [q, q, dh], [e, f, g] ],
               [ [c], [tq], [g] ], [ [c, g], [q, q, h], [g, a, b] ],
               [ [fsdim], [q], [c1] ], [ [g, dm7], [q, q, q, dh], [d1, g1, a,
               f1] ],
               [ [c, am], [q, q, q], [e1, b, c1] ], [ [dm7], [e], [r] ],
               [ [dm7, g], [s, s, t, t, t], [a, f, d, d, d] ],
               [ [g7, g7], [de, s], [e, trill, f] ], [ [c], [q], [e] ],
               [ [c, c], [e, e, q], [e, trill, f] ], [ [c], [dh], [g] ],
               [ [c], [tq], [g] ], [ [c, g], [q, q, q, q], [g, c1, d, b] ],
               [ [d7], [q], [c1] ], [ [g, d], [e, e, e, e, q, q, q],
               [c1, e1, d1, c1, d1, b, a] ],
               [ [g, am], [e, e, q], [b, trill, c1] ], [ [d], [dq], [a] ],
               [ [d], [e], [g] ], [ [g], [dh], [g] ], [ [c], [q], [r] ],
               [ [c], [q], [e1] ], [ [dm, g, g], [s, s, s, s, s, s, s, s, q],
               [d1, e1, f1, d1, b, c1, d1, b, g] ],
               [ [g], [q], [r] ], [ [g], [q], [r] ], [ [g, g], [e, e, q],
               [g1, d1, e1] ],
               [ [dm], [h], [f1] ], [ [g, g7], [s, s, e], [f1, c1, d1] ],
               [ [c], [h], [e1] ], [ [c, c], [s, s, e], [e1, b, c1] ],
               [ [d7], [h], [d1] ], [ [e, e], [s, s, s, s], [e1, d1, c1, b] ],
               [ [am, f], [t, t, t, q], [c1, c2, a1, f1] ], [ [em], [q], [e1] ],
               [ [dm], [q], [d1] ], [ [am], [q], [c1] ],
               [ [dm, am], [e, e, e], [b, trill, c1] ],
               [ [am, am], [e, s, s, q], [e1, d1, c1, b] ], [ [e], [e], [a] ],
               [ [am, dm], [t, t, t, q], [c1, b, a, f1] ],
               [ [dm, g7], [t, t, t, q], [f1, e1, f1, g] ], [ [c], [q], [e1] ],
               [ [c, f], [t, t, t, q], [e1, d1, e1, f] ], [ [dm], [q], [d1] ],
               [ [dm], [q], [c1] ], [ [g, c], [s, s, s, s, e, e], [b, b, a, g, f, g] ],
               [ [bdim], [q], [g] ], [ [c], [q], [g] ],
               [ [c, c, dm, dm, g], [e, s, s, e, s, s, q], [a, b, g, f, g, e, d] ],
               [ [c], [q], [e1] ], [ [c], [q], [e1] ], [ [am], [dh], [e1] ],
               [ [c, f], [ts, s, s, s, e, e], [e1, e1, d1, c1, b, c1] ],
               [ [dm], [q], [f1] ], [ [g7, g7], [e, e, e, e], [f1, d1, c1, d1] ],
               [ [c], [q], [e1] ], [ [dm, f], [s, e, s, dq], [d1, b, a, b] ],
               [ [f, c], [e, s, s, e], [c1, g, f, e] ], [ [c], [q], [e] ],
               [ [c, am], [t, t, t, q], [g, fs, e, c1] ],
               [ [c7, dm], [t, t, t, q], [e, e1, c1, a] ],
               [ [c, bdim, c, g, g], [q, q, q, e, e, dq, e], [g, a, f, e, f, d, c] ],
               [ [c, c], [t, t, t, h], [c, e, g, c1] ], [ [c], [dh], [g] ],
               [ [c], [q], [r] ], [ [c], [q], [e1] ],
               [ [c, c], [th, th, th, th, e], [d1, b, g, g, g] ],
               [ [am, am], [s, s, s, s], [c1, g, f, e] ],
               [ [fsdim, fsdim], [dq, e], [fs, trill, g] ], [ [g], [dh], [g] ]]
```

?-

```

Variation = [[c],[q],[r]],[[c,c],[q,q,dh],[e,f,g]],
[[c],[tq],[g]],[[c,g],[q,q,h],[g,a,b]],
[[fsdim],[q],[cl]],[[g,dm7],[q,q,q,dh],[dl,g1,a,
f1]],
[[c,am],[e,e,q,q],[f1,e1,dl,cl]],[[dm7],[e],[r]],
[[dm7,g],[th,th,th,th,s,s,s,s],[a,f,e,f,d,f,e,d]],
[[g7,g7],[de,s],[e,trill,f]],[[c],[q],[e]],
[[c,c],[e,e,q],[e,trill,f]],[[c],[dh],[g]],
[[c],[tq],[g]],[[c,g],[q,q,q,q],[g,cl,d,b]],
[[d7],[q],[cl]],[[g,d],[q,q,q,h],[dl,cl,dl,a]],
[[g,am],[e,e,q],[b,trill,cl]],[[d],[dq],[a]],
[[d],[e],[g]],[[g],[dh],[g]],[[c],[q],[r]],
[[c],[q],[e1]],[[dm,g,g],[e,e,s,e,s,q],
cl,dl,cl,b,a,g]],[[g],[q],[r]],
[[g],[q],[r]],[[g,g],[t,t,t,q],[a1,g1,f1,e1]],
[[dm],[h],[f1]],[[g,g7],[s,s,s,s],[g1,f1,e1,dl]],
[[c],[h],[e1]],[[c,c],[s,s,s,s],[f1,e1,dl,cl]],
[[d7],[h],[dl]],[[e,e],[s,s,e],[dl,a,b]],
[[am,f],[t,t,t,q],[cl,c2,a1,f1]],[[em],[q],[e1]],
[[dm],[q],[dl]],[[am],[q],[cl]],
[[dm,am],[e,e,e],[b,trill,cl]],
[[am,am],[e,e,q],[dl,a,b]],[[e],[e],[a]],
[[am,dm],[e,s,th,th,q],[a,cl,b,a,f1]],
[[dm,g7],[t,t,t,q],[f1,e1,f1,g]],[[c],[q],[e1]],
[[c,f],[t,t,t,q],[e1,dl,e1,f]],[[dm],[q],[dl]],
[[dm],[q],[cl]],[[g,c],[t,t,t,q],[b,g,f,g]],
[[bdim],[q],[g]],[[c],[q],[g]],
[[c,c,dm,dm,g],[e,s,s,e,s,s,q],[a,g,f,e,d,e,d]],
[[c],[q],[e1]],[[c],[q],[e1]],[[am],[dh],[e1]],
[[c,f],[e,e,e,e],[e1,cl,b,cl]],[[dm],[q],[f1]],
[[g7,g7],[tdq,e],[f1,f1,e1,dl,cl,dl]],
[[c],[q],[e1]],[[dm,f],[q,s,s,s,s,e],[dl,dl,cl,b,
a,b]],
[[f,c],[e,s,s,e],[cl,g,f,e]],[[c],[q],[e]],
[[c,am],[e,s,th,th,q],[e,g,f,e,cl]],
[[c7,dm],[t,t,t,q],[e,e1,cl,a]],
[[c,bdim,c,g,g],[q,q,q,e,e,dq,e],[g,f,e,d,c,d,c]],
[[c,c],[t,t,t,h],[ds,c,b,cl]],[[c],[dh],[g]],
[[c],[q],[r]],[[c],[q],[e1]],
[[c,c],[th,th,th,th,th,th,th,th],[dl,b,a,b,g,b,a,
g]],
[[am,am],[s,s,s,s],[cl,g,f,e]],
[[fsdim,fsdim],[q,e,e],[fs,trill,g]],[[g],[dh],
[g]]]

```

?-

Handwritten musical notation for the first system, showing four staves. The first staff is labeled "Variation 1" and the second "Variation 2". The third staff is labeled "Framework". The fourth staff is labeled "Original" and contains the following notes and chords: am, E, am, em, dm, bm7, E, E7.

Handwritten musical notation for the second system, showing four staves. The first three staves contain melodic lines with various accidentals and trills. The fourth staff contains the following notes and chords: am, A, dm, D7, G, E7, am, dm, E, am, em.

Handwritten musical score for the first system, consisting of four staves. The notation includes various musical symbols such as notes, rests, and accidentals. The fourth staff includes chord symbols: *dm*, *E*, *C*, *G*, *C*, *G*, *am*, and *D7*.

Handwritten musical score for the second system, consisting of four staves. The notation includes various musical symbols such as notes, rests, and accidentals. The fourth staff includes chord symbols: *G*, *G7*, *C*, *F*, *G*, and *C*.



Handwritten musical score for the first system, consisting of four staves. The notation includes various chords, triplets, and accidentals.

- Staff 1:** Treble clef. Measures 1-4: Chords with sharps and naturals, including a triplet of eighth notes in measure 4.
- Staff 2:** Treble clef. Measures 1-4: Chords with sharps and naturals, including a triplet of eighth notes in measure 4.
- Staff 3:** Treble clef. Measures 1-4: Chords with sharps and naturals, including a triplet of eighth notes in measure 4.
- Staff 4:** Treble clef. Measures 1-4: Chords with sharps and naturals, including a triplet of eighth notes in measure 4.

Handwritten musical score for the second system, consisting of four staves. The notation includes various chords, triplets, and accidentals.

- Staff 1:** Treble clef. Measures 1-4: Chords with sharps and naturals, including a triplet of eighth notes in measure 4.
- Staff 2:** Treble clef. Measures 1-4: Chords with sharps and naturals, including a triplet of eighth notes in measure 4.
- Staff 3:** Treble clef. Measures 1-4: Chords with sharps and naturals, including a triplet of eighth notes in measure 4.
- Staff 4:** Treble clef. Measures 1-4: Chords with sharps and naturals, including a triplet of eighth notes in measure 4.

Handwritten musical notation for a 4-staff system. The first three staves are treble clef, and the fourth is bass clef. The first staff has a trill (tr) and triplets (3). The second staff has a trill (tr) and a triplet (3). The third staff has a trill (tr). The fourth staff has chords labeled am, dm, G7, C, F, bm7, and bm.

Handwritten musical notation for a 4-staff system. The first three staves are treble clef, and the fourth is bass clef. The first staff has a sharp sign (#) and a trill (tr). The second staff has a sharp sign (#) and a trill (tr). The third staff has a sharp sign (#) and a trill (tr). The fourth staff has chords labeled E7, am, dm7, E7, am, em, am, dm, and bdim.

Handwritten musical score for guitar, consisting of four staves. The notation includes triplets, accidentals, and chord symbols.

**Staff 1:** Treble clef. First measure: triplet of eighth notes (F#, G, A), triplet of eighth notes (F#, G, A), half note (B). Second measure: quarter note (F#), half note (G), half note (A). Third measure: quarter note (F#), half note (G), half note (A). Fourth measure: quarter note (F#), half note (G), half note (A).

**Staff 2:** Treble clef. First measure: quarter note (F#), half note (G), half note (A). Second measure: quarter note (F#), half note (G), half note (A). Third measure: quarter note (F#), half note (G), half note (A). Fourth measure: quarter note (F#), half note (G), half note (A).

**Staff 3:** Treble clef. First measure: quarter note (F#), half note (G), half note (A). Second measure: quarter note (F#), half note (G), half note (A). Third measure: quarter note (F#), half note (G), half note (A). Fourth measure: quarter note (F#), half note (G), half note (A).

**Staff 4:** Treble clef. First measure: quarter note (F#), half note (G), half note (A). Second measure: quarter note (F#), half note (G), half note (A). Third measure: quarter note (F#), half note (G), half note (A). Fourth measure: quarter note (F#), half note (G), half note (A).

```

[
[[am],[q],[a]],[[e],[q],[e1]],[[e],[q],[e1]],
[[am],[q],[c1]],[[am],[q],[a]],[[em],[q],[e1]],
[[dm],[q],[a1]],[[dm],[e],[b1]],[[dm],[e],[c2]],[[bm7],[e],[b1]],[[bm7],[e],[a1]],
[[e],[h],[gs1]],[[e7],[q],[e1]],
[[am],[tq],[e1]],[[a],[e],[a1]],[[a],[e],[g1]],[[a],[e],[f1]],[[a],[e],[e1]],
[[dm],[q],[f1]],[[d7],[h],[fs1]],
[[g],[q],[g1]],[[e7],[h],[gs1]],
[[am],[e],[a1]],[[dm],[e],[b1]],[[e],[dq],[b1]],[[e],[e],[a1]],
[[am],[h],[a1]],[[em],[q],[e1]],
[[em],[tq],[e1]],[[dm],[dq],[d1]],[[dm],[e],[e1]],
[[e],[dh],[e1]],
[[c],[q],[c1]],[[g],[q],[g1]],[[g],[q],[g1]],[[c],[q],[e1]],[[c],[q],[c1]],
[[g],[q],[g1]],[[am],[q],[c2]],[[d7],[dq],[c2]],[[d7],[e],[d2]],
[[g],[h],[b1]],[[g7],[q],[g1]],[[c],[tq],[g1]],[[c],[h],[c1]],
[[f],[dq],[a1]],[[f],[e],[g1]],[[f],[e],[f1]],[[f],[e],[e1]],
[[g],[e],[d1]],[[g],[e],[c1]],[[g],[dq],[b]],[[g],[e],[c1]],
[[c],[q],[c1]],[[c],[q],[e1]],[[c],[q],[c1]],[[g],[q],[d1]],[[g],[h],[g1]],
[[d7],[tq],[g1]],[[d7],[dq],[fs1]],[[d],[e],[g1]],[[g],[dh],[g1]],
[[g],[q],[g1]],[[d],[q],[d2]],[[d],[q],[d2]],[[g],[q],[b1]],[[g],[h],[g1]],
[[c],[tq],[g1]],[[c],[q],[c2]],[[c],[q],[c2]],[[dsdim],[q],[a1]],
[[dsdim],[h],[fs1]],[[b],[tq],[fs1]],[[b],[q],[b1]],[[b],[q],[b1]],
[[e],[q],[gs1]],[[e],[h],[e1]],[[am],[tq],[e1]],[[am],[q],[e1]],
[[am],[e],[d1]],[[am],[e],[e1]],[[dm],[q],[f1]],[[dm],[q],[d1]],
[[dm],[q],[c1]],[[g7],[q],[b]],[[g7],[q],[d1]],[[g7],[e],[c1]],
[[g7],[e],[d1]],[[c],[q],[e1]],[[c],[q],[c1]],[[c],[q],[b]],
[[f],[q],[a]],[[f],[q],[c1]],[[f],[e],[b]],[[f],[e],[c1]],
[[bm7],[q],[d1]],[[bm],[q],[b]],[[bm],[q],[a]],[[e7],[h],[gs1]],
[[am],[e],[a1]],[[am],[e],[e1]],[[dm7],[e],[d1]],[[dm7],[e],[c1]],
[[e7],[dq],[b]],[[e7],[e],[a]],[[am],[dq],[a]],[[am],[e],[b]],
[[am],[e],[c1]],[[am],[e],[d1]],[[em],[e],[e1]],[[em],[e],[ds1]],
[[em],[e],[e1]],[[em],[e],[fs1]],[[em],[e],[g1]],[[em],[e],[e1]],
[[am],[e],[a1]],[[am],[e],[gs1]],[[dm],[e],[a1]],[[dm],[e],[c2]],
[[bdim],[e],[b1]],[[bdim],[e],[a1]],[[e],[q],[gs1]],[[e7],[q],[e1]],
[[am],[q],[a1]],[[bdim],[tq],[a1]],[[e],[dq],[gs1]],[[e],[e],[a1]],
[[am],[dh],[a1]]
].

```

```
[
[[am],[q],[a]],[[e],[q],[e1]],[[e],[q],[e1]],
[[am],[q],[c1]],[[am],[q],[a]],[[am],[q],[e1]],
[[dm],[q],[a1]],[[dm],[h],[b1]],[[e],[h],[gs1]],[[e],[q],[e1]],
[[am],[q],[e1]],[[a],[dq],[a1]],[[a],[e],[e1]],
[[dm],[q],[f1]],[[d],[h],[fs1]],[[g],[q],[g1]],
[[e],[h],[gs1]],[[am],[q],[a1]],[[e],[dq],[b1]],[[e],[e],[a1]],
[[am],[h],[a1]],
[[em],[q],[e1]],[[dm],[tq],[e1]],[[dm],[dq],[d1]],[[dm],[e],[e1]],
[[e],[dh],[e1]],
[[c],[q],[c1]],[[g],[q],[g1]],[[g],[q],[g1]],
[[c],[q],[e1]],[[c],[q],[c1]],
[[g],[q],[g1]],[[am],[q],[c2]],[[am],[h],[c2]],
[[g],[h],[b1]],[[g7],[q],[g1]],[[c],[tq],[g1]],[[c],[h],[c1]],
[[f],[h],[a1]],[[g],[q],[f1]],[[g],[q],[d1]],[[g],[dq],[b]],[[g],[e],[c1]],
[[c],[dh],[c1]],[[g],[q],[d1]],[[g],[h],[g1]],
[[d7],[tq],[g1]],[[d7],[dq],[fs1]],[[d7],[e],[g1]],[[g],[dh],[g1]],
[[g],[q],[g1]],[[d],[q],[d2]],[[d],[q],[d2]],[[g],[q],[b1]],[[g],[h],[g1]],
[[c],[tq],[g1]],[[c],[q],[c2]],[[c],[q],[c2]],[[dsdim],[q],[a1]],
[[dsdim],[h],[fs1]],[[b],[tq],[fs1]],[[b],[q],[b1]],[[b],[q],[b1]],
[[e],[q],[gs1]],[[e],[h],[e1]],[[am],[tq],[e1]],[[am],[h],[e1]],
[[dm],[q],[f1]],[[dm],[q],[d1]],
[[dm],[q],[c1]],[[g7],[q],[b]],[[g7],[h],[d1]],
[[c],[q],[e1]],[[c],[q],[c1]],[[c],[q],[b]],
[[f],[q],[a]],[[f],[h],[c1]],
[[bm7],[q],[d1]],[[bm],[q],[b]],[[bm],[q],[a]],[[e7],[h],[gs1]],
[[am],[e],[a1]],[[am],[e],[e1]],[[dm7],[e],[d1]],[[dm7],[e],[c1]],
[[e7],[dq],[b]],[[e7],[e],[a]],[[am],[dh],[a]],[[em],[q],[e1]],[[em],[q],[e1]],
[[em],[e],[g1]],[[em],[e],[e1]],[[bdim],[h],[a1]],
[[bdim],[q],[b1]],[[e],[q],[gs1]],[[e7],[q],[e1]],
[[am],[q],[a1]],[[bdim],[tq],[a1]],[[e],[dq],[gs1]],[[e],[e],[a1]],
[[am],[dh],[a1]]
].
```

?- v(Variation).

<Undefined Goal>:

| | | | | (12:168) alreadytried(4). :

```
Variation = [[am,e],[t,t,t,q],[a,c1,f1,e1]],
              [[e,am,am],[s,s,s,s,s,s,s,q],[e1,f1,gs1,e1,c1,
              d1,e1,c1,a]],
              [[am,dm],[t,t,t,q],[e1,e2,c2,a1]],
              [[dm,e,e],[e,e,e,e,e,e,e,q],[b1,c2,d2,b1,gs1,a1,
              b1,gs1,e1]],
              [[am,a],[s,e,s,dq],[e1,e2,cs2,a1]],
              [[a,dm],[s,s,q],[e1,tr11,f1]],
              [[d,g],[q,q,q],[fs1,tr11,g1]],
              [[e,am],[q,q,q],[gs1,tr11,a1],[[e],[dq],[b1]],
              [[e],[e],[a1]],[[am,em],[e,e,q,q],[a1,e1,ds1,e1]],
              [[dm],[tq],[e1]],[[dm],[dq],[d1]],[[dm],[e],[e1]],
              [[e,c],[e,e,e,e,q,q],[e1,e1,d1,c1,b,c1]],
              [[g],[q],[g1]],[[g,c,c],[e,e,s,e,s,q],
              [f1,g1,f1,e1,d1,c1]],
              [[g,am],[s,s,s,s,e,e],[g1,c2,d2,c2,b1,c2]],
              [[am],[h],[c2]],[[g,g7],[q,s,s,s,q],
              [b1,b1,a1,g1,f1,g1]],
              [[c,c],[s,s,s,s,s,s,e,q],[g1,e1,d1,e1,c1,e1,d1,c1]],
              [[f,g,g],[q,q,e,e,e,e],[g1,a1,g1,f1,e1,d1]],
              [[g,g],[dq,e],[b,tr11,c1]],[[c],[dh],[c1]],
              [[g,g],[q,s,s,s,s,q],[d1,g1,a1,g1,fs1,g1]],
              [[d7],[tq],[g1]],[[d7,d7],[dq,e],[fs1,tr11,g1]],
              [[g],[dh],[g1]],[[g],[q],[g1]],[[d],[q],[d2]],
              [[d,g,g],[er,e,tt,t,t,t,t,q],[d2,d2,e2,c2,b1,c2,
              a1,g1]],
              [[c],[tq],[g1]],[[c],[q],[c2]],
              [[c,dsdim,dsdim],[er,e,tt,t,t,t,t,t,q],
              [c2,c2,ds2,b1,a1,b1,g1,fs1]],[[b],[tq],[fs1]],
              [[b],[q],[b1]],[[b,e,e],[er,e,tt,t,t,t,t,t,q],
              [b1,b1,c2,a1,gs1,a1,fs1,e1]],
              [[am],[tq],[e1]],[[am,dm],[q,q,q],[e1,tr11,f1]],
              [[dm],[q],[d1]],[[dm],[q],[c1]],
              [[g7,g7],[t,t,t,h],[b,e1,f,d1]],
              [[c,c],[t,t,t,q],[e1,c1,b,c1]],[[c],[q],[b]],
              [[f,f],[t,t,t,h],[a,d1,e,c1]],
              [[bm7,bm],[t,t,t,q],[d1,b,a,b]],
              [[bm,e7],[t,t,t,h],[a,c1,e1,gs1]],
              [[am,am],[s,s,s,s],[a1,e1,ds1,e1]],[[dm7],[e],
              [d1]],
              [[dm7],[e],[c1]],[[e7],[dq],[b]],[[e7],[e],[a]],
              [[am,em],[q,q,q,q],[a,c1,f1,e1]],
              [[em,em],[e,e,e],[e1,f1,g1]],
              [[em,bdim],[e,s,s,e,q],[e1,g1,f1,e1,a1]],
              [[bdim,e,e7],[er,e,tst,st,st,st,q],
              [b1,b1,c2,a1,gs1,a1,f1,e1]],[[am],[q],[a1]],
              [[bdim],[tq],[a1]],[[e,e],[dq,e],[gs1,tr11,a1]],
              [[am],[dh],[a1]]]
```

?-

```

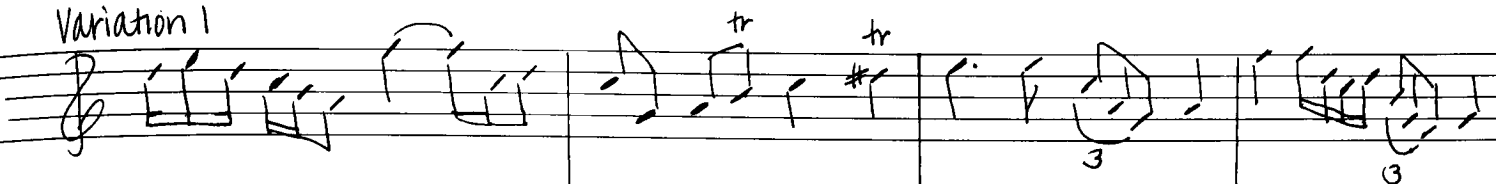
Variation = [[am,e],[s,s,s,s,q],[a,b,cl,d1,e1]],
              [[e,am,am],[s,s,s,s,s,s,s,q],[e1,f1,gs1,e1,cl,
              d1,e1,cl,a]],
              [[am,dm],[t,t,t,q],[e1,e2,c2,a1]],
              [[dm,e,e],[e,e,e,e,e,e,e,q],[b1,c2,d2,b1,gs1,a1,
              b1,gs1,e1]],
              [[am,a],[s,e,s,dq],[e1,e2,cs2,a1]],
              [[a,dm],[s,s,q],[e1,trill,f1]],
              [[d,g],[q,q,q],[fs1,trill,g1]],
              [[e,am],[q,q,q],[gs1,trill,a1]],[[e],[dq],[b1]],
              [[e],[e],[a1]],[[am,em],[e,e,q,q],[a1,e1,ds1,e1]],
              [[dm],[tq],[e1]],[[dm],[dq],[d1]],[[dm],[e],[e1]],
              [[e,c],[q,q,q,q],[f1,e1,d1,cl]],[[g],[q],[g1]],
              [[g,c,c],[er,e,tst,st,st,st,st,q],
              [g1,g1,a1,f1,e1,f1,d1,cl]],
              [[g,am],[s,s,s,s,q],[g1,b1,a1,g1,c2]],
              [[am],[h],[c2]],[[g,g7],[e,e,q,q],[c2,b1,a1,g1]],
              [[c,c],[s,s,s,s,s,s,e,q],[g1,e1,d1,e1,cl,e1,d1,cl]],
              [[f,g,g],[er,e,tt,t,t,t,t,q],[a1,a1,b1,g1,f1,g1,
              e1,d1]],
              [[g,g],[q,e,e],[b,trill,cl]],[[c],[dh],[c1]],
              [[g,g],[s,s,s,s,h],[d1,f1,e1,d1,g1]],
              [[d7],[tq],[g1]],[[d7,d7],[q,e,e],[fs1,trill,g1]],
              [[g],[dh],[g1]],[[g],[q],[g1]],[[d],[q],[d2]],
              [[d,g,g],[e,e,s,e,s,h],[c2,d2,c2,b1,a1,g1]],
              [[c],[tq],[g1]],[[c],[q],[c2]],
              [[c,dsdim,dsdim],[e,e,s,e,s,h],[b1,c2,b1,a1,g1,
              fs1]],
              [[b],[tq],[fs1]],[[b],[q],[b1]],
              [[b,e,e],[e,e,s,e,s,h],[a1,b1,a1,gs1,fs1,e1]],
              [[am],[tq],[e1]],[[am,dm],[q,q,q],[e1,trill,f1]],
              [[dm],[q],[d1]],[[dm],[q],[c1]],
              [[g7,g7],[q,q,q],[b,cl,d1]],[[c,c],[e,e,q],
              [e1,b,cl]],
              [[c],[q],[b]],[[f,f],[q,q,q],[a,b,cl]],
              [[bm7,bm],[e,e,q],[d1,a,b]],
              [[bm,e7],[t,t,t,h],[a,cl,e1,gs1]],
              [[am,am],[s,s,s,s],[a1,e1,ds1,e1]],[[dm7],[e],
              [d1]],
              [[dm7],[e],[c1]],[[e7],[dq],[b]],[[e7],[e],[a]],
              [[am,em],[dq,e,e,e,q],[a,b,cl,d1,e1]],
              [[em,em],[s,s,e,e],[e1,a1,b,g1]],
              [[em,bdim],[e,s,s,s,s,q],[e1,a1,b1,a1,g1,a1]],
              [[bdim,e,e7],[e,e,s,e,s,q],[a1,b1,a1,gs1,f1,e1]],
              [[am],[q],[a1]],[[bdim],[tq],[a1]],
              [[e,e],[q,e,e],[gs1,trill,a1]],[[am],[dh],[a1]]]

```

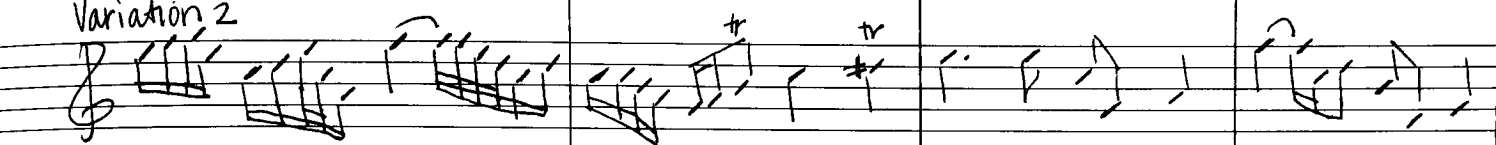
?-

# Loeillet-dm

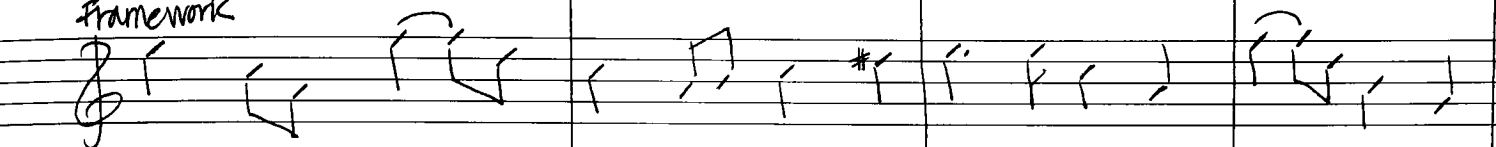
## Variation 1



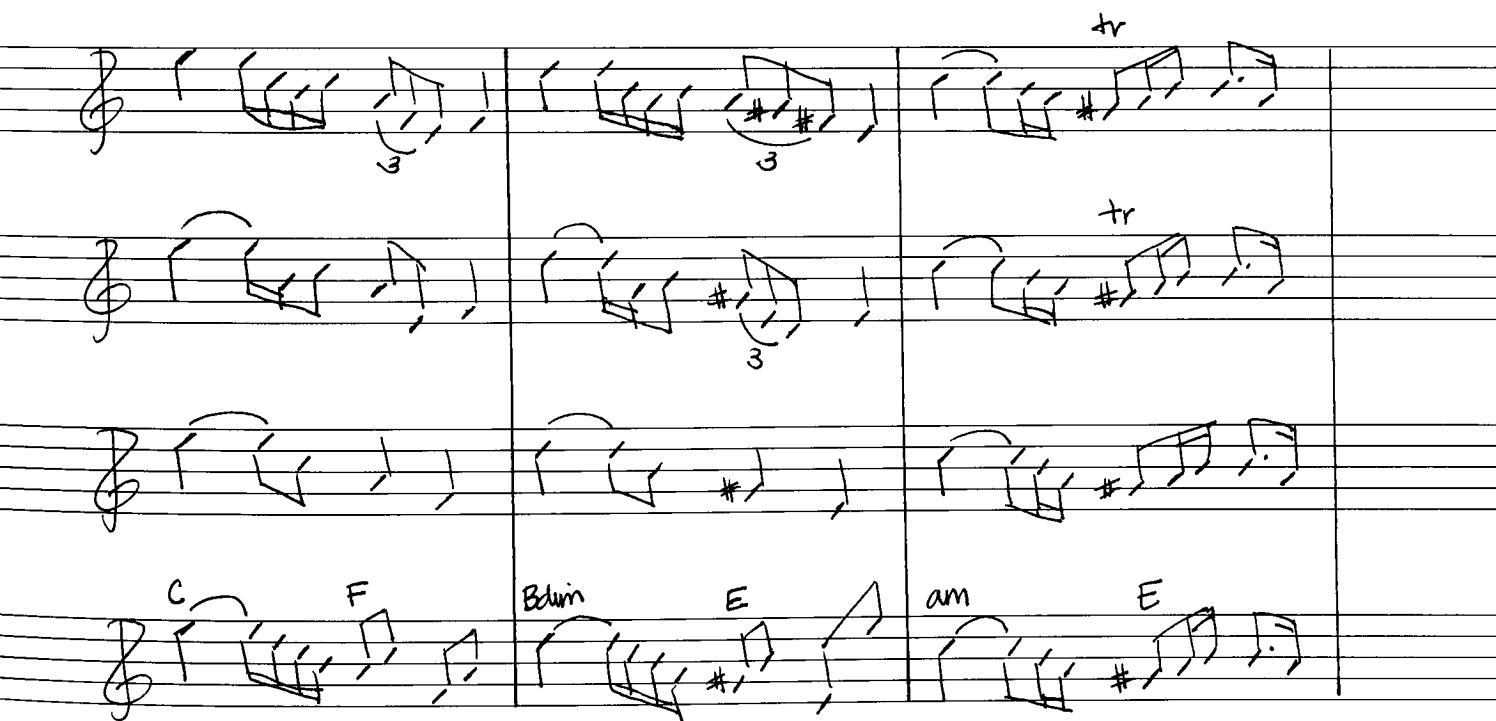
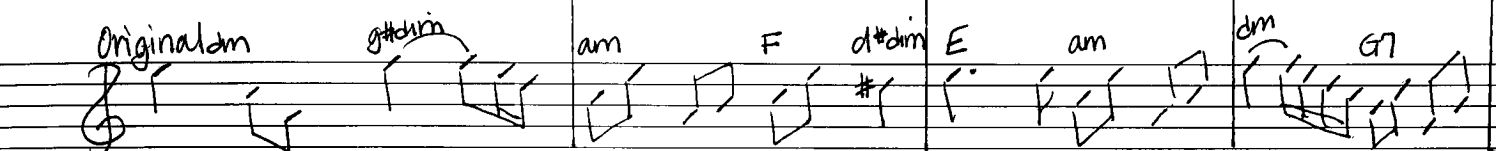
## Variation 2



## Framework



## Original dm





Handwritten musical notation for the first system, consisting of four staves. The first three staves contain melodic lines with triplets and slurs. The fourth staff contains a bass line with chords labeled 'am', 'E', 'am', 'F', 'bdim', and 'em'.

Handwritten musical notation for the second system, consisting of four staves. The first three staves contain melodic lines with triplets and slurs. The fourth staff contains a bass line with chords labeled 'am', 'dm', 'G7', 'C', 'F', 'dm', 'G', and 'C'.

Handwritten musical notation for the first system, consisting of four staves. The first three staves are treble clef, and the fourth is bass clef. The notation includes various chords, some with trills (tr) and ties. The bass staff has chord labels: G, dm, am, em, dm, and G.

Handwritten musical notation for the second system, consisting of four staves. The first three staves are treble clef, and the fourth is bass clef. The notation includes various chords, some with trills (tr) and ties. The bass staff has chord labels: C, am7, dm7, E7, am, dm, and A.

Handwritten musical score on four staves. The first three staves contain melodic lines with various notes, rests, and trills. The fourth staff contains a bass line with chords and accidentals. The music is written in treble and bass clefs with a key signature of one sharp (F#).

**Staff 1 (Treble Clef):** Measures 1-4. Notes include F#4, G#4, A4, B4, C5, D5, E5, F#5, G#5, A5, B5, C6, D6, E6, F#6, G#6, A6, B6, C7, D7, E7, F#7, G#7, A7, B7, C8, D8, E8, F#8, G#8, A8, B8, C9, D9, E9, F#9, G#9, A9, B9, C10, D10, E10, F#10, G#10, A10, B10, C11, D11, E11, F#11, G#11, A11, B11, C12, D12, E12, F#12, G#12, A12, B12, C13, D13, E13, F#13, G#13, A13, B13, C14, D14, E14, F#14, G#14, A14, B14, C15, D15, E15, F#15, G#15, A15, B15, C16, D16, E16, F#16, G#16, A16, B16, C17, D17, E17, F#17, G#17, A17, B17, C18, D18, E18, F#18, G#18, A18, B18, C19, D19, E19, F#19, G#19, A19, B19, C20, D20, E20, F#20, G#20, A20, B20, C21, D21, E21, F#21, G#21, A21, B21, C22, D22, E22, F#22, G#22, A22, B22, C23, D23, E23, F#23, G#23, A23, B23, C24, D24, E24, F#24, G#24, A24, B24, C25, D25, E25, F#25, G#25, A25, B25, C26, D26, E26, F#26, G#26, A26, B26, C27, D27, E27, F#27, G#27, A27, B27, C28, D28, E28, F#28, G#28, A28, B28, C29, D29, E29, F#29, G#29, A29, B29, C30, D30, E30, F#30, G#30, A30, B30, C31, D31, E31, F#31, G#31, A31, B31, C32, D32, E32, F#32, G#32, A32, B32, C33, D33, E33, F#33, G#33, A33, B33, C34, D34, E34, F#34, G#34, A34, B34, C35, D35, E35, F#35, G#35, A35, B35, C36, D36, E36, F#36, G#36, A36, B36, C37, D37, E37, F#37, G#37, A37, B37, C38, D38, E38, F#38, G#38, A38, B38, C39, D39, E39, F#39, G#39, A39, B39, C40, D40, E40, F#40, G#40, A40, B40, C41, D41, E41, F#41, G#41, A41, B41, C42, D42, E42, F#42, G#42, A42, B42, C43, D43, E43, F#43, G#43, A43, B43, C44, D44, E44, F#44, G#44, A44, B44, C45, D45, E45, F#45, G#45, A45, B45, C46, D46, E46, F#46, G#46, A46, B46, C47, D47, E47, F#47, G#47, A47, B47, C48, D48, E48, F#48, G#48, A48, B48, C49, D49, E49, F#49, G#49, A49, B49, C50, D50, E50, F#50, G#50, A50, B50, C51, D51, E51, F#51, G#51, A51, B51, C52, D52, E52, F#52, G#52, A52, B52, C53, D53, E53, F#53, G#53, A53, B53, C54, D54, E54, F#54, G#54, A54, B54, C55, D55, E55, F#55, G#55, A55, B55, C56, D56, E56, F#56, G#56, A56, B56, C57, D57, E57, F#57, G#57, A57, B57, C58, D58, E58, F#58, G#58, A58, B58, C59, D59, E59, F#59, G#59, A59, B59, C60, D60, E60, F#60, G#60, A60, B60, C61, D61, E61, F#61, G#61, A61, B61, C62, D62, E62, F#62, G#62, A62, B62, C63, D63, E63, F#63, G#63, A63, B63, C64, D64, E64, F#64, G#64, A64, B64, C65, D65, E65, F#65, G#65, A65, B65, C66, D66, E66, F#66, G#66, A66, B66, C67, D67, E67, F#67, G#67, A67, B67, C68, D68, E68, F#68, G#68, A68, B68, C69, D69, E69, F#69, G#69, A69, B69, C70, D70, E70, F#70, G#70, A70, B70, C71, D71, E71, F#71, G#71, A71, B71, C72, D72, E72, F#72, G#72, A72, B72, C73, D73, E73, F#73, G#73, A73, B73, C74, D74, E74, F#74, G#74, A74, B74, C75, D75, E75, F#75, G#75, A75, B75, C76, D76, E76, F#76, G#76, A76, B76, C77, D77, E77, F#77, G#77, A77, B77, C78, D78, E78, F#78, G#78, A78, B78, C79, D79, E79, F#79, G#79, A79, B79, C80, D80, E80, F#80, G#80, A80, B80, C81, D81, E81, F#81, G#81, A81, B81, C82, D82, E82, F#82, G#82, A82, B82, C83, D83, E83, F#83, G#83, A83, B83, C84, D84, E84, F#84, G#84, A84, B84, C85, D85, E85, F#85, G#85, A85, B85, C86, D86, E86, F#86, G#86, A86, B86, C87, D87, E87, F#87, G#87, A87, B87, C88, D88, E88, F#88, G#88, A88, B88, C89, D89, E89, F#89, G#89, A89, B89, C90, D90, E90, F#90, G#90, A90, B90, C91, D91, E91, F#91, G#91, A91, B91, C92, D92, E92, F#92, G#92, A92, B92, C93, D93, E93, F#93, G#93, A93, B93, C94, D94, E94, F#94, G#94, A94, B94, C95, D95, E95, F#95, G#95, A95, B95, C96, D96, E96, F#96, G#96, A96, B96, C97, D97, E97, F#97, G#97, A97, B97, C98, D98, E98, F#98, G#98, A98, B98, C99, D99, E99, F#99, G#99, A99, B99, C100, D100, E100, F#100, G#100, A100, B100, C101, D101, E101, F#101, G#101, A101, B101, C102, D102, E102, F#102, G#102, A102, B102, C103, D103, E103, F#103, G#103, A103, B103, C104, D104, E104, F#104, G#104, A104, B104, C105, D105, E105, F#105, G#105, A105, B105, C106, D106, E106, F#106, G#106, A106, B106, C107, D107, E107, F#107, G#107, A107, B107, C108, D108, E108, F#108, G#108, A108, B108, C109, D109, E109, F#109, G#109, A109, B109, C110, D110, E110, F#110, G#110, A110, B110, C111, D111, E111, F#111, G#111, A111, B111, C112, D112, E112, F#112, G#112, A112, B112, C113, D113, E113, F#113, G#113, A113, B113, C114, D114, E114, F#114, G#114, A114, B114, C115, D115, E115, F#115, G#115, A115, B115, C116, D116, E116, F#116, G#116, A116, B116, C117, D117, E117, F#117, G#117, A117, B117, C118, D118, E118, F#118, G#118, A118, B118, C119, D119, E119, F#119, G#119, A119, B119, C120, D120, E120, F#120, G#120, A120, B120, C121, D121, E121, F#121, G#121, A121, B121, C122, D122, E122, F#122, G#122, A122, B122, C123, D123, E123, F#123, G#123, A123, B123, C124, D124, E124, F#124, G#124, A124, B124, C125, D125, E125, F#125, G#125, A125, B125, C126, D126, E126, F#126, G#126, A126, B126, C127, D127, E127, F#127, G#127, A127, B127, C128, D128, E128, F#128, G#128, A128, B128, C129, D129, E129, F#129, G#129, A129, B129, C130, D130, E130, F#130, G#130, A130, B130, C131, D131, E131, F#131, G#131, A131, B131, C132, D132, E132, F#132, G#132, A132, B132, C133, D133, E133, F#133, G#133, A133, B133, C134, D134, E134, F#134, G#134, A134, B134, C135, D135, E135, F#135, G#135, A135, B135, C136, D136, E136, F#136, G#136, A136, B136, C137, D137, E137, F#137, G#137, A137, B137, C138, D138, E138, F#138, G#138, A138, B138, C139, D139, E139, F#139, G#139, A139, B139, C140, D140, E140, F#140, G#140, A140, B140, C141, D141, E141, F#141, G#141, A141, B141, C142, D142, E142, F#142, G#142, A142, B142, C143, D143, E143, F#143, G#143, A143, B143, C144, D144, E144, F#144, G#144, A144, B144, C145, D145, E145, F#145, G#145, A145, B145, C146, D146, E146, F#146, G#146, A146, B146, C147, D147, E147, F#147, G#147, A147, B147, C148, D148, E148, F#148, G#148, A148, B148, C149, D149, E149, F#149, G#149, A149, B149, C150, D150, E150, F#150, G#150, A150, B150, C151, D151, E151, F#151, G#151, A151, B151, C152, D152, E152, F#152, G#152, A152, B152, C153, D153, E153, F#153, G#153, A153, B153, C154, D154, E154, F#154, G#154, A154, B154, C155, D155, E155, F#155, G#155, A155, B155, C156, D156, E156, F#156, G#156, A156, B156, C157, D157, E157, F#157, G#157, A157, B157, C158, D158, E158, F#158, G#158, A158, B158, C159, D159, E159, F#159, G#159, A159, B159, C160, D160, E160, F#160, G#160, A160, B160, C161, D161, E161, F#161, G#161, A161, B161, C162, D162, E162, F#162, G#162, A162, B162, C163, D163, E163, F#163, G#163, A163, B163, C164, D164, E164, F#164, G#164, A164, B164, C165, D165, E165, F#165, G#165, A165, B165, C166, D166, E166, F#166, G#166, A166, B166, C167, D167, E167, F#167, G#167, A167, B167, C168, D168, E168, F#168, G#168, A168, B168, C169, D169, E169, F#169, G#169, A169, B169, C170, D170, E170, F#170, G#170, A170, B170, C171, D171, E171, F#171, G#171, A171, B171, C172, D172, E172, F#172, G#172, A172, B172, C173, D173, E173, F#173, G#173, A173, B173, C174, D174, E174, F#174, G#174, A174, B174, C175, D175, E175, F#175, G#175, A175, B175, C176, D176, E176, F#176, G#176, A176, B176, C177, D177, E177, F#177, G#177, A177, B177, C178, D178, E178, F#178, G#178, A178, B178, C179, D179, E179, F#179, G#179, A179, B179, C180, D180, E180, F#180, G#180, A180, B180, C181, D181, E181, F#181, G#181, A181, B181, C182, D182, E182, F#182, G#182, A182, B182, C183, D183, E183, F#183, G#183, A183, B183, C184, D184, E184, F#184, G#184, A184, B184, C185, D185, E185, F#185, G#185, A185, B185, C186, D186, E186, F#186, G#186, A186, B186, C187, D187, E187, F#187, G#187, A187, B187, C188, D188, E188, F#188, G#188, A188, B188, C189, D189, E189, F#189, G#189, A189, B189, C190, D190, E190, F#190, G#190, A190, B190, C191, D191, E191, F#191, G#191, A191, B191, C192, D192, E192, F#192, G#192, A192, B192, C193, D193, E193, F#193, G#193, A193, B193, C194, D194, E194, F#194, G#194, A194, B194, C195, D195, E195, F#195, G#195, A195, B195, C196, D196, E196, F#196, G#196, A196, B196, C197, D197, E197, F#197, G#197, A197, B197, C198, D198, E198, F#198, G#198, A198, B198, C199, D199, E199, F#199, G#199, A199, B199, C200, D200, E200, F#200, G#200, A200, B200, C201, D201, E201, F#201, G#201, A201, B201, C202, D202, E202, F#202, G#202, A202, B202, C203, D203, E203, F#203, G#203, A203, B203, C204, D204, E204, F#204, G#204, A204, B204, C205, D205, E205, F#205, G#205, A205, B205, C206, D206, E206, F#206, G#206, A206, B206, C207, D207, E207, F#207, G#207, A207, B207, C208, D208, E208, F#208, G#208, A208, B208, C209, D209, E209, F#209, G#209, A209, B209, C210, D210, E210, F#210, G#210, A210, B210, C211, D211, E211, F#211, G#211, A211, B211, C212, D212, E212, F#212, G#212, A212, B212, C213, D213, E213, F#213, G#213, A213, B213, C214, D214, E214, F#214, G#214, A214, B214, C215, D215, E215, F#215, G#215, A215, B215, C216, D216, E216, F#216, G#216, A216, B216, C217, D217, E217, F#217, G#217, A217, B217, C218, D218, E218, F#218, G#218, A218, B218, C219, D219, E219, F#219, G#219, A219, B219, C220, D220, E220, F#220, G#220, A220, B220, C221, D221, E221, F#221, G#221, A221, B221, C222, D222, E222, F#222, G#222, A222, B222, C223, D223, E223, F#223, G#223, A223, B223, C224, D224, E224, F#224, G#224, A224, B224, C225, D225, E225, F#225, G#225, A225, B225, C226, D226, E226, F#226, G#226, A226, B226, C227, D227, E227, F#227, G#227, A227, B227, C228, D228, E228, F#228, G#228, A228, B228, C229, D229, E229, F#229, G#229, A229, B229, C230, D230, E230, F#230, G#230, A230, B230, C231, D231, E231, F#231, G#231, A231, B231, C232, D232, E232, F#232, G#232, A232, B232, C233, D233, E233, F#233, G#233, A233, B233, C234, D234, E234, F#234, G#234, A234, B234, C235, D235, E235, F#235, G#235, A235, B235, C236, D236, E236, F#236, G#236, A236, B236, C237, D237, E237, F#237, G#237, A237, B237, C238, D238, E238, F#238, G#238, A238, B238, C239, D239, E239, F#239, G#239, A239, B239, C240, D240, E240, F#240, G#240, A240, B240, C241, D241, E241, F#241, G#241, A241, B241, C242, D242, E242, F#242, G#242, A242, B242, C243, D243, E243, F#243, G#243, A243, B243, C244, D244, E244, F#244, G#244, A244, B244, C245, D245, E245, F#245, G#245, A245, B245, C246, D246, E246, F#246, G#246, A246, B246, C247, D247, E247, F#247, G#247, A247, B247, C248, D248, E248, F#248, G#248, A248, B248, C249, D249, E249, F#249, G#249, A249, B249, C250, D250, E250, F#250, G#250, A250, B250, C251, D251, E251, F#251, G#251, A251, B251, C252, D252, E252, F#252, G#252, A252, B252, C253, D253, E253, F#253, G#253, A253, B253, C254, D254, E254, F#254, G#254, A254, B254, C255, D255, E255, F#255, G#255, A255, B255, C256, D256, E256, F#256, G#256, A256, B256, C257, D257, E257, F#257, G#257, A257, B257, C258, D258, E258, F#258, G#258, A258, B258, C259, D259, E259, F#259, G#259, A259, B259, C260, D260, E260, F#260, G#260, A260, B260, C261, D261, E261, F#261, G#261, A261, B261, C262, D262, E262, F#262, G#262, A262, B262, C263, D263, E263, F#263, G#263, A263, B263, C264, D264, E264, F#264, G#264, A264, B264, C265, D265, E265, F#265, G#265, A265, B265, C266, D266, E266, F#266, G#266, A266, B266, C267, D267, E267, F#267, G#267, A267, B267, C268, D268, E268, F#268, G#268, A268, B268, C269, D269, E269, F#269, G#269, A269, B269, C270, D270, E270, F#270, G#270, A270, B270, C271, D271, E271, F#271, G#271, A271, B271, C272, D272, E272, F#272, G#272, A272, B272, C273, D273, E273, F#273, G#273, A273, B273, C274, D274, E274, F#274, G#274, A274, B274, C275, D275, E275, F#275, G#275, A275, B275, C276, D276, E276, F#276, G#276, A276, B276, C277, D277, E277, F#277, G#277, A277, B277, C278, D278, E278, F#278, G#278, A278, B278, C279, D279, E279, F#279, G#279, A279, B279, C280, D280, E280, F#280, G#280, A280, B280, C281, D281, E281, F#281, G#281, A281, B281, C282, D282, E282, F#282, G#282, A282, B282, C283, D283, E283, F#283, G#283, A283, B283, C284, D284, E284, F#284, G#284, A284, B284, C285, D285, E285, F#285, G#285, A285, B285, C286, D286, E286, F#286, G#286, A286, B286, C287, D287, E287, F#287, G#287, A287, B287, C288, D288, E288, F#288, G#288, A288, B288, C289, D289, E289, F#289, G#289, A289, B289, C290, D290, E290, F#290, G#290, A290, B290, C291, D291, E291, F#291, G#291, A291, B291, C292, D292, E292, F#292, G#292, A292, B292, C293, D293, E293, F#293, G#293, A293, B293, C294, D294, E294, F#294, G#294, A294, B294, C295, D295, E295, F#295, G#295, A295, B295, C296, D296, E296, F#296, G#296, A296, B296, C297, D297, E297, F#297, G#297, A297, B297, C298, D298, E298, F#298, G#298, A298, B298, C299, D299, E299, F#299, G#299, A299, B299, C300, D300, E300, F#300, G#300, A300, B300, C301, D301, E301, F#301, G#301, A301, B301, C302, D302, E302, F#302, G#302, A302, B302, C303, D303, E303, F#303, G#303, A303, B303, C304, D304, E304, F#304, G#304, A304, B304, C305, D305, E305, F#305, G#305, A305, B305, C306, D306, E306, F#306, G#306, A306, B306, C307, D307, E307, F#307, G#307, A307, B307, C308, D308, E308, F#308, G#308, A308, B308, C309, D309, E309, F#309, G#309, A309, B309, C310, D310, E310, F#310, G#310, A310, B310, C311, D311, E311, F#311, G#311, A311, B311, C312, D312, E312, F#312, G#312, A312, B312, C313, D313, E313, F#313, G#313, A313, B313, C314, D314, E314, F#314, G#314, A314, B314, C315, D315, E315, F#315, G#315, A315, B315, C316, D316, E316, F#316, G#316, A316, B316, C317, D317, E317, F#317, G#317, A317, B317, C318, D318, E318, F#318, G#318, A318, B318, C319, D319, E319, F#319, G#319, A319, B319, C320, D320, E320, F#320, G#320, A320, B320, C321, D321, E321, F#321, G#321, A321, B321, C322, D322, E322, F#322, G#322, A322, B322, C323, D323, E323, F#323, G#323, A323, B323, C324, D324, E324, F#324, G#324, A324, B324, C325, D325, E325, F#325, G#325, A325, B325, C326, D326, E326, F#326, G#326, A326, B326, C327, D327, E327, F#327, G#327, A327, B327, C328, D328, E328, F#328, G#328, A328, B328, C329, D329, E329, F#329, G#329, A329, B329, C330, D330, E330, F#330, G#330, A330, B330, C331, D331, E331, F#331, G#331, A331, B331, C332, D332, E332, F#332, G#332, A332, B332, C333, D333, E333, F#333, G#333, A333, B333, C334, D334, E334, F#334, G#334, A334, B334, C335, D335, E335, F#335, G#335, A335, B335, C336, D336, E336, F#336, G#336, A336, B336, C337, D337, E337, F#337, G#337, A337, B337, C338, D338, E338, F#338, G#338, A338, B338, C339, D339, E339, F#339, G#339, A339, B339, C340, D340, E340, F#340, G#340, A340, B340, C341, D341, E341, F#341, G#341, A341, B341, C342, D342, E342, F#342, G#342, A342, B342, C343, D343, E343, F#343, G#343, A343, B343, C344, D344, E344, F#344, G#344, A344, B344, C345, D345, E345, F#345, G#345, A345, B345, C346, D346, E346, F#346, G#346, A346, B346, C347, D347, E347, F#347, G#347, A347, B347, C348, D348, E348, F#348, G#348, A348, B348, C349, D349, E349, F#349, G#349, A349, B349, C350, D350, E350, F#350, G#350, A350, B350, C351, D351, E351, F#351, G#351, A351, B351, C352, D352, E352, F#352, G#352, A352, B352, C353, D3

```
[
[[am],[q],[e1]],[[am],[e],[c1]],[[am],[e],[a]],[[gsdim],[q],[f1]],
[[gsdim],[te],[f1]],[[gsdim],[s],[e1]],[[gsdim],[s],[d1]],[[am],[e],[c1]],
[[am],[e],[e1]],[[am],[e],[a]],[[am],[e],[b]],[[f],[e],[c1]],[[f],[e],[e1]],
[[dsdim],[q],[ds1]],
[[e],[dq],[e1]],[[e],[e],[e1]],[[am],[e],[c1]],
[[am],[e],[e1]],[[am],[e],[a]],[[am],[e],[c1]],[[dm],[q],[f1]],
[[dm],[ts],[f1]],[[dm],[s],[e1]],[[dm],[s],[d1]],[[dm],[s],[c1]],
[[g7],[e],[b]],[[g7],[e],[d1]],[[g7],[e],[g]],[[g7],[e],[b]],
[[c],[q],[e1]],[[c],[ts],[e1]],[[c],[s],[d1]],[[c],[s],[c1]],[[c],[s],[b]],
[[f],[e],[a]],[[f],[e],[c1]],[[f],[e],[f]],[[f],[e],[a]],
[[bdim],[q],[d1]],[[bdim],[ts],[d1]],[[bdim],[s],[c1]],[[bdim],[s],[b]],
[[bdim],[s],[a]],[[e],[e],[gs]],[[e],[e],[b]],[[e],[e],[e]],[[e],[e],[e1]],
[[am],[q],[c1]],[[am],[te],[c1]],[[am],[s],[b]],[[am],[s],[a]],
[[e],[e],[gs]],[[e],[s],[a]],[[e],[s],[b]],[[e],[de],[b]],[[e],[s],[a]],
[[am],[dq],[a]],[[am],[e],[a]],[[e],[q],[e1]],[[e],[ts],[e1]],
[[e],[s],[d1]],[[e],[s],[c1]],[[e],[s],[b]],[[am],[e],[c1]],[[am],[e],[a]],
[[f],[s],[f1]],[[f],[s],[e1]],[[f],[s],[d1]],[[f],[s],[c1]],[[bdim],[e],[b]],
[[bdim],[e],[b]],[[em],[s],[e1]],[[em],[s],[d1]],[[em],[s],[c1]],
[[em],[s],[b]],[[am],[e],[a]],[[am],[e],[a]],[[dm],[s],[d1]],[[dm],[s],[c1]],
[[dm],[s],[b]],[[dm],[s],[a]],[[g7],[e],[b]],[[g7],[e],[g]],
[[c],[de],[g1]],[[c],[s],[f1]],[[f],[s],[e1]],[[f],[s],[d1]],[[dm],[s],[c1]],
[[dm],[s],[d1]],[[g],[de],[d1]],[[g],[s],[c1]],[[c],[q],[c1]],[[c],[q],[e1]],
[[c],[te],[e1]],[[c],[s],[d1]],[[c],[s],[c1]],[[g],[e],[d1]],
[[g],[e],[e1]],[[dm],[q],[f1]],[[dm],[q],[f1]],[[dm],[te],[f1]],
[[dm],[s],[e1]],[[dm],[s],[d1]],[[am],[e],[e1]],[[am],[e],[f1]],
[[em],[q],[g1]],[[em],[te],[g1]],[[em],[s],[f1]],[[em],[s],[e1]],
[[dm],[q],[d1]],[[dm],[ts],[d1]],[[dm],[s],[e1]],[[dm],[s],[d1]],
[[dm],[s],[c1]],[[g],[q],[b]],[[g],[e],[r]],[[g],[s],[g1]],[[g],[s],[f1]],
[[c],[s],[e1]],[[c],[s],[d1]],[[c],[s],[c1]],[[c],[s],[d1]],[[c],[de],[d1]],
[[c],[s],[c1]],[[am7],[q],[c1]],[[dm7],[te],[c1]],[[dm7],[s],[b]],
[[dm7],[s],[a]],[[e7],[e],[gs]],[[e7],[e],[e1]],[[am],[ts],[e1]],
[[am],[s],[d1]],[[am],[s],[c1]],[[am],[s],[b]],[[dm],[e],[a]],
[[dm],[e],[a1]],[[a],[ts],[a1]],[[a],[s],[g1]],[[a],[s],[f1]],[[a],[s],[e1]],
[[dm],[s],[f1]],[[dm],[s],[e1]],[[dm],[s],[d1]],[[dm],[s],[c1]],
[[b7],[de],[b]],[[b7],[s],[a]],[[e],[dq],[gs]],[[e],[e],[gs]],[[dsdim],[dq],[a]],
[[dsdim],[e],[a]],[[am],[s],[a]],[[am],[s],[c1]],[[b7],[s],[b]],
[[b7],[s],[a]],[[e],[de],[gs]],[[e],[s],[a]],[[am],[q],[a]],[[em],[h],[e1]],
[[dsdim],[q],[ds1]],[[e],[w],[e1]]
].
```

```
[
[am],[q],[e1]], [[am],[e],[c1]], [[am],[e],[a]], [[gsdim],[q],[f1]],
[[gsdim],[te],[f1]], [[gsdim],[e],[d1]], [[am],[q],[c1]],
[am],[e],[a]], [[am],[e],[b]], [[f],[q],[c1]],
[[dsdim],[q],[ds1]],
[e],[dq],[e1]], [[e],[e],[e1]], [[am],[q],[c1]],
[am],[q],[a]], [[dm],[q],[f1]],
[dm],[te],[f1]], [[dm],[e],[d1]],
[[g7],[q],[b]], [[g7],[q],[g]],
[[c],[q],[e1]], [[c],[te],[e1]], [[c],[e],[c1]],
[[f],[q],[a]], [[f],[q],[f]],
[[bdim],[q],[d1]], [[bdim],[te],[d1]], [[bdim],[e],[b]],
[[e],[q],[gs]], [[e],[q],[e]],
[am],[q],[c1]], [[am],[te],[c1]], [[am],[s],[b]], [[am],[s],[a]],
[[e],[e],[gs]], [[e],[s],[a]], [[e],[s],[b]], [[e],[de],[b]], [[e],[s],[a]],
[am],[dq],[a]], [[am],[e],[a]], [[e],[q],[e1]], [[e],[te],[e1]],
[[e],[e],[b]], [[am],[e],[c1]], [[am],[e],[a]],
[[f],[e],[f1]], [[f],[e],[d1]], [[bdim],[e],[b]],
[[bdim],[e],[b]], [[em],[e],[e1]], [[em],[e],[c1]],
[am],[e],[a]], [[am],[e],[a]], [[dm],[e],[d1]],
[dm],[e],[b]], [[g7],[e],[b]], [[g7],[e],[g]],
[[c],[q],[g1]], [[f],[e],[e1]], [[dm],[e],[c1]],
[[g],[de],[d1]], [[g],[s],[c1]], [[c],[q],[c1]], [[c],[q],[e1]],
[[c],[te],[e1]], [[c],[e],[c1]], [[g],[e],[d1]],
[[g],[e],[e1]], [[dm],[q],[f1]], [[dm],[q],[f1]], [[dm],[te],[f1]],
[[dm],[e],[d1]], [[am],[e],[e1]], [[am],[e],[f1]],
[[em],[q],[g1]], [[em],[te],[g1]], [[em],[e],[e1]],
[[dm],[q],[d1]], [[dm],[tq],[d1]],
[[g],[q],[b]], [[g],[e],[r]], [[g],[e],[g1]],
[[c],[e],[e1]], [[c],[e],[c1]], [[c],[de],[d1]],
[[c],[s],[c1]], [[am7],[q],[c1]], [[dm7],[te],[c1]],
[[dm7],[e],[a]], [[e7],[e],[gs]], [[e7],[e],[e1]], [[am],[te],[e1]],
[am],[e],[c1]], [[dm],[e],[a]],
[dm],[e],[a1]], [[a],[te],[a1]], [[a],[e],[f1]],
[dm],[e],[f1]], [[dm],[e],[d1]],
[[b7],[de],[b]], [[b7],[s],[a]], [[e],[dq],[gs]], [[e],[e],[gs]], [[dsdim],[dq],[a]],
[[dsdim],[e],[a]], [[am],[s],[a]], [[am],[s],[c1]], [[b7],[s],[b]],
[[b7],[s],[a]], [[e],[de],[gs]], [[e],[s],[a]], [[am],[q],[a]], [[em],[h],[e1]],
[[dsdim],[q],[ds1]], [[e],[w],[e1]]
].
```

```
?- v(Variation).
<Undefined Goal>:
| | | | | (12:249) alreadytried(4). :
Variation = [[am,am,am],[s,e,s,s,s,e],[dl,el,dl,cl,b,a]],
[[gsdim],[q],[fl]],[[gsdim,gsdim],[ts,s,e],
[fl,cl,dl]],
[[am,am],[e,e,e],[cl,g,a]],[[am,f],[s,s,q],
[b,trill,cl]],
[[dsdim,e],[q,dq],[ds1,trill,el]],[[e],[e],[el]],
[[am,am],[t,t,t,q],[cl,a,g,a]],[[dm],[q],[fl]],
[[dm,dm],[s,s,s,s],[fl,dl,cl,dl]],
[[g7,g7],[t,t,t,q],[b,g,f,g]],[[c],[q],[el]],
[[c,c],[s,s,s,s],[el,cl,b,cl]],[[f,f],[t,t,t,q],
[a,f,e,f]],
[[bdim],[q],[dl]],[[bdim,bdim],[s,s,s,s],[dl,b,a,
b]],
[[e,e],[t,t,t,q],[a,gs,fs,e]],[[am],[q],[cl]],
[[am],[te],[cl]],[[am],[s],[b]],[[am],[s],[a]],
[[e,e],[s,s,s],[gs,trill,a]],[[e],[s],[b]],
[[e],[de],[b]],[[e],[s],[a]],[[am],[dq],[a]],
[[am,e],[s,s,t,t,t],[a,b,cl,dl,el]],
[[e,e],[s,s,s,s],[el,b,as,b]],
[[am,am],[s,s,s,s],[dl,cl,b,a]],
[[f,f,bdim],[s,s,s,s,s,s],[el,fl,el,dl,cl,b]],
[[bdim,em],[th,th,th,th,s,s],[b,el,fl,el,dl,el]],
[[em,am],[th,th,th,th,s,s],[cl,cl,b,a,g,a]],
[[am,dm],[th,th,th,th,s,s],[a,dl,el,dl,cl,dl]],
[[dm],[e],[b]],[[g7,g7],[th,th,th,th,s,s],
[b,b,a,g,f,g]],
[[c,f,dm],[s,s,s,s,s,s,th,th,s],[gl,al,b1,gl,el,
fl,gl,el,cl]],
[[g],[de],[dl]],[[g],[s],[cl]],
[[c,c],[t,t,t,q],[cl,fl,g,el]],
[[c,c],[s,s,s,s],[el,cl,b,cl]],[[g],[e],[dl]],
[[g,dm],[s,s,q],[el,trill,fl]],[[dm],[q],[fl]],
[[dm,dm],[s,s,s,s],[fl,dl,cl,dl]],
[[am,am],[s,s,e],[el,trill,fl]],[[em],[q],[gl]],
[[em,em],[tth,th,th,th,s,s],[gl,gl,fl,el,dl,el]],
[[dm],[q],[dl]],[[dm,g],[ts,s,s,s,e,e],
[dl,dl,cl,b,a,b]],[[g],[e],[r]],
[[g,c,c],[th,th,th,th,th,th,th,th,e],
[gl,al,b1,gl,el,fl,gl,el,cl]],[[c],[de],[dl]],
[[c],[s],[cl]],[[am7],[q],[cl]],
[[dm7,dm7],[s,s,s,s],[dl,cl,b,a]],
[[e7,e7],[s,s,s,s],[b,as,gs,el]],
[[am,am,dm],[ts,s,s,s,th,th,th,th],[el,el,fl,dl,
cl,dl,b,a]],
[[dm],[e],[al]],[[a,a],[s,s,s,s],[b1,al,gl,fl]],
[[dm,dm,b7],[s,s,s,s,th,th,s,s],[fs1,fs1,gl,el,
ds1,el,cs1,b]],
[[b7],[s],[a]],[[e],[dq],[gs]],
[[e,dsdim],[e,e,q],[gs,trill,a]],[[dsdim],[e],[a]],
[[am,am],[th,th,th,th],[a,dl,e,cl]],[[b7],[s],[b]],
[[b7],[s],[a]],[[e,e],[de,s],[gs,trill,a]],
[[am,em],[s,s,s,s,h],[a,b,cl,dl,el]],
[[dsdim,e],[q,w],[ds1,trill,el]]]
```

?-

```

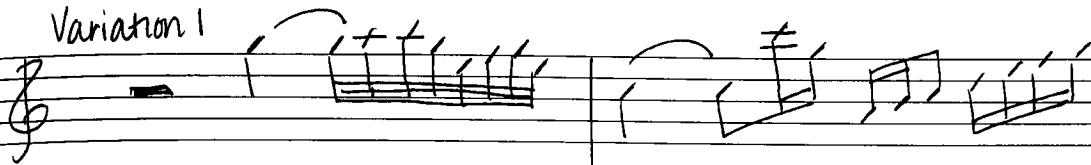
Variation = [[am,am,am],[s,s,s,s,s,s,th,th,s],[el,fl,gl,el,cl,
                                                    dl,el,cl,a]],
[[gsdim],[q],[fl]],
[[gsdim,gsdim],[tth,th,th,th,s,s],[fl,fl,el,dl,cl,
                                                    dl]],
[[am,am],[s,s,s,s,s,s],[cl,cl,b,a,g,a]],
[[am,f],[s,s,q],[b,trill,cl]],
[[dsdim,e],[q,dq],[dsl,trill,el]],[[e],[e],[el]],
[[am,am],[e,e,q],[cl,g,a]],[[dm],[q],[fl]],
[[dm,dm],[ts,s,e],[fl,cl,dl]],[[g7,g7],[e,e,q],
                                                    [b,f,g]],
[[c],[q],[el]],[[c,c],[ts,s,e],[el,b,cl]],
[[f,f],[e,e,q],[a,e,f]],[[bdim],[q],[dl]],
[[bdim,bdim],[ts,s,e],[dl,a,b]],
[[e,e],[t,t,t,q],[gs,e,d,e]],[[am],[q],[cl]],
[[am],[te],[cl]],[[am],[s],[b]],[[am],[s],[a]],
[[e,e],[s,s,s],[gs,trill,a]],[[e],[s],[b]],
[[e],[de],[b]],[[e],[s],[a]],[[am],[dq],[a]],
[[am,e],[e,s,s,e],[a,cl,fl,el]],
[[e,e],[tst,st,st,st,st,tht,tht],[dl,fsl,el,dl,el,
                                                    csl,b]],
[[am,am],[s,s,s,s],[cl,a,g,a]],
[[f,f,bdim],[th,th,th,th,th,th,th,th,e],
  [fl,gl,al,fl,dl,el,fl,dl,b]],
[[bdim,em],[th,th,th,th,e],[b,dl,cl,b,el]],
[[em,am],[s,s,s,s],[dl,cl,b,a]],
[[am,dm],[th,th,th,th,e],[a,cl,b,a,dl]],
[[dm],[e],[b]],[[g7,g7],[s,s,s,s],[cl,b,a,g]],
[[c,f,dm],[s,ts,s,s,s,s,s],[gl,gl,al,fl,el,fl,
                                                    dl,cl]],
[[g],[de],[dl]],[[g],[s],[cl]],
[[c,c],[t,t,t,q],[cl,fl,g,el]],
[[c,c],[ts,s,e],[el,b,cl]],[[g],[e],[dl]],
[[g,dm],[s,s,q],[el,trill,fl]],[[dm],[q],[fl]],
[[dm,dm],[ts,s,e],[fl,cl,dl]],
[[am,am],[s,s,e],[el,trill,fl]],[[em],[q],[gl]],
[[em,em],[s,s,s,s],[al,gl,fl,el]],[[dm],[q],[dl]],
[[dm,g],[e,e,e,e],[el,dl,cl,b]],[[g],[e],[r]],
[[g,c,c],[s,ts,s,s,th,th,th,th],[gl,gl,al,fl,el,
                                                    fl,dl,cl]],
[[c],[de],[dl]],[[c],[s],[cl]],[[am7],[q],[cl]],
[[dm7,dm7],[tth,th,th,th,s,s],[cl,cl,b,a,g,a]],
[[e7,e7],[s,s,s,s],[b,as,gs,el]],
[[am,am,dm],[ts,s,s,s,s,s],[dl,el,dl,cl,b,a]],
[[dm],[e],[al]],[[a,a],[tth,th,th,th,s,s],
  [al,al,gl,fl,el,fl]],
[[dm,dm,b7],[s,s,s,s,s,e],[el,fsl,el,dsl,cl,b]],
[[b7],[s],[a]],[[e],[dq],[gs]],
[[e,dsdim],[e,e,q],[gs,trill,a]],[[dsdim],[e],[a]],
[[am,am],[th,th,th,th],[a,dl,e,cl]],[[b7],[s],[b]],
[[b7],[s],[a]],[[e,e],[de,s],[gs,trill,a]],
[[am,em],[t,t,t,h],[a,cl,fl,el]],
[[dsdim,e],[q,w],[dsl,trill,el]]

```

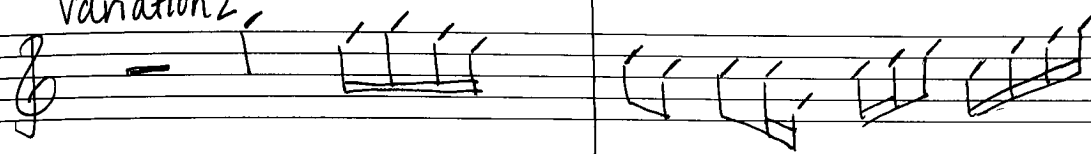
?-

# Loeillet - G

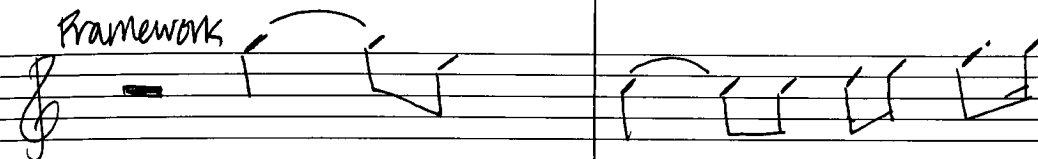
## Variation 1



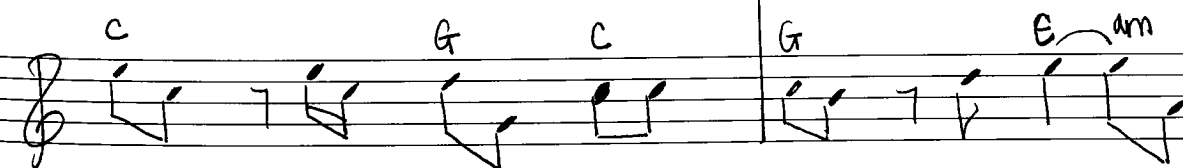
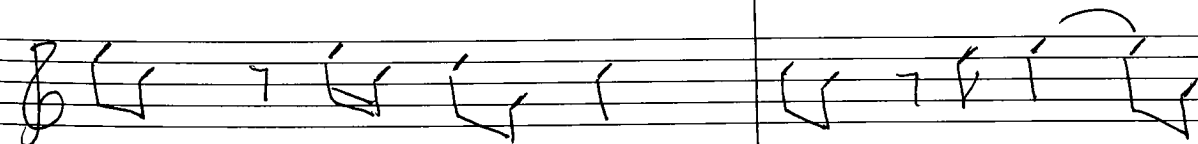
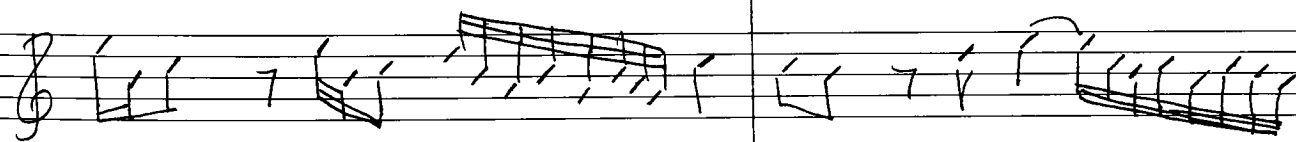
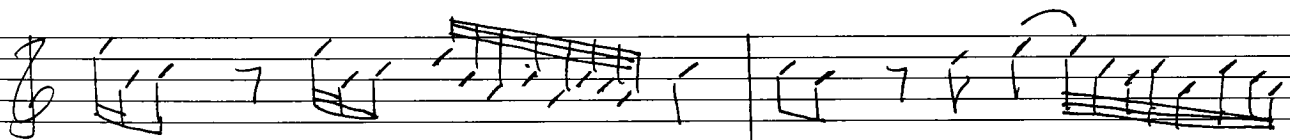
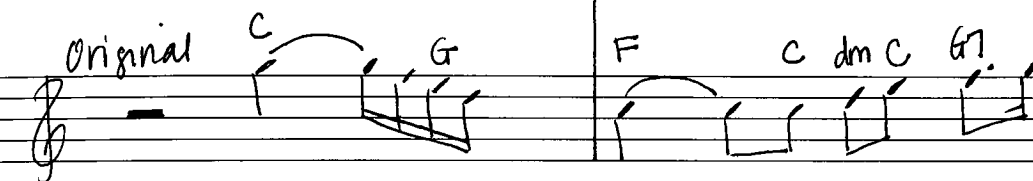
## Variation 2



## Framework



## Original





Handwritten musical notation for the first system, consisting of four staves. The first three staves are treble clef, and the fourth is bass clef. The key signature has one sharp (F#). The first staff contains a 7th chord, a triplet of eighth notes, a triplet of sixteenth notes with a sharp, and a triplet of eighth notes. The second staff contains a 7th chord, a triplet of eighth notes, and a triplet of eighth notes. The third staff contains a 7th chord, a half note, and a half note. The fourth staff contains a D major chord, a D7 chord, a G major chord, and a C major chord, an am chord, a D major chord, and a G major chord.

Handwritten musical notation for the second system, consisting of four staves. The first three staves are treble clef, and the fourth is bass clef. The key signature has one sharp (F#). The first staff contains a half note, a half note, a half note, and a half note. The second staff contains a half note, a half note, a half note, and a half note. The third staff contains a half note, a half note, a half note, and a half note. The fourth staff contains a G major chord, an A major chord, a dm chord, an edim chord, an A major chord, an A7 dm chord, an A major chord, a dm chord, an edim chord, an A major chord, a dm chord, and a bdim chord.

Handwritten musical notation for the first system, consisting of four staves. The first three staves are treble clef with a key signature of one sharp (F#). The fourth staff contains chord labels: E, E7, am, G, F, bdim, E, am, A7. The notation includes various note values, rests, and triplets.

Handwritten musical notation for the second system, consisting of four staves. The first three staves are treble clef with a key signature of one sharp (F#). The fourth staff contains chord labels: dm, G, G7, C, bdim C. The notation includes various note values, rests, and triplets.

Handwritten musical notation for the first system, consisting of four staves. The first three staves are treble clef, and the fourth is bass clef. The notation includes various chords and melodic lines with accidentals and ties.

Chord labels for the first system:

- Staff 4, Measure 1: F, dm
- Staff 4, Measure 2: G
- Staff 4, Measure 3: bdim
- Staff 4, Measure 4: C
- Staff 4, Measure 5: bdim C
- Staff 4, Measure 6: G

Handwritten musical notation for the second system, consisting of four staves. The first three staves are treble clef, and the fourth is bass clef. The notation includes various chords and melodic lines with accidentals and ties.

Chord labels for the second system:

- Staff 4, Measure 1: C
- Staff 4, Measure 2: cm
- Staff 4, Measure 3: ~~dim~~ bdim
- Staff 4, Measure 4: dm
- Staff 4, Measure 5: G7
- Staff 4, Measure 6: C

```
[
[[c],[h],[r]],[[c],[q],[gl]],[[c],[ts],[gl]],[[c],[s],[fl]],[[g],[s],[el]],
[[g],[s],[dl]],[[f],[q],[cl]],[[f],[te],[cl]],[[c],[e],[cl]],
[[dm],[e],[dl]],[[c],[e],[el]],[[g7],[de],[fl]],[[g7],[s],[gl]],
[[c],[e],[el]],[[c],[e],[cl]],[[c],[e],[r]],[[c],[s],[el]],
[[c],[s],[cl]],[[g],[e],[dl]],[[g],[e],[gl]],[[c],[e],[cl]],
[[c],[e],[cl]],[[g],[e],[cl]],[[g],[e],[b]],[[g],[e],[r]],
[[g],[e],[dl]],[[c],[q],[el]],[[am],[te],[el]],[[am],[e],[a]],
[[d],[q],[fs1]],[[d],[e],[r]],[[d7],[e],[dl]],[[g],[q],[gl]],
[[g],[ts],[gl]],[[g],[s],[fs1]],[[g],[s],[el]],[[g],[s],[dl]],
[[c],[s],[el]],[[c],[s],[dl]],[[am],[s],[cl]],[[am],[s],[b]],
[[d],[de],[a]],[[d],[s],[g]],[[g],[h],[g]],
[[g],[dq],[dl]],[[a],[e],[el]],[[dm],[q],[fl]],[[dm],[te],[fl]],
[[edim],[s],[el]],[[edim],[s],[dl]],[[a],[e],[cs1]],[[a],[e],[a]],
[[a],[e],[r]],[[a7],[e],[cs1]],[[dm],[q],[dl]],[[dm],[te],[dl]],
[[a],[e],[el]],[[dm],[e],[fl]],[[edim],[s],[el]],[[edim],[s],[dl]],
[[a],[de],[cs1]],[[a],[s],[dl]],[[dm],[e],[dl]],[[dm],[s],[fl]],
[[dm],[s],[el]],[[dm],[s],[dl]],[[dm],[s],[cl]],[[bdim],[s],[b]],
[[bdim],[s],[a]],[[e],[q],[gs]],[[e],[e],[r]],[[e7],[e],[b]],
[[am],[q],[el]],[[am],[ts],[el]],[[am],[s],[fl]],[[g],[s],[el]],
[[g],[s],[dl]],[[f],[e],[cl]],[[bdim],[s],[fl]],[[bdim],[s],[b]],
[[e],[de],[b]],[[e],[s],[a]],[[am],[q],[a]],[[am],[te],[a]],[[a7],[e],[a]],
[[dm],[q],[fl]],[[dm],[ts],[fl]],[[dm],[s],[el]],[[dm],[s],[dl]],
[[dm],[s],[cl]],[[g],[q],[b]],[[g],[ts],[b]],[[g],[s],[dl]],[[g7],[s],[el]],
[[g7],[s],[fl]],[[c],[q],[gl]],[[c],[te],[gl]],[[bdim],[e],[fl]],
[[c],[q],[el]],[[c],[te],[el]],[[c],[s],[fl]],[[c],[s],[gl]],
[[f],[q],[a1]],[[dm],[ts],[a1]],[[dm],[s],[gl]],[[dm],[s],[fl]],
[[dm],[s],[el]],[[g],[q],[dl]],[[g],[e],[r]],[[bdim],[e],[dl]],
[[c],[q],[gl]],[[c],[te],[gl]],[[bdim],[e],[fl]],[[c],[s],[el]],
[[c],[s],[dl]],[[c],[s],[cl]],[[c],[s],[dl]],[[g],[de],[dl]],[[g],[s],[cl]],
[[c],[e],[cl]],[[cm],[s],[ef1]],[[cm],[s],[dl]],[[cm],[q],[ef1]],
[[fsdim],[te],[ef1]],[[fsdim],[s],[dl]],[[fsdim],[s],[cl]],[[dm],[s],[dl]],
[[dm],[s],[ef1]],[[dm],[s],[dl]],[[dm],[s],[cl]],[[g7],[s],[b]],
[[g7],[s],[g]],[[g7],[s],[cl]],[[g7],[s],[dl]],[[g7],[e],[g]],[[g7],[e],[b]],
[[c],[h],[cl]]
].
```

```
[
[[c],[h],[r]],[[c],[q],[g1]],[[c],[te],[g1]],[[g],[e],[e1]],
[[f],[q],[c1]],[[f],[te],[c1]],[[c],[e],[c1]],
[[dm],[e],[d1]],[[c],[e],[e1]],[[g7],[de],[f1]],[[g7],[s],[g1]],
[[c],[e],[e1]],[[c],[e],[c1]],[[c],[e],[r]],[[c],[s],[e1]],
[[c],[s],[c1]],[[g],[e],[d1]],[[g],[e],[g]],[[c],[q],[c1]],
[[g],[e],[c1]],[[g],[e],[b]],[[g],[e],[r]],
[[g],[e],[d1]],[[c],[q],[e1]],[[am],[te],[e1]],[[am],[e],[a]],
[[d],[q],[fs1]],[[d],[e],[r]],[[d7],[e],[d1]],[[g],[q],[g1]],
[[g],[te],[g1]],[[g],[e],[d1]],
[[c],[q],[e1]],[[d],[de],[a]],[[d],[s],[g]],[[g],[h],[g]],
[[g],[h],[d1]],[[dm],[q],[f1]],[[dm],[te],[f1]],
[[edim],[e],[e1]],[[a],[e],[cs1]],[[a],[e],[a]],
[[a],[e],[r]],[[a7],[e],[cs1]],[[dm],[h],[d1]],[[dm],[q],[f1]],
[[a],[de],[cs1]],[[a],[s],[d1]],[[dm],[e],[d1]],[[dm],[e],[f1]],
[[dm],[e],[d1]],[[bdim],[e],[b]],
[[e],[q],[gs]],[[e],[e],[r]],[[e7],[e],[b]],
[[am],[q],[e1]],[[am],[te],[e1]],[[g],[e],[e1]],
[[f],[e],[c1]],[[bdim],[s],[f1]],[[bdim],[s],[b]],
[[e],[de],[b]],[[e],[s],[a]],[[am],[q],[a]],[[am],[te],[a]],[[a7],[e],[a]],
[[dm],[q],[f1]],[[dm],[te],[f1]],[[dm],[e],[d1]],
[[g],[q],[b]],[[g],[te],[b]],[[g7],[e],[e1]],
[[c],[q],[g1]],[[c],[te],[g1]],[[bdim],[e],[f1]],
[[c],[q],[e1]],[[c],[tq],[e1]],
[[f],[q],[a1]],[[dm],[te],[a1]],[[dm],[e],[f1]],
[[g],[q],[d1]],[[g],[e],[r]],[[bdim],[e],[d1]],
[[c],[q],[g1]],[[c],[te],[g1]],[[bdim],[e],[f1]],[[c],[s],[e1]],
[[c],[s],[d1]],[[c],[e],[c1]],[[g],[de],[d1]],[[g],[s],[c1]],
[[c],[e],[c1]],[[cm],[s],[ef1]],[[cm],[s],[d1]],[[cm],[q],[ef1]],
[[fsdim],[te],[ef1]],[[fsdim],[s],[d1]],[[fsdim],[s],[c1]],[[dm],[q],[d1]],
[[g7],[s],[b]],
[[g7],[s],[g]],[[g7],[s],[c1]],[[g7],[s],[d1]],[[g7],[e],[g]],[[g7],[e],[b]],
[[c],[h],[c1]]
].
```

?- v(Variation).

```
Variation = [[ [c], [h], [r]], [ [c], [q], [g1]],
  [ [c, g, f], [tth, th, th, th, th, th, th, q],
    [g1, a1, b1, g1, e1, f1, g1, e1, c1]], [ [f], [te], [c1]],
  [ [c, dm, c, g7, g7], [s, s, s, s, e, s, s, s], [c1, g, a, b, c1,
    dl, e1, f1, g1]],
  [ [c, c], [s, s, e], [e1, b, c1]], [ [c], [e], [r]],
  [ [c, c], [th, th, s], [e1, b, c1]],
  [ [g, g], [th, th, th, th, th, th, th, th], [dl, b, a, b, g, b, a,
    g]],
  [ [c], [q], [c1]], [ [g], [e], [c1]], [ [g], [e], [b]],
  [ [g], [e], [r]], [ [g], [e], [dl]], [ [c], [q], [e1]],
  [ [am, am], [tth, th, th, th, th, th, th, th], [e1, c1, b, c1, a,
    c1, b, a]],
  [ [d], [q], [fs1]], [ [d], [e], [r]],
  [ [d7, g], [s, s, t, t, t], [dl, fs1, e1, dl, g1]],
  [ [g, g], [tst, st, st, st, st, tht, tht], [f1, as1, g1, f1, g1,
    e1, dl]],
  [ [c, d], [e, e, s, s, s], [e1, c1, a, a, a]], [ [d], [s], [g]],
  [ [g, g], [t, t, t, q, h], [g, a, b, c1, dl]], [ [dm], [q], [f1]],
  [ [dm], [te], [f1]],
  [ [edim, a, a], [th, th, th, th, th, th, th, th, e],
    [e1, f1, g1, e1, cs1, dl, e1, cs1, a]], [ [a], [e], [r]],
  [ [a7, dm], [e, h], [cs1, trill, dl]],
  [ [dm, a], [s, s, s, s, s, s, s], [e1, gs1, f1, e1, f1, dl, cs1]],
  [ [a], [s], [dl]], [ [dm, dm], [s, s, s, s], [dl, g1, a, f1]],
  [ [dm, bdim, e], [th, th, th, th, s, s, e, e], [dl, dl, e1, c1, b,
    c1, a, gs]],
  [ [e], [e], [r]], [ [e7, am], [s, s, t, t, t], [b, dl, c1, b, e1]],
  [ [am], [te], [e1]], [ [g, f], [s, s, s, s], [f1, e1, dl, c1]],
  [ [bdim, bdim], [th, th, tht, tht, tht], [f1, dl, b, b, b]],
  [ [e], [de], [b]], [ [e], [s], [a]], [ [am], [q], [a]],
  [ [am], [te], [a]], [ [a7, dm], [e, s, s, e], [cs1, b, a, f1]],
  [ [dm, dm, g], [s, th, th, th, th, th, th, q], [f1, f1, g1, e1,
    dl, e1, cs1, b]],
  [ [g, g7], [tth, th, th, th, s, s], [b, e1, f1, e1, dl, e1]],
  [ [c], [q], [g1]], [ [c], [te], [g1]], [ [bdim], [e], [f1]],
  [ [c], [q], [e1]], [ [c, f], [ts, s, s, s, e, e],
    [e1, a1, b1, a1, g1, a1]],
  [ [dm, dm, g], [s, s, s, s, e, e], [g1, a1, g1, f1, e1, dl]],
  [ [g], [e], [r]], [ [bdim, c], [e, s, s, e], [dl, d2, b1, g1]],
  [ [c, bdim, c, c, c], [ts, s, s, s, s, s, e], [g1, a1, f1, e1, f1,
    dl, c1]],
  [ [g], [de], [dl]], [ [g], [s], [c1]], [ [c], [e], [c1]],
  [ [cm], [s], [ef1]], [ [cm], [s], [dl]], [ [cm], [q], [ef1]],
  [ [fsdim], [te], [ef1]], [ [fsdim], [s], [dl]],
  [ [fsdim], [s], [c1]],
  [ [dm, g7, g7], [s, e, s, th, th, s], [c1, dl, c1, b, a, g]],
  [ [g7], [s], [c1]], [ [g7, g7], [s, s, s], [dl, d, g]],
  [ [g7, c], [e, h], [b, trill, c1]]]
```

?-

?- v(Variation).

```
Variation = [[c],[h],[r]],[[c],[q],[g1]],
[[c,g,f],[s,s,s,s,e,e],[f1,g1,f1,e1,d1,c1]],
[[f],[te],[c1]],[[c,dm,c,g7,g7],[s,s,s,s,e,s,s,s,
s],[c1,g,c1,d1,
e1,c1,e1,f1,
g1]],
[[c,c],[s,s,e],[e1,b,c1]],[[c],[e],[r]],
[[c,c],[th,th,s],[e1,b,c1]],
[[g,g],[th,th,th,th,th,th,th,th],[d1,b,a,b,g,b,a,
g]],
[[c],[q],[c1]],[[g],[e],[c1]],[[g],[e],[b]],
[[g],[e],[r]],[[g],[e],[d1]],[[c],[q],[e1]],
[[am,am],[tth,th,th,th,th,th,th,th],[e1,c1,b,c1,a,
c1,b,a]],
[[d],[q],[fs1]],[[d],[e],[r]],
[[d7,g],[e,s,s,e],[d1,d2,b1,g1]],
[[g,g],[s,s,s,s],[g1,f1,g1,d1]],
[[c,d],[e,e,s,s,s],[e1,c1,a,a,a]],[[d],[s],[g]],
[[g,g],[q,q,q,q],[g,b,e1,d1]],[[dm],[q],[f1]],
[[dm],[te],[f1]],[[edim,a,a],[s,s,s,s,s,s],
[d1,e1,d1,cs1,b,a]],[[a],[e],
[r]],
[[a7,dm],[e,h],[cs1,trill,d1]],
[[dm,a],[e,s,s,de],[f1,e1,f1,cs1]],[[a],[s],[d1]],
[[dm,dm],[s,s,s,s],[d1,g1,a,f1]],
[[dm,bdim,e],[th,th,th,th,th,th,th,th,q],
[d1,e1,f1,d1,b,c1,d1,b,gs]],[[e],[e],[r]],
[[e7,am],[e,s,s,e],[b,b1,gs1,e1]],[[am],[te],[e1]],
[[g,f],[s,s,s,s],[e1,c1,b,c1]],
[[bdim,bdim],[th,th,tth,tth,tth],[f1,d1,b,b,b]],
[[e],[de],[b]],[[e],[s],[a]],[[am],[q],[a]],
[[am],[te],[a]],[[a7,dm],[e,s,s,e],[cs1,b,a,f1]],
[[dm,dm,g],[tth,th,th,th,th,th,th,th,q],
[f1,g1,a1,f1,d1,e1,f1,d1,b]],
[[g,g7],[tth,th,th,th,e],[b,d1,c1,b,e1]],
[[c],[q],[g1]],[[c],[te],[g1]],[[bdim],[e],[f1]],
[[c],[q],[e1]],[[c,f],[te,s,th,th,q],[e1,g1,f1,e1,
a1]],
[[dm,dm,g],[s,th,th,th,th,th,th,q],[a1,a1,b1,g1,
f1,g1,e1,d1]],
[[g],[e],[r]],[[bdim,c],[s,s,s,s,s,s],
[d1,g1,a1,g1,f1,g1]],
[[c,bdim,c,c,c],[ts,s,s,s,s,s,e],[g1,f1,e1,d1,c1,
d1,c1]],
[[g],[de],[d1]],[[g],[s],[c1]],[[c],[e],[c1]],
[[cm],[s],[ef1]],[[cm],[s],[d1]],[[cm],[q],[ef1]],
[[fsdim],[te],[ef1]],[[fsdim],[s],[d1]],
[[fsdim],[s],[c1]],
[[dm,g7,g7],[s,s,s,s,th,th,th,th],[d1,d1,e1,c1,b,
c1,a,g]],
[[g7],[s],[c1]],[[g7,g7],[s,s,s],[d1,d,g]],
[[g7,c],[e,h],[b,trill,c1]]]
```

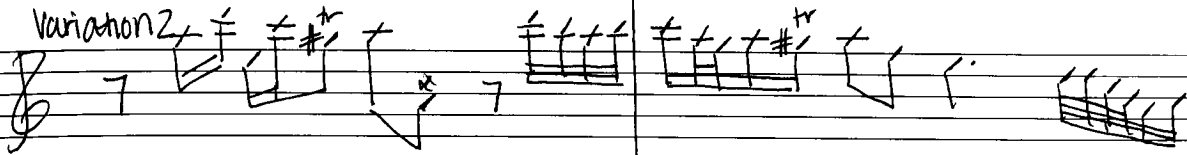
?-

# Telemann - BF

Variation 1



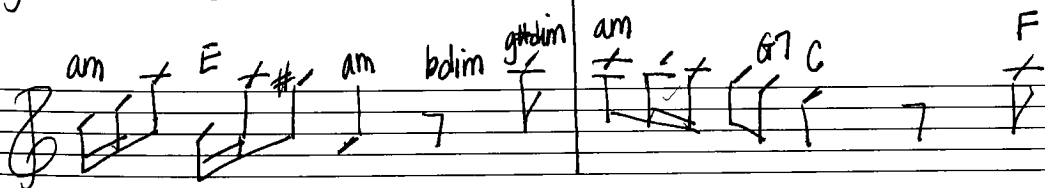
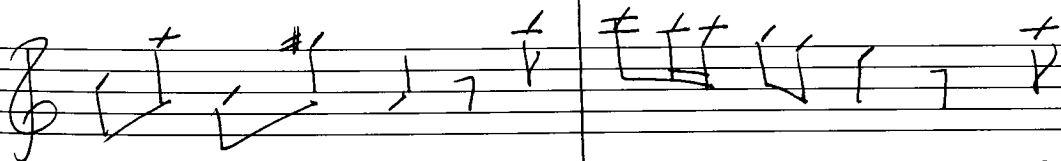
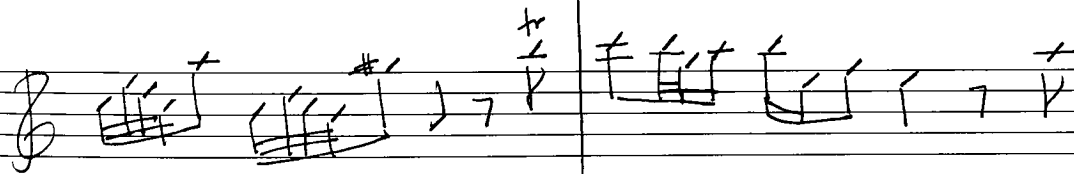
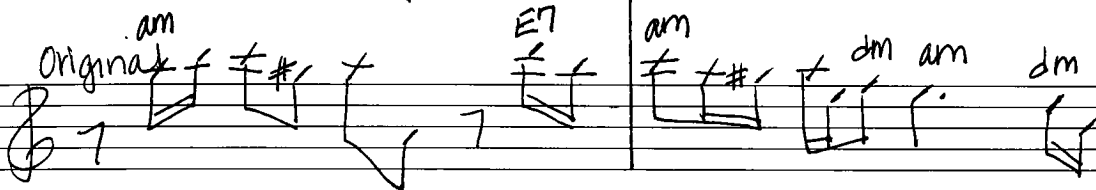
Variation 2



Framework



Original



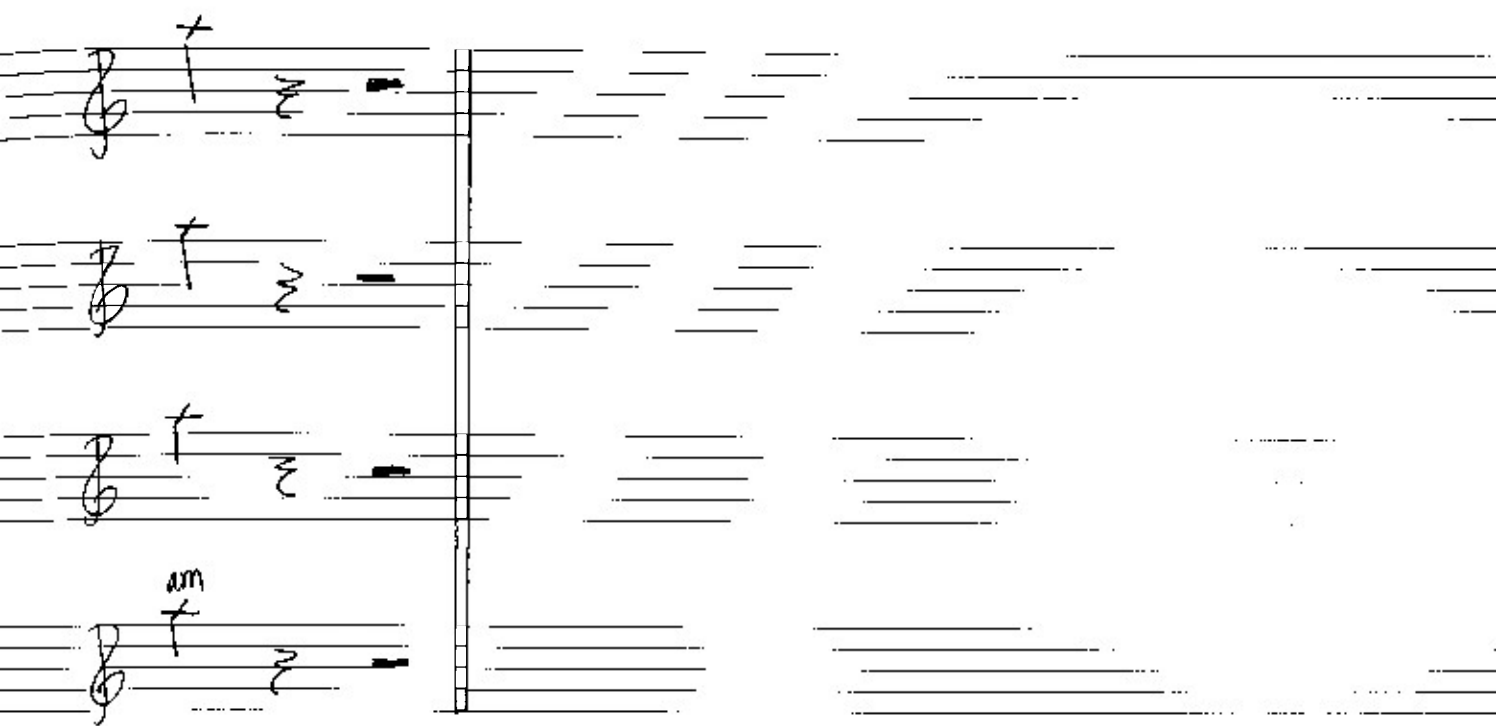


Handwritten musical notation on four staves. The first three staves show melodic lines with various chords and accidentals. The fourth staff includes chord labels: C, F#dim, G, F, C, G, G7.

Handwritten musical notation on four staves. The first three staves show melodic lines with various chords and accidentals. The fourth staff includes chord labels: C, dm, G, C, G, bdim, C, g#dim.

Handwritten musical notation for the first system, consisting of four staves. The first three staves are treble clef and contain complex chordal textures with many beamed notes. The fourth staff contains chord labels: am, E7, am, dm, am, dm, am, E.

Handwritten musical notation for the second system, consisting of four staves. The first three staves continue the complex chordal textures. The fourth staff contains chord labels: am, F, E, am, dm, E, am, E behind E7, am.



```

[
[[am],[e],[r]], [[am],[s],[a1]], [[am],[s],[b1]], [[am],[e],[c2]],
[[am],[e],[gs1]], [[am],[e],[a1]], [[am],[e],[a]], [[am],[e],[r]],
[[e7],[s],[d2]], [[e7],[s],[b1]], [[am],[e],[c2]], [[am],[s],[a1]], [[am],[s],[gs1]],
[[am],[s],[a1]], [[am],[s],[e1]], [[dm],[e],[f1]], [[am],[dq],[e1]],
[[dm],[s],[d1]], [[dm],[s],[b]], [[am],[s],[c1]], [[am],[s],[e1]], [[am],[e],[a1]],
[[e],[s],[b]], [[e],[s],[a1]], [[e],[e],[gs1]], [[am],[q],[a]], [[bdim],[e],[r]],
[[gsdim],[e],[b1]], [[am],[e],[c2]], [[am],[s],[b1]], [[am],[s],[a1]],
[[am],[e],[g1]], [[g7],[e],[f1]], [[c],[q],[e1]], [[c],[e],[r]], [[f],[e],[a1]],
[[c],[e],[g1]], [[c],[s],[a1]], [[c],[s],[b1]], [[c],[s],[c2]], [[c],[s],[b1]],
[[fsdim],[s],[c2]], [[fsdim],[s],[d2]], [[g],[s],[b1]], [[g],[s],[a1]],
[[g],[e],[g1]], [[g],[e],[r]], [[g],[e],[a1]], [[g],[s],[g1]], [[g],[s],[fs1]],
[[g],[e],[g1]], [[g],[e],[g]], [[f],[e],[f1]], [[c],[s],[e1]], [[c],[s],[d1]],
[[c],[s],[e1]], [[c],[s],[c1]], [[g],[e],[d1]], [[g7],[e],[b]], [[c],[q],[c1]],
[[dm],[e],[r]], [[g],[s],[g1]], [[g],[s],[d1]], [[c],[s],[e1]], [[c],[s],[g1]],
[[c],[s],[c2]], [[c],[s],[d2]], [[c],[s],[e2]], [[c],[s],[c2]], [[g],[e],[d2]],
[[c],[e],[c2]], [[c],[e],[c1]], [[c],[e],[r]], [[bdim],[e],[b]], [[c],[s],[c1]],
[[c],[s],[e1]], [[c],[s],[a1]], [[c],[s],[b1]], [[c],[e],[c2]], [[gsdim],[e],[gs1]],
[[am],[e],[a1]], [[am],[e],[a]], [[am],[e],[r]], [[e7],[s],[d2]], [[e7],[s],[b1]],
[[am],[e],[c2]], [[am],[s],[a1]], [[am],[s],[gs1]], [[am],[s],[a1]], [[am],[s],[e1]],
[[dm],[e],[f1]], [[am],[dq],[e1]], [[dm],[s],[d1]], [[dm],[s],[b]],
[[am],[s],[c1]], [[am],[s],[e1]], [[am],[e],[a1]], [[e],[s],[b]], [[e],[s],[a1]],
[[e],[e],[gs1]], [[am],[e],[a]], [[f],[q],[f1]], [[e],[s],[e1]], [[e],[s],[d1]],
[[am],[e],[c1]], [[dm],[e],[a1]], [[e],[s],[gs1]], [[e],[s],[a1]], [[e],[e],[b1]],
[[am],[e],[e1]], [[am],[s],[c2]], [[am],[s],[a1]], [[e],[e],[b1]],
[[bdim],[e],[d2]], [[e7],[ts],[d2]], [[e7],[s],[b1]], [[am],[s],[c2]],
[[am],[s],[a1]], [[e],[de],[gs1]], [[e],[s],[a1]], [[am],[q],[a1]],
[[am],[q],[r]], [[am],[h],[r]]
].

```

```
[
[[am],[e],[r]],[[am],[e],[a1]],[[am],[e],[c2]],
[[am],[e],[gs1]],[[am],[e],[a1]],[[am],[e],[a]],[[am],[e],[r]],
[[e7],[s],[d2]],[[e7],[s],[b1]],[[am],[e],[c2]],[[am],[s],[a1]],[[am],[s],[gs1]],
[[am],[e],[a1]],[[dm],[e],[f1]],[[am],[dq],[e1]],
[[dm],[s],[d1]],[[dm],[s],[b]],[[am],[e],[c1]],[[am],[e],[a1]],
[[e],[e],[b]],[[e],[e],[gs1]],[[am],[q],[a]],[[bdim],[e],[r]],
[[gsdim],[e],[b1]],[[am],[e],[c2]],[[am],[s],[b1]],[[am],[s],[a1]],
[[am],[e],[g1]],[[g7],[e],[f1]],[[c],[q],[e1]],[[c],[e],[r]],[[f],[e],[a1]],
[[c],[q],[g1]],[[c],[e],[c2]],
[[fsdim],[e],[c2]],[[g],[e],[b1]],
[[g],[e],[g1]],[[g],[e],[r]],[[g],[e],[a1]],[[g],[e],[g1]],
[[g],[e],[g1]],[[g],[e],[g]],[[f],[e],[f1]],[[c],[e],[e1]],
[[c],[e],[c1]],[[g],[e],[d1]],[[g7],[e],[b]],[[c],[q],[c1]],
[[dm],[e],[r]],[[g],[e],[g1]],[[c],[e],[e1]],
[[c],[e],[c2]],[[c],[e],[e2]],[[g],[e],[d2]],
[[c],[e],[c2]],[[c],[e],[c1]],[[c],[e],[r]],[[bdim],[e],[b]],[[c],[e],[c1]],
[[c],[e],[a1]],[[c],[e],[c2]],[[gsdim],[e],[gs1]],
[[am],[e],[a1]],[[am],[e],[a]],[[am],[e],[r]],[[e7],[s],[d2]],[[e7],[s],[b1]],
[[am],[e],[c2]],[[am],[e],[a1]],[[am],[e],[a1]],
[[dm],[e],[f1]],[[am],[dq],[e1]],[[dm],[e],[d1]],
[[am],[e],[c1]],[[am],[e],[a1]],[[e],[e],[b]],
[[e],[e],[gs1]],[[am],[e],[a]],[[f],[q],[f1]],[[e],[e],[e1]],
[[am],[e],[c1]],[[dm],[e],[a1]],[[e],[e],[gs1]],[[e],[e],[b1]],
[[am],[e],[e1]],[[am],[e],[c2]],[[e],[e],[b1]],
[[bdim],[e],[d2]],[[e7],[ts],[d2]],[[e7],[s],[b1]],[[am],[s],[c2]],
[[am],[s],[a1]],[[e],[de],[gs1]],[[e],[s],[a1]],[[am],[q],[a1]],
[[am],[q],[r]],[[am],[h],[r]]
].
```

?- v(Variation).

```
Variation = [[am],[e],[r]], [[am,am],[s,s,e],[a1,b1,c2]],
[[am,am],[s,s,e],[gs1,tr11,a1]], [[am],[e],[a]],
[[am],[e],[r]], [[e7,e7],[th,th,s],[d2,a1,b1]],
[[am,am],[s,s,s],[c2,g1,a1]],
[[am,am],[s,e],[gs1,tr11,a1]], [[dm],[e],[f1]],
[[am],[dq],[e1]],
[[dm,dm],[sf,sf,sf,sf,th,th],[d1,d1,c1,b,a,b]],
[[am,am],[s,s,s,s],[e1,ds1,c1,a1]],
[[e,e],[s,s,s,s],[ds1,cs1,b,gs1]], [[am],[q],[a]],
[[bdim],[e],[r]], [[gsdim,am],[s,s,e],[b1,tr11,c2]],
[[am,am,am,g7,c],[th,th,s,s,s,e,q],[b1,c2,a1,g1,
a1,f1,e1]],

[[c],[e],[r]], [[f],[e],[a1]],
[[c,c],[s,s,e,e],[g1,g2,e2,c2]], [[fsdim],[e],[c2]],
[[g,g],[th,th,th,th,s,s],[b1,b1,a1,g1,f1,g1]],
[[g],[e],[r]], [[g],[e],[a1]], [[g],[e],[g1]],
[[g,g],[s,s,s,s],[g1,f1,g1,g1]], [[f],[e],[f1]],
[[c,c,g,g7,c],[e,s,s,e,s,s,q],[e1,c1,e1,d1,b,d1,
c1]], [[dm],[e],[r]],

[[g,c],[s,s,s,s],[a1,g1,f1,e1]],
[[c,c],[s,s,e],[c2,d2,e2]], [[g],[e],[d2]],
[[c,c],[s,s,s,s],[c2,b1,c2,c1]], [[c],[e],[r]],
[[bdim,c],[s,s,e],[b,tr11,c1]],
[[c,c],[s,s,s,s],[a1,d2,e1,c2]],
[[gsdim,am],[s,s,e],[gs1,tr11,a1]], [[am],[e],[a]],
[[am],[e],[r]], [[e7,e7],[th,th,th,th],[e2,d2,c2,
b1]],

[[am,am],[s,s,s,s],[c2,a1,g1,a1]],
[[am,dm],[s,s,s,s],[a1,f1,e1,f1]], [[am],[dq],[e1]],
[[dm],[e],[d1]], [[am,am],[th,th,th,th,e],
[c1,e1,d1,c1,a1]],
[[e,e],[th,th,th,th,e],[b,d1,c1,b,gs1]],
[[am,f],[s,s,t,t,t],[a,c1,b,a,f1]],
[[e,am],[s,s,s,s],[e1,c1,b,c1]], [[dm],[e],[a1]],
[[e,e],[s,s,s,s],[gs1,c2,d1,b1]],
[[am,am],[th,th,th,th,e],[e1,g1,f1,e1,c2]],
[[e,bdim],[s,s,s,s],[b1,e2,e2,d2]],
[[e7,e7],[tth,th,th,th],[d2,b1,a1,b1]],
[[am,am],[th,th,s],[c2,g1,a1]],
[[e,e],[de,s],[gs1,tr11,a1]], [[am],[q],[a1]],
[[am],[q],[r]], [[am],[h],[r]]]
```

?-

?- v(Variation).

```
Variation = [[ [am], [e], [r]], [[am,am], [s,s,s,s], [a1,d2,e1,c2]],
[[am,am], [s,s,e], [gs1,trill,a1]], [[am], [e], [a]],
[[am], [e], [r]], [[e7,e7], [th,th,th,th], [d2,b1,a1,
b1]],

[[am,am], [s,th,th,s], [c2,a1,g1,a1]],
[[am,am], [s,s,s], [gs1,trill,a1]], [[dm], [e], [f1]],
[[am], [dq], [e1]],
[[dm,dm], [sf,sf,sf,sf,th,th], [d1,d1,c1,b,a,b]],
[[am,am], [th,th,th,th,e], [c1,e1,d1,c1,a1]],
[[e,e], [th,th,th,th,e], [b,d1,c1,b,gs1]],
[[am], [q], [a]], [[bdim], [e], [r]],
[[gsdim,am], [s,s,e], [b1,trill,c2]],
[[am,am,am,g7,c], [th,th,s,s,s,e,q], [b1,g1,a1,b1,
e1,f1,e1]],

[[c], [e], [r]], [[f], [e], [a1]],
[[c,c], [e,tth,th,th,th,e], [g1,b1,a1,g1,c2]],
[[fsdim], [e], [c2]],
[[g,g], [th,th,th,th,s,s], [b1,b1,a1,g1,f1,g1]],
[[g], [e], [r]], [[g], [e], [a1]], [[g], [e], [g1]],
[[g,g], [s,s,s,s], [g1,f1,g1,g]], [[f], [e], [f1]],
[[c,c,g,g7,c], [e,s,s,e,s,s,q], [e1,c1,e1,d1,b,d1,
c1]], [[dm], [e], [r]],

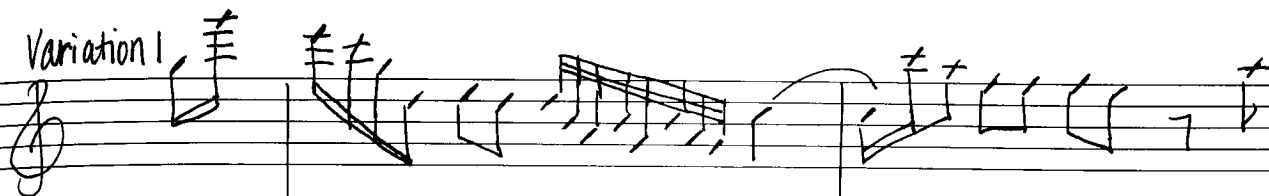
[[g,c], [s,s,e], [g1,d1,e1]], [[c,c], [s,s,s,s],
[c2,f2,g1,e2]],

[[g], [e], [d2]], [[c,c], [s,s,s,s], [c2,b1,c2,c1]],
[[c], [e], [r]], [[bdim,c], [s,s,e], [b,trill,c1]],
[[c,c], [s,s,e], [a1,b1,c2]],
[[gsdim,am], [s,s,e], [gs1,trill,a1]], [[am], [e], [a]],
[[am], [e], [r]], [[e7,e7], [th,th,s], [d2,a1,b1]],
[[am,am], [s,s,s,s], [d2,c2,b1,a1]],
[[am,dm], [s,s,s,s], [b1,a1,g1,f1]], [[am], [dq], [e1]],
[[dm], [e], [d1]], [[am,am], [s,s,s,s], [e1,ds1,c1,a1]],
[[e,e], [s,s,s,s], [ds1,cs1,b,gs1]],
[[am,f], [e,s,s,e], [c1,b,a,f1]],
[[e,am], [th,th,th,th,s,s], [e1,e1,d1,c1,b,c1]],
[[dm], [e], [a1]], [[e,e], [s,s,e], [gs1,a1,b1]],
[[am,am], [s,s,s,s], [gs1,fs1,e1,c2]],
[[e,bdim], [s,s,s,s], [b1,e2,e2,d2]],
[[e7,e7], [tsf,sf,sf,sf,th,th], [d2,d2,c2,b1,a1,b1]],
[[am,am], [th,th,th,th], [c2,a1,g1,a1]],
[[e,e], [de,s], [gs1,trill,a1]], [[am], [q], [a1]],
[[am], [q], [r]], [[am], [h], [r]]]
```

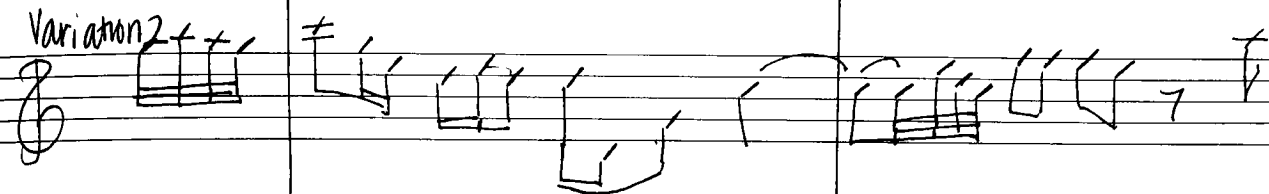
?-

# Telemann - C

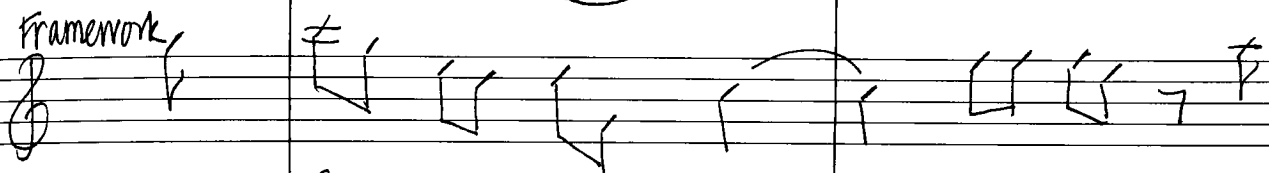
## Variation 1



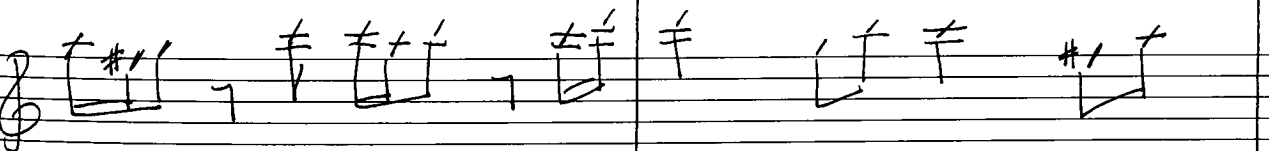
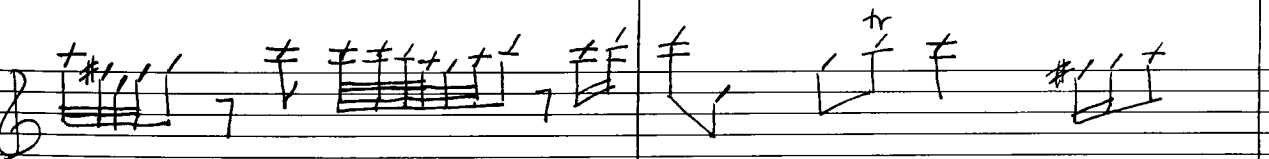
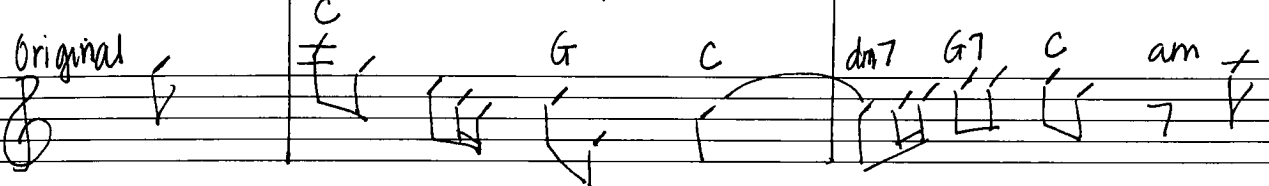
## Variation 2



## Framework



## Original





Handwritten musical notation for a four-staff system. The first three staves are treble clefs with various chords and melodic lines. The fourth staff includes chord symbols: G, C, D7, G, D, G, and G7.

Handwritten musical notation for a four-staff system. The first three staves are treble clefs with various chords and melodic lines. The fourth staff includes chord symbols: C, F, gm, F, dm, and D7.

Handwritten musical notation for the first system, consisting of four staves. The first three staves are treble clef, and the fourth is bass clef. The notation includes various chords and melodic lines with a triplet in the first staff.

Handwritten musical notation for the second system, consisting of four staves. The first three staves are treble clef, and the fourth is bass clef. The notation includes various chords and melodic lines with a triplet in the first staff.

Handwritten musical score for guitar, consisting of four staves. The first three staves contain melodic lines with various chords and a trill. The fourth staff contains a bass line with chord labels: *em*, *am*, *dm7*, *G7*, *C*, *G7*, *C*, *G*, *C*. The music is written in treble clef with a key signature of one flat and a 4/4 time signature.

```
[
[[c],[e],[g1]],[[c],[e],[c2]],[[c],[e],[g1]],[[c],[e],[e1]],
[[c],[s],[d1]],[[c],[s],[c1]],[[g],[e],[d1]],[[g],[e],[g]],
[[c],[q],[c1]],[[dm7],[te],[c1]],[[dm7],[s],[d1]],[[dm7],[s],[e1]],
[[g7],[e],[f1]],[[g7],[e],[f1]],[[c],[e],[f1]],[[c],[e],[e1]],
[[c],[e],[r]],[[am],[e],[a1]],[[bm7],[s],[a1]],[[bm7],[s],[fs1]],
[[bm7],[e],[g1]],[[am7],[e],[r]],[[am7],[e],[c2]],[[g],[s],[c2]],
[[g],[s],[a1]],[[g],[e],[b1]],[[g],[e],[r]],[[am],[s],[c2]],
[[am],[th],[d2]],[[am],[th],[e2]],[[bm7],[e],[d2]],[[bm7],[e],[fs1]],
[[em7],[e],[g1]],[[em7],[s],[b1]],[[em7],[th],[c2]],[[em7],[th],[d2]],
[[am7],[e],[c2]],[[am7],[e],[e1]],[[d7],[e],[fs1]],[[d7],[s],[a1]],
[[d7],[th],[b1]],[[d7],[th],[c2]],[[g],[s],[b1]],[[g],[s],[a1]],
[[c],[q],[g1]],[[d7],[s],[a1]],[[d7],[s],[fs1]],[[g],[de],[g1]],
[[g],[s],[a1]],[[d],[de],[a1]],[[d],[s],[g1]],
[[g],[q],[g1]],[[g],[e],[r]],[[g],[e],[d2]],[[g],[e],[g2]],
[[g],[e],[d2]],[[g7],[e],[b1]],[[g7],[s],[a1]],[[g7],[s],[g1]],
[[c],[e],[c2]],[[c],[e],[c1]],[[c],[e],[r]],[[c],[e],[bf1]],
[[f],[e],[a1]],[[f],[s],[g1]],[[f],[s],[f1]],[[gm],[e],[g1]],
[[gm],[s],[f1]],[[gm],[s],[e1]],[[f],[e],[f1]],[[f],[e],[f]],[[f],[e],[r]],
[[f],[e],[a1]],[[dm],[e],[d2]],[[dm],[e],[a1]],[[d7],[e],[fs1]],
[[d7],[s],[e1]],[[d7],[s],[d1]],[[g],[e],[g1]],[[g],[e],[g]],[[g7],[e],[r]],
[[g7],[e],[f1]],[[c],[e],[e1]],[[c],[s],[d1]],[[c],[s],[c1]],
[[g7],[e],[f1]],[[c],[e],[e1]],[[c],[e],[e1]],[[g],[e],[d1]],
[[c],[e],[r]],[[c],[e],[g1]],[[f],[e],[a1]],[[f],[q],[f1]],[[dm],[s],[e1]],
[[dm],[s],[d1]],[[g],[e],[b1]],[[g],[q],[g1]],[[em],[s],[f1]],
[[em],[s],[e1]],[[am],[s],[c2]],[[am],[s],[b1]],[[am],[q],[c2]],
[[g7],[e],[f2]],[[c],[e],[e2]],[[c],[s],[d2]],[[c],[s],[c2]],[[g],[e],[d2]],
[[g],[s],[c2]],[[g],[s],[b1]],[[c],[e],[c2]],[[c],[e],[c1]],[[dm],[e],[r]],
[[dm],[s],[f1]],[[dm],[th],[g1]],[[dm],[th],[a1]],[[em],[e],[g1]],
[[em],[e],[b]],[[am],[e],[c1]],[[am],[s],[e1]],[[am],[th],[f1]],
[[am],[th],[g1]],[[dm7],[e],[f1]],[[dm7],[e],[a]],[[g7],[e],[b]],
[[g7],[s],[d1]],[[g7],[th],[e1]],[[g7],[th],[f1]],[[c],[s],[e1]],
[[c],[s],[d1]],[[c],[q],[c1]],[[g7],[s],[d1]],[[g7],[s],[b]],
[[c],[de],[c1]],[[c],[s],[d1]],[[g],[de],[d1]],[[g],[s],[c1]],
[[c],[q],[c1]],[[c],[q],[r]],[[c],[h],[r]]
].
```

```

[
[[c],[e],[g1]],[[c],[e],[c2]],[[c],[e],[g1]],[[c],[e],[e1]],
[[c],[e],[d1]],[[g],[e],[d1]],[[g],[e],[g]],
[[c],[q],[c1]],[[dm7],[tq],[c1]],
[[g7],[e],[f1]],[[g7],[e],[f1]],[[c],[e],[f1]],[[c],[e],[e1]],
[[c],[e],[r]],[[am],[e],[a1]],[[bm7],[s],[a1]],[[bm7],[s],[fs1]],
[[bm7],[e],[g1]],[[am7],[e],[r]],[[am7],[e],[c2]],[[g],[s],[c2]],
[[g],[s],[a1]],[[g],[e],[b1]],[[g],[e],[r]],[[am],[s],[c2]],
[[am],[s],[d2]],[[bm7],[q],[d2]],
[[em7],[e],[g1]],[[em7],[e],[b1]],
[[am7],[q],[c2]],[[d7],[e],[fs1]],[[d7],[e],[a1]],
[[g],[e],[b1]],
[[c],[q],[g1]],[[d7],[e],[a1]],[[g],[de],[g1]],
[[g],[s],[a1]],[[d],[de],[a1]],[[d],[s],[g1]],
[[g],[q],[g1]],[[g],[e],[r]],[[g],[e],[d2]],[[g],[e],[g2]],
[[g],[e],[d2]],[[g7],[e],[b1]],[[g7],[e],[a1]],
[[c],[e],[c2]],[[c],[e],[c1]],[[c],[e],[r]],[[c],[e],[bf1]],
[[f],[e],[a1]],[[f],[e],[g1]],[[gm],[e],[g1]],
[[gm],[e],[f1]],[[f],[e],[f1]],[[f],[e],[f]],[[f],[e],[r]],
[[f],[e],[a1]],[[dm],[e],[d2]],[[dm],[e],[a1]],[[d7],[e],[fs1]],
[[d7],[e],[e1]],[[g],[e],[g1]],[[g],[e],[g]],[[g7],[e],[r]],
[[g7],[e],[f1]],[[c],[e],[e1]],[[c],[e],[d1]],
[[g7],[e],[f1]],[[c],[e],[e1]],[[c],[e],[e1]],[[g],[e],[d1]],
[[c],[e],[r]],[[c],[e],[g1]],[[f],[e],[a1]],[[f],[q],[f1]],
[[dm],[e],[d1]],[[g],[e],[b1]],[[g],[q],[g1]],
[[em],[e],[e1]],[[am],[e],[c2]],[[am],[q],[c2]],
[[g7],[e],[f2]],[[c],[e],[e2]],[[c],[e],[d2]],[[g],[e],[d2]],
[[g],[e],[c2]],[[c],[e],[c2]],[[c],[e],[c1]],[[dm],[e],[r]],
[[dm],[s],[f1]],[[dm],[th],[g1]],[[dm],[th],[a1]],[[em],[q],[g1]],
[[am],[e],[c1]],[[am],[e],[e1]],
[[dm7],[q],[f1]],[[g7],[e],[b]],
[[g7],[e],[d1]],[[c],[e],[e1]],
[[c],[q],[c1]],[[g7],[e],[d1]],
[[c],[de],[c1]],[[c],[s],[d1]],[[g],[de],[d1]],[[g],[s],[c1]],
[[c],[q],[c1]],[[c],[q],[r]],[[c],[h],[r]]
].

```

?- v(Variation).

```
Variation = [[c,c],[s,s,s,s],[g1,g2,e2,c2]],
[[c,c],[s,s,e],[g1,d1,e1]],[[c],[e],[d1]],
[[g,g],[th,th,th,th,th,th,th,th],[d1,b,a,b,g,b,a,
g]],
[[c],[q],[c1]],[[dm7,g7],[ts,s,e,e],[c1,c2,a1,f1]],
[[g7],[e],[f1]],[[c],[e],[f1]],[[c],[e],[e1]],
[[c],[e],[r]],[[am],[e],[a1]],
[[bm7,bm7],[th,th,s],[a1,e1,fs1]],[[bm7],[e],[g1]],
[[am7],[e],[r]],[[am7],[e],[c2]],
[[g,g],[th,th,th,th],[c2,a1,g1,a1]],[[g],[e],[b1]],
[[g],[e],[r]],[[am],[s],[c2]],[[am],[s],[d2]],
[[bm7,em7],[e,th,th,th,th,th,th,s],[d2,b1,a1,b1,
g1,b1,a1,g1]],
[[em7,am7],[s,s,q],[b1,trill,c2]],
[[d7,d7],[s,s,s,s],[fs1,b1,b1,a1]],
[[g,c],[e,s,s,e],[b1,g1,f1,g1]],[[d7],[e],[a1]],
[[g],[de],[g1]],[[g],[s],[a1]],[[d],[de],[a1]],
[[d],[s],[g1]],[[g],[q],[g1]],[[g],[e],[r]],
[[g,g],[th,th,th,th,e],[d2,f2,e2,d2,g2]],
[[g,g7],[th,th,th,th,s,s],[d2,d2,c2,b1,a1,b1]],
[[g7,c],[s,s,s,s],[a1,d2,d2,c2]],[[c],[e],[c1]],
[[c],[e],[r]],[[c],[e],[bf1]],[[f],[e],[a1]],
[[f],[e],[g1]],[[gm],[e],[g1]],[[gm],[e],[f1]],
[[f,f],[s,s,s,s],[f1,c1,a,f]],[[f],[e],[r]],
[[f,dm],[th,th,th,th,e],[a1,c2,b1,a1,d2]],
[[dm,d7],[th,th,th,th,s,s],[a1,a1,g1,f1,e1,f1]],
[[d7,g],[s,s,s,s],[e1,a1,b,g1]],[[g],[e],[g]],
[[g7],[e],[r]],[[g7],[e],[f1]],[[c],[e],[e1]],
[[c,g7],[s,s,s,s],[d1,g1,a,f1]],[[c],[e],[e1]],
[[c],[e],[e1]],[[g],[e],[d1]],[[c],[e],[r]],
[[c],[e],[g1]],[[f,f,dm],[st,st,st,de,s,e],
g1,a1,g1,f1,e1,d1]],
[[g,g,em],[st,st,st,de,s,e],[a1,b1,a1,g1,f1,e1]],
[[am],[e],[c2]],[[am,g7],[s,s,s,s,s,s],
c2,f2,g2,f2,e2,f2]],
[[c],[e],[e2]],[[c],[e],[d2]],[[g],[e],[d2]],
[[g],[e],[c2]],[[c,c],[s,s,s,s],[c2,g1,e1,c1]],
[[dm],[e],[r]],[[dm],[s],[f1]],[[dm],[th],[g1]],
[[dm],[th],[a1]],
[[em,am],[e,tth,th,th,th,e],[g1,e1,c1,c1,c1]],
[[am,dm7],[s,s,q],[e1,trill,f1]],
[[g7,g7],[s,s,e],[b,c1,d1]],
[[c,c],[e,s,s,e],[f1,e1,d1,c1]],[[g7],[e],[d1]],
[[c],[de],[c1]],[[c],[s],[d1]],[[g],[de],[d1]],
[[g],[s],[c1]],[[c],[q],[c1]],[[c],[q],[r]],
[[c],[h],[r]]]
```

?-

```

v = [[c,c],[th,th,th,th,e],[g1,b1,a1,g1,c2]],
      [[c,c],[s,s,s,s],[g1,e1,d1,e1],[[c],[e],[d1]],
      [[g,g],[s,s,e],[d1,d,g],[[c],[q],[c1]],
      [[dm7,g7],[te,tth,th,th,th,e],[c1,e1,d1,c1,f1]],
      [[g7],[e],[f1]],[[c],[e],[f1]],[[c],[e],[e1]],[[c],[e],
      [r]],
      [[am],[e],[a1]],[[bm7,bm7],[th,th,th,th],[a1,fs1,e1,fs1]],
      [[bm7],[e],[g1]],[[am7],[e],[r]],[[am7],[e],[c2]],
      [[g,g],[sf,sf,sf,sf,th,th],[c2,c2,b1,a1,g1,a1]],
      [[g],[e],[b1]],[[g],[e],[r]],[[am],[s],[c2]],
      [[am],[s],[d2]],[[bm7,em7],[e,e,e],[d2,d1,g1]],
      [[em7,am7],[s,s,q],[b1,trill,c2]],[[d7,d7],[s,s,e],
      [fs1,g1,a1]],
      [[g,c],[s,s,s,s,s,s],[b1,b1,a1,g1,f1,g1]],[[d7],[e],[a1]],
      [[g],[de],[g1]],[[g],[s],[a1]],[[d],[de],[a1]],
      [[d],[s],[g1]],[[g],[q],[g1]],[[g],[e],[r]],
      [[g,g],[th,th,th,th,e],[d2,f2,e2,d2,g2]],
      [[g,g7],[s,s,s,s],[e2,d2,c2,b1]],[[g7,c],[s,s,e],[a1,b1,
      c2]],
      [[c],[e],[c1]],[[c],[e],[r]],[[c],[e],[bf1]],[[f],[e],
      [a1]],
      [[f],[e],[g1]],[[gm],[e],[g1]],[[gm],[e],[f1]],
      [[f,f],[s,s,s,s],[f1,c1,a,f]],[[f],[e],[r]],
      [[f,dm],[th,th,th,th,e],[a1,c2,b1,a1,d2]],
      [[dm,d7],[s,s,s,s],[b1,a1,g1,f1]],[[d7,g],[s,s,s,s],
      [e1,a1,b,g1]],
      [[g],[e],[g]],[[g7],[e],[r]],[[g7],[e],[f1]],[[c],[e],
      [e1]],
      [[c,g7],[s,s,s,s],[d1,g1,a,f1]],[[c],[e],[e1]],
      [[c],[e],[e1]],[[g],[e],[d1]],[[c],[e],[r]],[[c],[e],[g1]],
      [[f,f,dm],[s,ts,s,s,s,s,s,s],[a1,a1,b1,g1,f1,g1,e1,d1]],
      [[g,g,em],[s,ts,s,s,s,s,s,s],[b1,b1,c2,a1,g1,a1,fs1,e1]],
      [[am],[e],[c2]],[[am,g7],[s,s,s,s,s,s],[c2,f2,g2,f2,e2,f2]],
      [[c],[e],[e2]],[[c],[e],[d2]],[[g],[e],[d2]],[[g],[e],
      [c2]],
      [[c,c],[s,s,s,s],[c2,g1,e1,c1]],[[dm],[e],[r]],
      [[dm],[s],[f1]],[[dm],[th],[g1]],[[dm],[th],[a1]],
      [[em,am],[e,tth,th,th,th,e],[g1,e1,c1,c1,c1]],
      [[am,dm7],[s,s,q],[e1,trill,f1]],[[g7,g7],[s,s,s,s],
      [b,e1,e1,d1]],
      [[c,c],[s,s,q],[e1,b,c1]],[[g7],[e],[d1]],[[c],[de],[c1]],
      [[c],[s],[d1]],[[g],[de],[d1]],[[g],[s],[c1]],
      [[c],[q],[c1]],[[c],[q],[r]],[[c],[h],[r]]]

```

?-

# Telemann - gm

Handwritten musical notation for Variation 1, Variation 2, Framework, and Original, showing different melodic and harmonic treatments of a Telemann piece in G minor.

**Variation 1:** Features a melodic line with many accidentals (sharps and naturals) and a complex, dense texture in the lower voice.

**Variation 2:** Similar to Variation 1, but with a different melodic contour and a slightly different lower voice texture.

**Framework:** A simplified version of the piece, focusing on the basic harmonic structure with fewer accidentals.

**Original:** The original notation, including figured bass (am, bdm, E, am, dm, E, am, E) and a clear melodic line.

Handwritten musical notation for a second system, featuring triplets and various accidentals, continuing the variations on the Telemann piece.

The notation includes triplets (marked with '3') and various accidentals (sharps, naturals, and flats) across the melodic and harmonic lines.

The figured bass line includes: F, am, B, em, B, em, sm, A, dm, A.



Handwritten musical notation for a 4-measure piece in treble clef. The notation includes various chords and a trill (tr). The first measure contains a chord with a sharp sign (#). The second measure contains a trill (tr) over a chord. The third measure contains a chord with a sharp sign (#). The fourth measure contains a chord with a sharp sign (#). The notation is written on four staves.

Chords: *dm gm A dm A7 dm D7 G C*

Handwritten musical notation for a 4-measure piece in treble clef. The notation includes a triplet (3) over a chord in the first measure. The second measure contains a chord with a sharp sign (#). The third measure contains a chord with a sharp sign (#). The fourth measure contains a chord with a sharp sign (#). The notation is written on four staves.

Chords: *C dm G C E7 am. bdim E A*

Handwritten musical notation for a 4-measure piece. The first two measures are in treble clef with a key signature of one sharp (F#). The first measure contains a complex sixteenth-note pattern with a trill (tr) over the final note. The second measure contains a similar pattern. The third and fourth measures are in bass clef and contain simpler chords. Chord labels are written above the notes: dm, E, am, E, am, A7, dm, A, dm, D7, G, E7.

Handwritten musical notation for a 4-measure piece. The first two measures are in treble clef with a key signature of one sharp (F#). The first measure contains a complex sixteenth-note pattern with a trill (tr) over the final note. The second measure contains a similar pattern. The third and fourth measures are in bass clef and contain simpler chords. Chord labels are written above the notes: am, bdim, E, am, dm, E, am, E, am, A7, dm, A, dm, D7, G, E7, am, bdim, E.



```
[
[[am],[e],[e1]],[[am],[dq],[a1]],[[bdim],[s],[b1]],[[bdim],[s],[c2]],
[[e],[q],[gs1]],[[e],[e],[r]],[[am],[s],[a1]],[[am],[s],[e1]],
[[dm],[e],[f1]],[[dm],[e],[d1]],[[e],[e],[b]],[[e],[e],[e1]],[[am],[e],[c1]],
[[am],[e],[a]],[[am],[e],[r]],[[e],[e],[e1]],[[f],[e],[f1]],[[f],[s],[f1]],
[[f],[s],[g1]],[[f],[e],[a1]],[[am],[e],[e1]],[[b],[e],[ds1]],[[em],[e],[e1]],
[[em],[e],[r]],[[b],[e],[b]],[[em],[e],[e1]],[[em],[s],[e1]],[[em],[s],[f1]],
[[em],[e],[g1]],[[gm],[e],[d1]],[[a],[e],[cs1]],[[dm],[e],[d1]],[[dm],[e],[r]],
[[a],[e],[a1]],[[dm],[e],[f1]],[[gm],[s],[e1]],[[gm],[s],[d1]],[[a],[e],[a]],
[[a],[e],[cs1]],[[dm],[q],[d1]],[[dm],[e],[r]],[[a7],[e],[a]],[[dm],[s],[d1]],
[[dm],[s],[e1]],[[dm],[s],[f1]],[[dm],[s],[e1]],[[d7],[e],[d1]],
[[d7],[e],[c1]],[[g],[e],[b]],[[c],[e],[c1]],[[c],[e],[r]],[[c],[e],[g1]],
[[c],[e],[e1]],[[dm],[s],[d1]],[[dm],[s],[c1]],[[g],[e],[g]],[[g],[e],[b]],
[[c],[q],[c1]],[[c],[e],[r]],[[e7],[e],[e1]],[[am],[dq],[a1]],
[[bdim],[s],[b1]],[[bdim],[s],[c2]],[[e],[q],[gs1]],[[e],[e],[r]],
[[a],[s],[a1]],[[a],[s],[e1]],[[dm],[e],[f1]],[[dm],[e],[d1]],[[e],[e],[b]],
[[e],[e],[e1]],[[am],[e],[c1]],[[am],[e],[a]],[[am],[e],[r]],[[e],[e],[b]],
[[am],[e],[c1]],[[a7],[e],[cs1]],[[dm],[e],[d1]],[[a],[e],[e1]],[[dm],[e],[f1]],
[[d7],[e],[fs1]],[[g],[e],[g1]],[[e7],[e],[gs1]],[[am],[s],[a1]],
[[am],[s],[c2]],[[bdim],[q],[b1]],[[e],[e],[e1]],[[am],[e],[c1]],
[[dm],[s],[b]],[[dm],[s],[a]],[[e],[de],[gs]],[[e],[s],[a]],[[am],[q],[a]],
[[am],[e],[r]],[[e],[e],[b]],
[[am],[e],[c1]],[[a7],[e],[cs1]],[[dm],[e],[d1]],[[a],[e],[e1]],[[dm],[e],[f1]],
[[d7],[e],[fs1]],[[g],[e],[g1]],[[e7],[e],[gs1]],[[am],[s],[a1]],
[[am],[s],[c2]],[[bdim],[q],[b1]],[[e],[e],[e1]],[[am],[e],[c1]],
[[dm],[s],[b]],[[dm],[s],[a]],[[e],[de],[gs]],[[e],[s],[a]],[[am],[q],[a]],
[[em],[q],[e1]],[[f],[tq],[e1]],[[dm],[q],[d1]],[[e],[h],[e1]]
].
```

```
[
[[am],[e],[e1]],[[am],[dq],[a1]],[[bdim],[e],[b1]],
[[e],[q],[gs1]],[[e],[e],[r]],[[am],[e],[a1]],
[[dm],[e],[f1]],[[dm],[e],[d1]],[[e],[e],[b]],[[e],[e],[e1]],[[am],[e],[c1]],
[[am],[e],[a]],[[am],[e],[r]],[[e],[e],[e1]],[[f],[q],[f1]],[[f],[e],[a1]],
[[am],[e],[e1]],[[b],[e],[ds1]],[[em],[e],[e1]],
[[em],[e],[r]],[[b],[e],[b]],[[em],[q],[e1]],
[[em],[e],[g1]],[[gm],[e],[d1]],[[a],[e],[cs1]],[[dm],[e],[d1]],[[dm],[e],[r]],
[[a],[e],[a1]],[[dm],[q],[f1]],[[a],[e],[a]],
[[a],[e],[cs1]],[[dm],[q],[d1]],[[dm],[e],[r]],
[[a7],[e],[a]],[[dm],[e],[d1]],
[[dm],[e],[f1]],[[d7],[e],[d1]],
[[d7],[e],[c1]],[[g],[e],[b]],[[c],[e],[c1]],[[c],[e],[r]],[[c],[e],[g1]],
[[c],[q],[e1]],[[g],[q],[g]],
[[c],[q],[c1]],[[c],[e],[r]],[[e7],[e],[e1]],[[am],[dq],[a1]],
[[bdim],[e],[b1]],[[e],[q],[gs1]],[[e],[e],[r]],
[[a],[e],[a1]],[[dm],[e],[f1]],[[dm],[e],[d1]],[[e],[e],[b]],
[[e],[e],[e1]],[[am],[e],[c1]],[[am],[e],[a]],[[am],[e],[r]],[[e],[e],[b]],
[[am],[dq],[c1]],[[a],[e],[e1]],[[dm],[dq],[f1]],
[[e7],[e],[gs1]],[[am],[e],[a1]],
[[bdim],[q],[b1]],[[e],[e],[e1]],[[am],[e],[c1]],
[[dm],[e],[b]],[[e],[de],[gs]],[[e],[s],[a]],[[am],[q],[a]],
[[am],[e],[r]],[[e],[e],[b]],
[[am],[dq],[c1]],[[a],[e],[e1]],[[dm],[dq],[f1]],
[[e7],[e],[gs1]],[[am],[e],[a1]],
[[bdim],[q],[b1]],[[e],[e],[e1]],[[am],[e],[c1]],
[[dm],[e],[b]],[[e],[de],[gs]],[[e],[s],[a]],[[am],[q],[a]],
[[em],[q],[e1]],[[f],[tq],[e1]],[[dm],[q],[d1]],[[e],[h],[e1]]
].
```

?- v(Variation).

```

Variation = [[am,am],[e,s,s,s,s,e],[el,al,bl,al,gs1,al]],
[[bdim,e],[e,s,s,e],[bl,gs1,fl,gs1]],[[e],[e],[r]],
[[am,dm,dm],[s,s,s,s,s,s],[gl,al,gl,fl,el,d1]],
[[e,e],[th,th,th,th,e],[b,d1,cl,b,el]],
[[am,am],[s,s,e],[cl,g,a]],[[am],[e],[r]],
[[e,f],[s,s,q],[el,trill,fl]],
[[f,am],[st,st,st,st,st,tht,tht],[gl,bl,al,gl,al,
fl,el]],
[[b,em],[s,s,e],[ds1,trill,el]],[[em],[e],[r]],
[[b,em],[s,s,t,t,t],[b,ds1,cl,b,el]],
[[em,gm],[st,st,st,st,st,tht,tht],[fl,as1,gl,fl,
gl,el,d1]],
[[a,dm],[s,s,e],[cs1,trill,d1]],[[dm],[e],[r]],
[[a,dm],[s,s,q],[al,el,fl]],[[a,a],[s,s,s,s],
[a,d1,e,cs1]],
[[dm],[q],[d1]],[[dm],[e],[r]],
[[a7,dm],[s,s,s,s],[a,al,fl,d1]],
[[dm,d7],[th,th,th,th,s,s],[fs1,fs1,el,d1,cl,d1]],
[[d7],[e],[cl]],[[g,c],[s,s,e],[b,trill,cl]],
[[c],[e],[r]],[[c,c],[s,s,s,s,s,s],[gl,gl,fl,el,
d1,el]],
[[g,c],[t,t,t,q],[g,gl,el,cl]],[[c],[e],[r]],
[[e7,am],[e,s,s,s,s,e],[el,al,bl,al,gs1,al]],
[[bdim,e],[e,s,s,e],[bl,gs1,fl,gs1]],[[e],[e],[r]],
[[a,dm,dm],[s,s,s,s,s,s],[gl,al,gl,fl,el,d1]],
[[e,e],[th,th,th,th,s,s],[b,el,fl,el,ds1,el]],
[[am,am],[s,s,s,s],[cl,a,g,a]],[[am],[e],[r]],
[[e,am],[e,e,q],[b,trill,cl]],
[[a,dm,e7,am,bdim],[s,s,e,s,s,e,e,e,q],
[el,b,cs1,d1,el,fl,gs1,al,bl]],
[[e,am],[s,s,s,s],[fl,el,d1,cl]],
[[dm,e],[s,s,s,e],[b,gs,f,gs]],[[e],[s],[a]],
[[am],[q],[a]],[[am],[e],[r]],
[[e,am],[e,e,q],[b,trill,cl]],
[[a,dm,e7,am,bdim],[s,s,e,s,s,e,e,e,q],
[el,b,cs1,d1,el,fl,gs1,al,bl]],
[[e,am],[s,s,e],[el,b,cl]],[[dm,e],[s,s,s,e],
[cl,b,a,gs]],
[[e],[s],[a]],[[am,em],[e,s,th,th,q],[a,b,cl,d1,
el]],
[[f],[tq],[el]],[[dm],[q],[d1]],[[e],[h],[el]]]

```

?-

?- v(Variation).

```
Variation = [[am,am],[s,s,s,s,q],[el,g1,f1,el,al]],
[[bdim,e],[s,s,q],[b1,f1,gs1]],[[e],[e],[r]],
[[am,dm,dm],[s,ts,s,s,th,th,th,th],[al,al,b1,g1,
f1,g1,el,d1]],
[[e,e],[th,th,th,th,e],[b,d1,cl,b,el]],
[[am,am],[s,s,e],[cl,g,a]],[[am],[e],[r]],
[[e,f],[s,s,q],[el,trill,f1]],
[[f,am],[s,s,s,s],[al,el,ds1,el]],
[[b,em],[s,s,e],[ds1,trill,el]],[[em],[e],[r]],
[[b,em],[e,s,s,e],[b,b1,g1,el]],
[[em,gm],[s,s,s,s],[g1,d1,cs1,d1]],
[[a,dm],[s,s,e],[cs1,trill,d1]],[[dm],[e],[r]],
[[a,dm],[e,s,s,e],[b1,al,g1,f1]],
[[a,a],[s,s,s,s],[a,d1,e,cs1]],[[dm],[q],[d1]],
[[dm],[e],[r]],[[a7,dm],[s,s,s,s],[a,al,f1,d1]],
[[dm,d7],[s,s,s,s],[g1,fs1,el,d1]],[[d7],[e],[cl]],
[[g,c],[s,s,e],[b,trill,cl]],[[c],[e],[r]],
[[c,c],[s,s,s,s,s,s],[g1,g1,f1,el,d1,el]],
[[g,c],[s,s,s,s,e,e],[g,cl,d1,cl,b,cl]],
[[c],[e],[r]],[[e7,am],[e,s,s,s,s,e],
[el,al,b1,al,gs1,al]],
[[bdim,e],[s,s,s,s,s,s],[b1,b1,al,gs1,f1,gs1]],
[[e],[e],[r]],[[a,dm,dm],[s,ts,s,s,th,th,th,th],
[al,al,b1,g1,f1,g1,el,d1]],
[[e,e],[s,s,s,s],[b,b1,gs1,el]],
[[am,am],[s,s,s,s],[cl,a,g,a]],[[am],[e],[r]],
[[e,am],[e,e,q],[b,trill,cl]],
[[a,dm,e7,am,bdim],[s,s,e,s,s,e,e,e,q],
[el,b,e1,f1,gs1,el,gs1,al,b1]],
[[e,am],[s,s,s,s],[el,cl,b,cl]],
[[dm,e],[s,s,s,e],[b,gs,f,gs]],[[e],[s],[a]],
[[am],[q],[a]],[[am],[e],[r]],
[[e,am],[e,e,q],[b,trill,cl]],
[[a,dm,e7,am,bdim],[s,s,e,s,s,e,e,e,q],
[el,b,e1,f1,gs1,el,gs1,al,b1]],
[[e,am],[s,s,s,s],[el,cl,b,cl]],
[[dm,e],[s,s,de],[b,f,gs]],[[e],[s],[a]],
[[am,em],[e,s,th,th,q],[a,b,cl,d1,el]],
[[f],[tq],[el]],[[dm],[q],[d1]],[[e],[h],[el]]]
```

?-

# Telemann - em

Handwritten musical notation for Telemann - em, showing four variations of a melody and its accompaniment.

**Variation 1**: Melody starts with a triplet of eighth notes (F#, G#, A) followed by a half note (B). The accompaniment consists of a bass line with a half note (B) and a whole note (F#).

**Variation 2**: Melody starts with a triplet of eighth notes (F#, G#, A) followed by a half note (B). The accompaniment consists of a bass line with a half note (B) and a whole note (F#).

**Framework**: Melody starts with a half note (B). The accompaniment consists of a bass line with a half note (B) and a whole note (F#).

**Original**: Melody starts with a half note (B). The accompaniment consists of a bass line with a half note (B) and a whole note (F#).

Handwritten musical notation for Telemann - em, showing four variations of a melody and its accompaniment.

**Variation 1**: Melody starts with a half note (B). The accompaniment consists of a bass line with a half note (B) and a whole note (F#).

**Variation 2**: Melody starts with a half note (B). The accompaniment consists of a bass line with a half note (B) and a whole note (F#).

**Variation 3**: Melody starts with a half note (B). The accompaniment consists of a bass line with a half note (B) and a whole note (F#).

**Variation 4**: Melody starts with a half note (B). The accompaniment consists of a bass line with a half note (B) and a whole note (F#).



Handwritten musical notation for a 4-measure piece. The first three staves show a melody with various chords and a trill. The fourth staff shows the chord progression: dm, C, F, C, am, dm, G7, C, f#dim.

Handwritten musical notation for a 4-measure piece. The first three staves show a melody with triplets and various chords. The fourth staff shows the chord progression: G, am, C, F, dm, G, C, R, G, C, G.

Handwritten musical score for the first system, featuring four staves with treble clefs. The first three staves contain melodic lines with various notes, rests, and triplets. The fourth staff contains a bass line with chords labeled C, em7, F, G, C, bdim, edim, bdim7, cdim, dm, and D. The notation includes many accidentals and dynamic markings like p.

Handwritten musical score for the second system, featuring four staves with treble clefs. The notation continues with complex melodic lines, including many triplets and accidentals. The fourth staff includes chord labels: d#dim7, d#dim, em, am, f#dim, and B. The handwriting is dense and expressive.

Handwritten musical notation for the first system, consisting of four staves. The notation includes various chords and melodic lines with accidentals and trills. Chord labels below the bottom staff include *em*, *am*, *B*, *em*, *bdim*, *C*, *C7*, and *c#dim*.

Handwritten musical notation for the second system, consisting of four staves. The notation includes various chords and melodic lines with accidentals and trills. Chord labels below the bottom staff include *dm*, *D*, *D7*, *G*, *C*, *F*, *am*, *g#dim*, and *am*.

Handwritten musical notation for a four-staff system. The first three staves are treble clef, and the fourth is bass clef. The notation includes various chords and melodic lines with accidentals and trills. The fourth staff has chord labels: bdim, E, dm, E, D, bm, dm, E7, and am.

Handwritten musical notation for a four-staff system. The first three staves are treble clef, and the fourth is bass clef. The notation includes various chords and melodic lines with accidentals and trills. The fourth staff has chord labels: E, am, E, F, E, and A.

Handwritten musical notation on four staves. The first three staves are in treble clef with a key signature of one flat (Bb). The first staff contains four measures with trills (tr) and a sharp sign (#). The second staff contains four measures with trills and triplets (3). The third staff contains four measures with trills and a sharp sign. The fourth staff contains four measures with notes and a sharp sign, with handwritten labels 'Bb', 'A', 'E', 'F', 'E', 'am', 'dm', 'bdm', and 'E' above the notes.

Handwritten musical notation on four staves. The first three staves are in treble clef with a key signature of one flat (Bb). The first staff contains four measures with a sharp sign (#) and a triplet (3). The second staff contains four measures with a sharp sign and a triplet. The third staff contains four measures with a sharp sign and a triplet. The fourth staff contains four measures with notes and a sharp sign, with handwritten labels 'an', 'g#dim', 'am', 'bdm', 'am', 'bdm', 'am', 'E', and 'am' above the notes.

```

[
[[r],[q],[e1]],[[r],[h],[c2]],[[e],[q],[gs1]],[[am],[h],[a1]],[[e7],[q],[e1]],
[[am],[q],[c1]],[[em],[q],[b]],[[bdim],[q],[a]],[[e],[dh],[e1]],[[am],[tdh],[e1]],
[[e7],[th],[e1]],[[e],[q],[e2]],[[a],[e],[cs2]],[[a],[e],[g1]],[[a],[dq],[a]],
[[a7],[e],[cs2]],[[dm],[q],[d2]],[[dm],[q],[d1]],[[d],[q],[d2]],
[[g],[e],[b1]],[[g],[e],[f1]],[[g],[dq],[g]],[[g7],[e],[b1]],[[c],[q],[c2]],
[[c],[q],[c1]],[[c],[q],[g1]],[[f],[e],[a1]],[[f],[e],[f1]],[[c],[dq],[g1]],
[[c],[e],[c1]],[[dm],[e],[d1]],[[dm],[e],[b]],[[c],[dq],[c1]],[[c],[e],[g1]],
[[f],[e],[a1]],[[f],[e],[f1]],[[c],[dq],[g1]],[[am],[e],[c2]],[[dm],[q],[f1]],
[[dm],[q],[f1]],[[dm],[q],[f1]],[[g7],[dq],[f1]],[[g7],[s],[e1]],
[[g7],[s],[d1]],[[c],[e],[e1]],[[fsdim],[e],[fs1]],[[g],[e],[g1]],
[[g],[e],[d1]],[[g],[e],[e1]],[[g],[e],[b]],[[am],[e],[c1]],[[c],[e],[g]],
[[f],[e],[a]],[[dm],[e],[f1]],[[g],[dq],[d1]],[[g],[e],[c1]],[[c],[h],[c1]],
[[c],[q],[g1]],[[r],[h],[e2]],[[g],[q],[b1]],[[c],[h],[c2]],[[g],[q],[g1]],
[[c],[q],[e1]],[[em7],[q],[d1]],[[f],[q],[c1]],[[g],[dh],[g1]],
[[c],[tdh],[g1]],[[bdim],[th],[g1]],[[edim],[e],[g1]],[[edim],[e],[a1]],
[[csdim],[e],[bf1]],[[csdim],[e],[g1]],[[csdim],[e],[cs1]],[[csdim],[e],[g1]],
[[csdim],[e],[bf1]],[[csdim],[s],[a1]],[[csdim],[s],[g1]],[[dm],[e],[f1]],
[[dm],[e],[e1]],[[dm],[q],[d1]],[[d],[e],[a1]],[[d],[e],[b1]],
[[dsdim],[e],[c2]],[[dsdim],[e],[a1]],[[dsdim],[e],[ds1]],[[dsdim],[e],[a1]],
[[dsdim],[e],[c2]],[[dsdim],[s],[b1]],[[dsdim],[s],[a1]],[[em],[e],[g1]],
[[em],[e],[fs1]],[[em],[e],[e1]],[[em],[e],[g1]],[[em],[q],[b1]],
[[am],[tq],[b1]],[[am],[q],[a1]],[[fsdim],[q],[fs1]],[[b],[dq],[ds1]],
[[b],[e],[c2]],[[b7],[e],[b1]],[[b7],[e],[ds2]],[[em],[e],[e2]],
[[am],[e],[a1]],[[b],[dq],[fs1]],[[b],[e],[e1]],[[em],[q],[e1]],
[[em],[e],[r]],[[em],[e],[b1]],[[bdim],[de],[b1]],[[bdim],[th],[a1]],
[[bdim],[th],[b1]],[[c],[e],[c2]],[[c],[e],[g1]],[[c],[q],[g1]],
[[c7],[de],[a1]],[[c7],[s],[bf1]],[[f],[q],[a1]],[[f],[e],[r]],
[[csdim],[e],[cs2]],[[csdim],[de],[cs2]],[[csdim],[th],[b1]],
[[csdim],[th],[cs2]],[[dm],[e],[d2]],[[dm],[e],[a1]],[[d],[q],[a1]],
[[d7],[de],[b1]],[[d7],[s],[c2]],[[g],[q],[b1]],[[g],[e],[r]],
[[g],[e],[ds2]],[[c],[e],[e2]],[[c],[e],[c2]],[[f],[dq],[a1]],
[[f],[e],[gs1]],[[f],[q],[a1]],[[am],[tq],[a1]],[[gsdim],[q],[b1]],
[[am],[q],[c2]],[[bdim],[q],[b1]],[[e],[e],[a1]],[[e],[e],[gs1]],
[[dm],[q],[a1]],[[e],[dq],[gs1]],[[e],[e],[a1]],[[e],[q],[b1]],
[[d],[q],[d1]],[[bm],[q],[d1]],[[dm],[q],[d1]],[[e7],[dq],[d2]],
[[e7],[e],[b1]],[[am],[q],[c2]],[[am],[te],[c2]],[[am],[e],[d2]],
[[e],[dq],[b1]],[[e],[e],[a1]],[[am],[h],[a1]],[[e],[q],[e1]],
[[f],[e],[f1]],[[f],[e],[ds1]],[[e],[dq],[e1]],[[a],[e],[a]],
[[bf],[e],[bf]],[[bf],[e],[gs]],[[a],[q],[a]],[[a],[e],[r]],
[[e],[e],[e1]],[[f],[e],[f1]],[[f],[e],[ds1]],[[e],[dq],[e1]],
[[e],[e],[b1]],[[am],[q],[c2]],[[dm],[q],[a1]],[[dm],[q],[f1]],
[[bdim],[q],[d1]],[[e],[q],[b]],[[e],[q],[gs]],[[am],[dq],[e1]],
[[gsdim],[e],[gs1]],[[am],[e],[a1]],[[bdim],[e],[f1]],[[am],[e],[e1]],
[[bdim],[e],[d1]],[[am],[q],[c1]],[[e],[de],[b]],[[e],[s],[a]],
[[am],[dh],[a]]
].

```

```

[
[[r],[q],[e1]],[[r],[h],[c2]],[[e],[q],[gs1]],[[am],[h],[a1]],[[e7],[q],[e1]],
[[am],[q],[c1]],[[em],[q],[b1]],[[bdim],[q],[a]],[[e],[dh],[e1]],[[am],[tdh],[e1]],
[[e7],[th],[e1]],[[e],[q],[e2]],[[a],[e],[cs2]],[[a],[e],[g1]],[[a],[dq],[a]],
[[a7],[e],[cs2]],[[dm],[q],[d2]],[[dm],[q],[d1]],[[d],[q],[d2]],
[[g],[e],[b1]],[[g],[e],[f1]],[[g],[dq],[g]],[[g7],[e],[b1]],[[c],[q],[c2]],
[[c],[q],[c1]],[[c],[q],[g1]],[[f],[e],[a1]],[[f],[e],[f1]],[[c],[dq],[g1]],
[[c],[e],[c1]],[[dm],[e],[d1]],[[dm],[e],[b]],[[c],[dq],[c1]],[[c],[e],[g1]],
[[f],[e],[a1]],[[f],[e],[f1]],[[c],[dq],[g1]],[[am],[e],[c2]],[[dm],[q],[f1]],
[[dm],[q],[f1]],[[dm],[q],[f1]],[[g7],[dq],[f1]],[[g7],[s],[e1]],
[[g7],[s],[d1]],[[c],[e],[e1]],[[fsdim],[e],[fs1]],[[g],[q],[g1]],
[[g],[q],[e1]],[[am],[q],[c1]],
[[f],[e],[a]],[[dm],[e],[f1]],[[g],[dq],[d1]],[[g],[e],[c1]],[[c],[h],[c1]],
[[c],[q],[g1]],[[r],[h],[e2]],[[g],[q],[b1]],[[c],[h],[c2]],[[g],[q],[g1]],
[[c],[q],[e1]],[[em7],[q],[d1]],[[f],[q],[c1]],[[g],[dh],[g1]],
[[c],[tdh],[g1]],[[bdim],[th],[g1]],[[edim],[q],[g1]],
[[csdim],[q],[bf1]],[[csdim],[q],[cs1]],
[[csdim],[q],[bf1]],[[dm],[q],[f1]],
[[dm],[q],[d1]],[[d],[q],[a1]],
[[dsdim],[q],[c2]],[[dsdim],[q],[ds1]],
[[dsdim],[q],[c2]],[[em],[q],[g1]],
[[em],[q],[e1]],[[em],[q],[b1]],
[[am],[tq],[b1]],[[am],[q],[a1]],[[fsdim],[q],[fs1]],[[b],[h],[ds1]],
[[b7],[q],[b1]],[[em],[e],[e2]],
[[am],[e],[a1]],[[b],[dq],[fs1]],[[b],[e],[e1]],[[em],[q],[e1]],
[[em],[e],[r]],[[em],[e],[b1]],[[bdim],[de],[b1]],[[bdim],[th],[a1]],
[[bdim],[th],[b1]],[[c],[e],[c2]],[[c],[e],[g1]],[[c],[q],[g1]],
[[c7],[de],[a1]],[[c7],[s],[bf1]],[[f],[q],[a1]],[[f],[e],[r]],
[[csdim],[e],[cs2]],[[csdim],[de],[cs2]],[[csdim],[th],[b1]],
[[csdim],[th],[cs2]],[[dm],[e],[d2]],[[dm],[e],[a1]],[[d],[q],[a1]],
[[d7],[de],[b1]],[[d7],[s],[c2]],[[g],[q],[b1]],[[g],[e],[r]],
[[g],[e],[ds2]],[[c],[e],[e2]],[[c],[e],[c2]],[[f],[dq],[a1]],
[[f],[e],[gs1]],[[f],[q],[a1]],[[am],[tq],[a1]],[[gsdim],[q],[b1]],
[[am],[q],[c2]],[[bdim],[q],[b1]],[[e],[e],[a1]],[[e],[e],[gs1]],
[[dm],[q],[a1]],[[e],[dq],[gs1]],[[e],[e],[a1]],[[e],[q],[b1]],
[[d],[q],[d1]],[[bm],[q],[d1]],[[dm],[q],[d1]],[[e7],[dq],[d2]],
[[e7],[e],[b1]],[[am],[q],[c2]],[[am],[te],[c2]],[[am],[e],[d2]],
[[e],[dq],[b1]],[[e],[e],[a1]],[[am],[h],[a1]],[[e],[q],[e1]],
[[f],[e],[f1]],[[f],[e],[ds1]],[[e],[dq],[e1]],[[a],[e],[a]],
[[bf],[e],[bf]],[[bf],[e],[gs]],[[a],[q],[a]],[[a],[e],[r]],
[[e],[e],[e1]],[[f],[e],[f1]],[[f],[e],[ds1]],[[e],[dq],[e1]],
[[e],[e],[b1]],[[am],[q],[c2]],[[dm],[q],[a1]],[[dm],[q],[f1]],
[[bdim],[q],[d1]],[[e],[q],[b]],[[e],[q],[gs]],[[am],[dq],[e1]],
[[gsdim],[e],[gs1]],[[am],[e],[a1]],[[bdim],[e],[f1]],[[am],[e],[e1]],
[[bdim],[e],[d1]],[[am],[q],[c1]],[[e],[de],[b]],[[e],[s],[a]],
[[am],[dh],[a]]
].

```

?- v(Variation).

```
Variation = [[r,r],[t,t,t,h],[gs1,fs1,e1,c2]],
[[e,am],[q,q,q],[gs1,trill,a1]],
[[e7,am],[e,e,q],[e1,b,c1]],[[em],[q],[b]],
[[bdim,e],[q,q,q,q],[a,c1,f1,e1]],[[am],[tdh],
[e1]],
[[e7,e],[tq,e,e,q],[fs1,e1,d2,e2]],
[[a,a],[s,s,s,s],[cs2,g1,fs1,g1]],[[a],[dq],[a]],
[[a7,dm],[s,s,q],[cs2,trill,d2]],
[[dm,d],[t,t,t,q],[e1,d1,c2,d2]],
[[g,g],[s,s,s,s],[b1,f1,e1,f1]],[[g],[dq],[g]],
[[g7,c],[s,s,q],[b1,trill,c2]],[[c],[q],[c1]],
[[c],[q],[g1]],[[f,f],[s,s,e],[a1,e1,f1]],
[[c],[dq],[g1]],[[c],[e],[c1]],
[[dm,dm],[s,s,e],[d1,a,b]],[[c],[dq],[c1]],
[[c],[e],[g1]],[[f,f],[s,s,e],[a1,e1,f1]],
[[c],[dq],[g1]],
[[am,dm],[th,th,th,th,s,s,s,s],[c2,a1,g1,a1,f1,a1,
g1,f1]],
[[dm],[q],[f1]],[[dm],[q],[f1]],[[g7],[dq],[f1]],
[[g7],[s],[e1]],[[g7],[s],[d1]],[[c],[e],[e1]],
[[fsdim,g],[s,s,q],[fs1,trill,g1]],
[[g,am,f],[er,e,tst,st,st,st,st,e],
[e1,e1,f1,d1,c1,d1,b,a]],[[dm,g],[e,s,s,q],
[f1,d1,c1,d1]],
[[g],[e],[c1]],[[c,c],[e,e,q,q],[c1,e1,a1,g1]],
[[r,g],[e,e,q,q],[e2,d2,e2,b1]],
[[c,g],[e,e,q,q],[c2,b1,c2,g1]],[[c],[q],[e1]],
[[em7],[q],[d1]],[[f,g],[s,s,s,s,dh],[c1,d1,e1,f1,
g1]],
[[c],[tdh],[g1]],[[bdim],[th],[g1]],
[[edim],[q],[g1]],[[csdim],[q],[bf1]],
[[csdim],[q],[cs1]],[[csdim],[q],[bf1]],
[[dm,dm],[t,t,t,q],[f1,d1,c1,d1]],
[[d,dsdim],[e,e,q],[a1,b1,c2]],
[[dsdim,dsdim],[t,t,t,q],[ds1,fs1,a1,c2]],
[[em,em],[s,s,s,s,e,e],[g1,g1,f1,e1,d1,e1]],
[[em],[q],[b1]],[[am],[tq],[b1]],
[[am,fsdim,b],[er,e,tt,t,t,t,t,t,q],
[a1,a1,b1,g1,fs1,g1,e1,ds1]],
[[b7,em],[s,s,s,s,s,s],[b1,e2,fs2,e2,ds2,e2]],
[[am,b],[e,s,s,s,s,e],[a1,a1,g1,fs1,e1,fs1]],
[[b],[e],[e1]],[[em],[q],[e1]],[[em],[e],[r]],
[[em],[e],[b1]],[[bdim],[de],[b1]],
[[bdim],[th],[a1]],[[bdim,c],[th,e],[b1,trill,c2]],
[[c],[e],[g1]],[[c],[q],[g1]],[[c7],[de],[a1]],
[[c7],[s],[bf1]],[[f],[q],[a1]],[[f],[e],[r]],
[[csdim],[e],[cs2]],[[csdim],[de],[cs2]],
[[csdim],[th],[b1]],[[csdim,dm],[th,e],
[cs2,trill,d2]],
[[dm],[e],[a1]],[[d],[q],[a1]],
[[d7,d7],[de,s],[b1,trill,c2]],[[g],[q],[b1]],
[[g],[e],[r]],[[g,c],[s,s,e],[d2,trill,e2]],
[[c,f],[e,s,s,q],[d2,c2,b1,a1]],
[[f,f],[s,s,q],[gs1,trill,a1]],[[am],[tq],[a1]],
[[gsdim,am],[e,e,q],[b1,trill,c2]],
[[bdim],[q],[b1]],[[e],[e],[a1]],
[[e,dm],[s,s,q],[gs1,trill,a1]],
[[e,e],[q,e,e],[gs1,trill,a1]],
[[e,d],[s,s,s,s,s,s,e],[b1,a1,gs1,fs1,e1,fs1,d1]],
[[bm],[q],[d1]],[[dm,e7],[s,e,s,dq],[d1,f1,a1,d2]],
[[e7,am],[s,s,q],[b1,trill,c2]],[[am],[te],[c2]],
[[am,e],[e,s,s,q],[e2,d2,c2,b1]],[[e],[e],[a1]],
[[am,e],[th,th,th,th,dq,s,de],[gs1,b1,a1,gs1,a1,
```



```

fs1,e1]],
[[f,f],[s,s,e],[f1,c1,ds1]],
[[e,a],[th,th,th,th,q,th,th,s],[e1,cs1,b,cs1,a,
cs1,b,a]],
[[bf],[e],[bf]],[[bf,a],[s,s,q],[gs,trill,a]],
[[a],[e],[r]],[[e,f],[s,s,e],[e1,trill,f1]],
[[f,e],[e,e,q],[ds1,trill,e1]],
[[e,am],[s,s,q],[b1,trill,c2]],
[[dm,dm,bdim],[e,e,s,e,s,q],[g1,a1,g1,f1,e1,d1]],
[[e,e],[e,e,q],[b,f,gs]],
[[am,gsdim],[q,e,e],[e1,f1,gs1]],
[[am,bdim],[th,th,th,th,s,s],[a1,a1,g1,f1,e1,f1]],
[[am,bdim,am,e,e],[s,s,s,s,q,de,s],[e1,c1,d1,e1,a,
b,a]],
[[am],[dh],[a]]

```

?-

?- v(Variation).

```
Variation = [[r,r],[t,t,t,h],[gs1,fs1,e1,c2]],
[[e,am],[q,q,q],[gs1,trill1,a1]],
[[e7,am],[s,s,s,s,e,e],[e1,e1,d1,c1,b,c1]],
[[em],[q],[b]],[[bdim,e],[s,s,s,s,dh],[a,b,c1,d1,
e1]],
[[am],[tdh],[e1]],[[e7,e],[tq,e,e,q],[e1,gs1,b1,
e2]],
[[a,a],[st,st,st,st,st,tht,tht],[b1,ds2,cs2,b1,
cs2,a1,g1]],
[[a],[dq],[a]],[[a7,dm],[s,s,q],[cs2,trill1,d2]],
[[dm,d],[t,t,t,q],[d1,fs1,a1,d2]],
[[g,g],[s,s,s,s],[b1,f1,e1,f1]],[[g],[dq],[g]],
[[g7,c],[s,s,q],[b1,trill1,c2]],[[c],[q],[c1]],
[[c],[q],[g1]],[[f,f],[s,s,s,s],[a1,f1,e1,f1]],
[[c],[dq],[g1]],[[c],[e],[c1]],
[[dm,dm],[s,s,s,s],[e1,d1,c1,b]],[[c],[dq],[c1]],
[[c],[e],[g1]],[[f,f],[s,s,s,s],[b1,a1,g1,f1]],
[[c],[dq],[g1]],
[[am,dm],[th,th,th,th,s,s,s,s],[c2,a1,g1,a1,f1,a1,
g1,f1]],
[[dm],[q],[f1]],[[dm],[q],[f1]],[[g7],[dq],[f1]],
[[g7],[s],[e1]],[[g7],[s],[d1]],[[c],[e],[e1]],
[[fsdim,g],[s,s,q],[fs1,trill1,g1]],
[[g,am,f],[e,e,s,e,s,e],[d1,e1,d1,c1,b,a]],
[[dm,g],[e,s,s,q],[f1,d1,c1,d1]],[[g],[e],[c1]],
[[c,c],[t,t,t,q,q],[c1,g,c1,e1,g1]],
[[r,g],[e,e,q,q],[e2,b1,as1,b1]],
[[c,g],[e,e,q,q],[c2,b1,c2,g1]],[[c],[q],[e1]],
[[em7],[q],[d1]],[[f,g],[s,s,s,s,dh],[c1,g,c1,e1,
g1]],
[[c],[tdh],[g1]],[[bdim],[th],[g1]],
[[edim],[q],[g1]],[[csdim],[q],[bf1]],
[[csdim],[q],[cs1]],[[csdim],[q],[bf1]],
[[dm,dm],[t,t,t,q],[g1,f1,e1,d1]],
[[d,dsdim],[t,t,t,q],[a1,ds2,e1,c2]],
[[dsdim,dsdim],[t,t,t,q],[ds1,fs1,a1,c2]],
[[em,em],[t,t,t,q],[a1,g1,f1,e1]],[[em],[q],[b1]],
[[am],[tq],[b1]],
[[am,fsdim,b],[er,e,tt,t,t,t,t,t,q],
[a1,a1,b1,g1,fs1,g1,e1,ds1]],
[[b7,em],[e,tth,th,th,th,e],[b1,ds2,c2,b1,e2]],
[[am,b],[e,e,q],[a1,e1,fs1]],[[b],[e],[e1]],
[[em],[q],[e1]],[[em],[e],[r]],[[em],[e],[b1]],
[[bdim],[de],[b1]],[[bdim],[th],[a1]],
[[bdim,c],[th,e],[b1,trill1,c2]],[[c],[e],[g1]],
[[c],[q],[g1]],[[c7],[de],[a1]],[[c7],[s],[bf1]],
[[f],[q],[a1]],[[f],[e],[r]],[[csdim],[e],[cs2]],
[[csdim],[de],[cs2]],[[csdim],[th],[b1]],
[[csdim,dm],[th,e],[cs2,trill1,d2]],[[dm],[e],[a1]],
[[d],[q],[a1]],[[d7,d7],[de,s],[b1,trill1,c2]],
[[g],[q],[b1]],[[g],[e],[r]],
[[g,c],[s,s,e],[d2,trill1,e2]],
[[c,f],[e,s,s,s,s,e],[c2,c2,b1,a1,g1,a1]],
[[f,f],[s,s,q],[gs1,trill1,a1]],[[am],[tq],[a1]],
[[gsdim,am],[e,e,q],[b1,trill1,c2]],
[[bdim],[q],[b1]],[[e],[e],[a1]],
[[e,dm],[s,s,q],[gs1,trill1,a1]],
[[e,e],[dq,e],[gs1,trill1,a1]],
[[e,d],[s,s,s,s,s,s,e],[b1,a1,gs1,fs1,e1,fs1,d1]],
[[bm],[q],[d1]],[[dm,e7],[s,e,s,dq],[d1,f1,a1,d2]],
[[e7,am],[s,s,q],[b1,trill1,c2]],[[am],[te],[c2]],
[[am,e],[e,e,q],[d2,a1,b1]],[[e],[e],[a1]],
[[am,e],[e,e,q,q],[a1,gs1,a1,e1]],
```

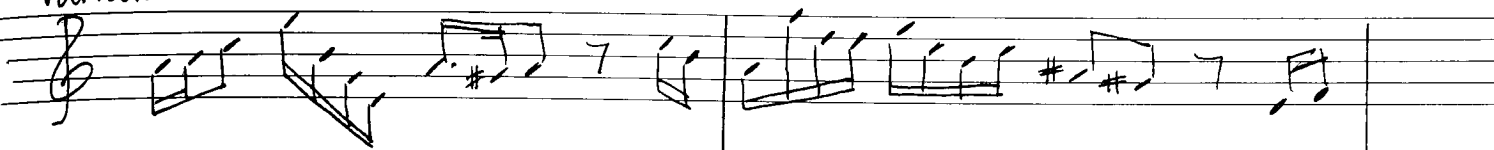
```

[[f,f],[s,s,s,s],[f1,ds1,c1,ds1]],
[[e,a],[th,th,th,th,q,th,th,s],[e1,cs1,b,cs1,a,
                                     cs1,b,a]],
[[bf],[e],[bf]],[[bf,a],[s,s,q],[gs,trill,a]],
[[a],[e],[r]],[[e,f],[s,s,e],[e1,trill,f1]],
[[f,e],[e,e,q],[ds1,trill,e1]],
[[e,am],[s,s,q],[b1,trill,c2]],
[[dm,dm,bdim],[er,e,tst,st,st,st,st,q],
 [a1,a1,b1,g1,f1,g1,e1,d1]],[[e,e],[e,e,q],[b,f,
                                     gs]],
[[am,gsdim],[e,e,e,e],[e1,a1,b,gs1]],
[[am,bdim],[th,th,th,th,s,s],[a1,a1,g1,f1,e1,f1]],
[[am,bdim,am,e,e],[s,s,s,s,q,de,s],[e1,d1,c1,b,a,
                                     b,a]],
[[am],[dh],[a]]

```

?-

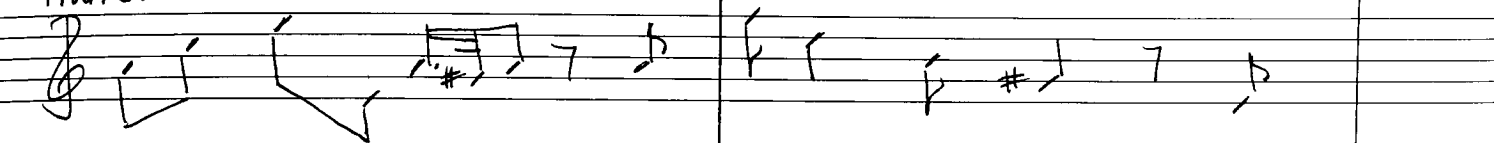
Variation 1



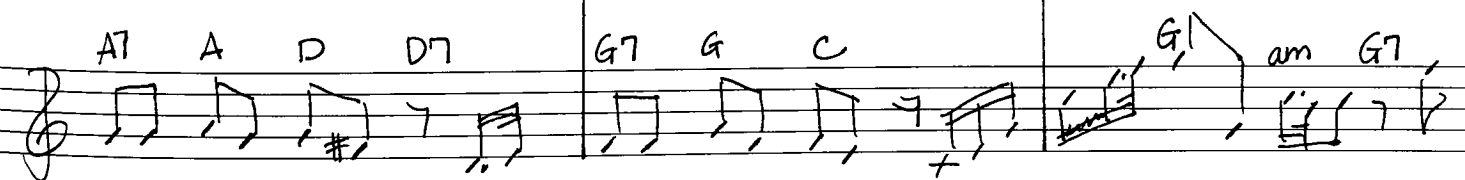
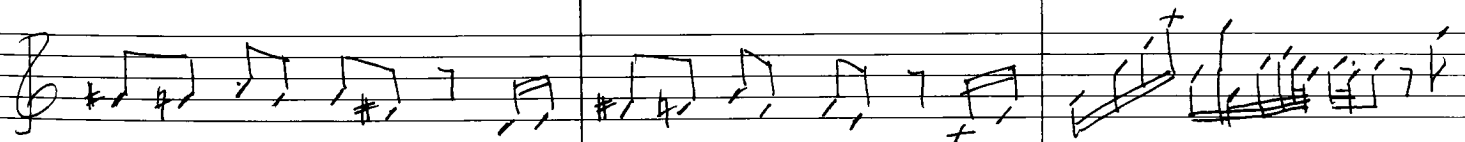
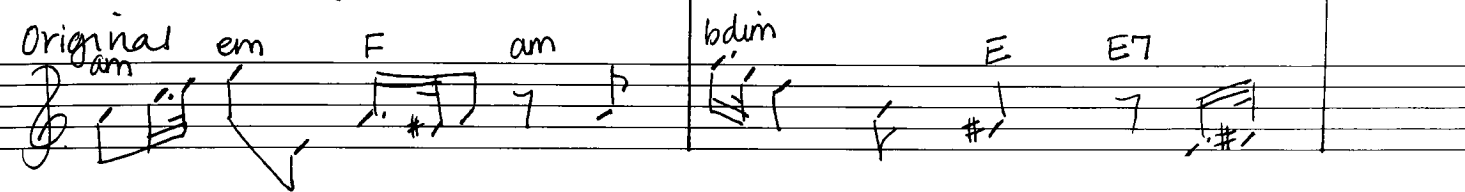
Variation 2



Framework



Original



Handwritten musical score for the first system, featuring four staves with treble clefs. The notation includes various musical symbols such as chords, accidentals, and trills.

Staff 1: Treble clef. First measure has a trill (tr) over a sharp sign. Subsequent measures contain complex chordal structures with accidentals.

Staff 2: Treble clef. Similar to Staff 1, with complex chordal structures and accidentals.

Staff 3: Treble clef. Contains simpler chordal structures with accidentals.

Staff 4: Treble clef. Includes handwritten chord labels: C, G, C, A, dm, edim, A, dm, G, G7, C.

Handwritten musical score for the second system, continuing the musical notation from the first system. It consists of four staves with treble clefs.

Staff 1: Treble clef. Continues the musical notation with various chords and accidentals.

Staff 2: Treble clef. Continues the musical notation with various chords and accidentals.

Staff 3: Treble clef. Continues the musical notation with various chords and accidentals.

Staff 4: Treble clef. Includes handwritten chord labels: F, F7, bdim, E, E7, am, F, E7, am, E.

Handwritten musical notation for a guitar piece, consisting of four staves. The first three staves show a melodic line with trills and a 7th fret barre. The fourth staff includes chord names: A, Bb, d#dim, D, d#dim, em, E, am, bdim, E.

Handwritten musical notation for a guitar piece, consisting of four staves. The first three staves show a melodic line with a triplet and a 7th fret barre. The fourth staff includes chord names: am, F#, dm, E.

```
[
[[am],[e],[a]],[[am],[ds],[cl]],[[am],[th],[dl]],[[em],[e],[el]],
[[em],[e],[el]],[[f],[ds],[a]],[[f],[th],[gs]],[[f],[e],[a]],[[am],[e],[r]],
[[am],[e],[a]],[[bdim],[ds],[fl]],[[bdim],[ts],[el]],[[bdim],[q],[dl]],
[[bdim],[e],[b]],[[e],[q],[gs]],[[e7],[e],[r]],[[e7],[ds],[e]],
[[e7],[th],[fs]],[[a7],[e],[g]],[[a7],[e],[g]],[[a],[e],[a]],[[a],[e],[g]],
[[d],[e],[g]],[[d],[e],[fs]],[[d7],[e],[r]],[[d7],[ds],[d]],[[d7],[th],[e]],
[[g7],[e],[f]],[[g7],[e],[f]],[[g],[e],[g]],[[g],[e],[f]],[[c],[e],[f]],
[[c],[e],[e]],[[c],[s],[r]],[[c],[s],[c]],[[c],[s],[e]],[[c],[s],[g]],
[[c],[e],[cl]],[[c],[ds],[el]],[[c],[th],[fl]],[[g],[e],[gl]],[[g],[e],[g]],
[[am],[ds],[cl]],[[am],[th],[b]],[[am],[e],[cl]],[[g7],[e],[r]],
[[g7],[e],[fl]],[[c],[e],[el]],[[c],[ds],[dl]],[[c],[th],[cl]],[[g],[de],[b]],
[[g],[s],[cl]],[[c],[q],[cl]],[[a],[q],[cs1]],[[dm],[e],[dl]],
[[edim],[ds],[el]],[[edim],[th],[fl]],[[a],[e],[a]],[[a],[e],[cs1]],
[[dm],[e],[dl]],[[dm],[ds],[fl]],[[dm],[th],[el]],[[dm],[e],[dl]],
[[dm],[e],[cl]],[[g],[q],[b]],[[g7],[q],[r]],[[c],[e],[cl]],[[c],[ds],[el]],
[[c],[th],[dl]],[[c],[e],[cl]],[[c],[e],[b]],[[f],[q],[a]],[[f7],[e],[r]],
[[f7],[e],[cl]],[[bdim],[e],[b]],[[bdim],[ds],[dl]],[[bdim],[th],[cl]],
[[bdim],[e],[b]],[[bdim],[e],[a]],[[e],[q],[gs]],[[e7],[q],[r]],
[[am],[e],[a]],[[am],[ds],[cl]],[[am],[th],[dl]],[[am],[e],[el]],
[[am],[e],[e]],[[f],[ds],[a]],[[f],[th],[gs]],[[f],[e],[a]],[[e7],[e],[r]],
[[e7],[e],[dl]],[[am],[e],[cl]],[[am],[ds],[b]],[[am],[th],[a]],
[[e],[de],[gs]],[[e],[s],[a]],[[a],[e],[a]],[[bf],[q],[bf]],[[csdim],[q],[cs1]],
[[d],[q],[dl]],[[dsdim],[e],[ds1]],[[em],[te],[ds1]],[[em],[e],[el]],
[[e],[e],[r]],[[e],[e],[e]],[[am],[e],[cl]],[[bdim],[ds],[b]],
[[bdim],[th],[a]],[[e],[de],[gs]],[[e],[s],[a]],[[am],[q],[a]],
[[f],[h],[el]],[[dm],[q],[dl]],[[e],[h],[el]],[[e],[h],[r]]
].
```

```
[
[[am],[e],[a]],[[am],[e],[cl]],[[em],[e],[el]],
[[em],[e],[el]],[[f],[ds],[a]],[[f],[th],[gs]],[[f],[e],[a]],[[am],[e],[r]],
[[am],[e],[a]],[[bdim],[e],[fl]],[[bdim],[q],[dl]],
[[bdim],[e],[b]],[[e],[q],[gs]],[[e7],[e],[r]],[[e7],[e],[e]],
[[a7],[e],[g]],[[a7],[e],[g]],[[a],[e],[a]],[[a],[e],[g]],
[[d],[e],[g]],[[d],[e],[fs]],[[d7],[e],[r]],[[d7],[e],[d]],
[[g7],[e],[f]],[[g7],[e],[f]],[[g],[e],[g]],[[g],[e],[f]],[[c],[e],[f]],
[[c],[e],[e]],[[c],[e],[r]],[[c],[e],[c]],
[[c],[e],[cl]],[[c],[e],[el]],[[g],[e],[gl]],[[g],[e],[g]],
[[am],[ds],[cl]],[[am],[th],[b]],[[am],[e],[cl]],[[g7],[e],[r]],
[[g7],[e],[fl]],[[c],[e],[el]],[[c],[e],[dl]],[[g],[de],[b]],
[[g],[s],[cl]],[[c],[q],[cl]],[[a],[q],[cs]],[[dm],[e],[dl]],
[[edim],[e],[fl]],[[a],[e],[a]],[[a],[e],[cs]],
[[dm],[e],[dl]],[[dm],[e],[fl]],[[dm],[e],[dl]],
[[dm],[e],[cl]],[[g],[q],[b]],[[g7],[q],[r]],[[c],[e],[cl]],[[c],[e],[el]],
[[c],[e],[cl]],[[c],[e],[b]],[[f],[q],[a]],[[f7],[e],[r]],
[[f7],[e],[cl]],[[bdim],[e],[b]],[[bdim],[e],[dl]],
[[bdim],[e],[b]],[[bdim],[e],[a]],[[e],[q],[gs]],[[e7],[q],[r]],
[[am],[e],[a]],[[am],[e],[cl]],[[am],[e],[el]],
[[am],[e],[e]],[[f],[ds],[a]],[[f],[th],[gs]],[[f],[e],[a]],[[e7],[e],[r]],
[[e7],[e],[dl]],[[am],[e],[cl]],[[am],[e],[b]],
[[e],[de],[gs]],[[e],[s],[a]],[[a],[e],[a]],[[bf],[q],[bf]],[[csdim],[q],[cs]],
[[d],[q],[dl]],[[dsdim],[e],[dsl]],[[em],[te],[dsl]],[[em],[e],[el]],
[[e],[e],[r]],[[e],[e],[e]],[[am],[e],[cl]],[[bdim],[e],[b]],
[[e],[de],[gs]],[[e],[s],[a]],[[am],[q],[a]],
[[f],[h],[el]],[[dm],[q],[dl]],[[e],[h],[el]],[[e],[h],[r]]
].
```



?- v(Variation).

```

Variation = [[am,am],[s,s,e],[a,b,cl]],[[em,em],[s,s,s,s],
                                             [el,b,g,e]],
[[f],[ds],[a]],[[f,f],[th,e],[gs,trill,a]],
[[am],[e],[r]],[[am,bdim],[s,s,s,s],[cl,b,a,fl]],
[[bdim,bdim,e],[s,s,s,s,s,s,e,e],[dl,dl,el,cl,b,
                                             cl,as,gs]],
[[e7],[e],[r]],[[e7,a7],[s,s,e],[e,f,gs]],
[[a7],[e],[g]],[[a],[e],[a]],[[a],[e],[g]],
[[d],[e],[g]],[[d],[e],[fs]],[[d7],[e],[r]],
[[d7,g7],[s,s,s,s],[d,g,g,fs]],[[g7],[e],[f]],
[[g],[e],[g]],[[g],[e],[f]],[[c],[e],[f]],
[[c],[e],[e]],[[c],[e],[r]],[[c,cl],[s,s,s,s],
                                             [c,e,g,cl]],
[[c,g],[s,s,s,s],[el,a1,a1,g1]],
[[g,am],[s,th,th,ds],[g,g1,el,cl]],
[[am,am],[th,e],[b,trill,cl]],[[g7],[e],[r]],
[[g7],[e],[f1]],[[c],[e],[el]],[[c,g],[s,s,de],
                                             [dl,a,b]],
[[g],[s],[cl]],[[c,a],[e,e,q],[cs1,trill,cl]],
[[dm,edim],[s,s,s,s],[dl,g1,a,fl]],
[[a,a],[s,s,s,s],[a,dl,e,cs1]],
[[dm,dm],[s,s,s,s],[dl,g1,g1,fl]],[[dm],[e],[dl]],
[[dm],[e],[cl]],[[g],[q],[b]],[[g7],[q],[r]],
[[c,c],[s,s,s,s],[cl,fl,fl,el]],[[c],[e],[cl]],
[[c],[e],[b]],[[f],[q],[a]],[[f7],[e],[r]],
[[f7],[e],[cl]],[[bdim,bdim],[s,s,s,s],[b,el,f,dl]],
[[bdim],[e],[b]],[[bdim],[e],[a]],[[e],[q],[gs]],
[[e7],[q],[r]],[[am,am],[s,s,s,s],[a,dl,e,cl]],
[[am,am],[s,s,s,s],[el,b,g,e]],[[f],[ds],[a]],
[[f,f],[th,e],[gs,trill,a]],[[e7],[e],[r]],
[[e7],[e],[dl]],[[am],[e],[cl]],
[[am,e],[s,s,de],[b,f,gs]],[[e],[s],[a]],
[[a],[e],[a]],[[bf],[q],[bf]],
[[csdim,d],[e,e,q],[cs1,trill,dl]],
[[dsdim],[e],[ds1]],[[em,em],[ts,s,e],
                                             [ds1,trill,el]],[[e],[e],[r]],
[[e,am],[s,s,s,s],[gs,fs,e,cl]],
[[bdim,e],[s,th,th,s,s,s],[b,b,a,gs,f,gs]],
[[e],[s],[a]],[[am,f],[s,s,s,s,h],[a,b,cl,dl,el]],
[[dm],[q],[dl]],[[e],[h],[el]],[[e],[h],[r]]]

```

?-

?- v(Variation).

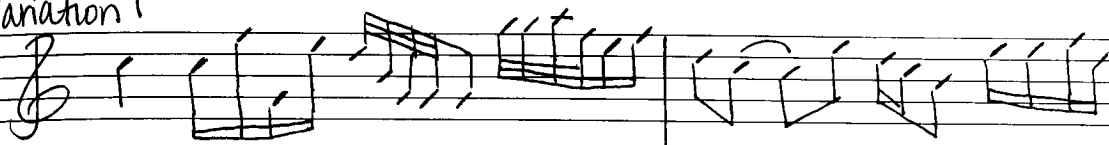
<Undefined Goal>:

| | | | | (12:137) alreadytried(4). :

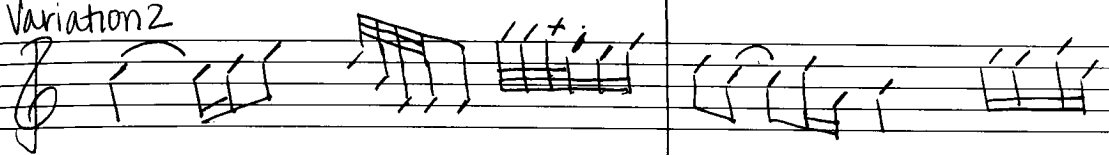
```
Variation = [[am,am],[s,s,s,s],[a,d1,e,c1]],
              [[em,em],[s,s,s,s],[e1,b,g,e]],[[f],[ds],[a]],
              [[f,f],[th,e],[gs,trill,a]],[[am],[e],[r]],
              [[am,bdim],[th,th,th,th,e],[a,c1,b,a,f1]],
              [[bdim,bdim,e],[s,s,s,s,s,s,s,e],[d1,e1,f1,d1,b,
              c1,d1,b,gs]],
              [[e7],[e],[r]],[[e7,a7],[s,s,e],[e,f,gs]],
              [[a7],[e],[g]],[[a],[e],[a]],[[a],[e],[g]],
              [[d],[e],[g]],[[d],[e],[fs]],[[d7],[e],[r]],
              [[d7,g7],[s,s,e],[d,e,fs]],[[g7],[e],[f]],
              [[g],[e],[g]],[[g],[e],[f]],[[c],[e],[f]],
              [[c],[e],[e]],[[c],[e],[r]],[[c,c],[s,s,s,s],
              [c,e,g,c1]],
              [[c,g],[s,s,s,s],[e1,a1,b,g1]],
              [[g,am],[th,th,th,sf,sf,ds],[g,c1,d1,c1,b,c1]],
              [[am,am],[th,e],[b,trill,c1]],[[g7],[e],[r]],
              [[g7],[e],[f1]],[[c],[e],[e1]],
              [[c,g],[s,th,th,s,s,s],[d1,d1,c1,b,a,b]],
              [[g],[s],[c1]],[[c,a],[e,e,q],[cs1,trill,c1]],
              [[dm,edim],[s,s,s,s],[d1,g1,g1,f1]],
              [[a,a],[s,s,s,s],[a,d1,d1,cs1]],
              [[dm,dm],[s,s,s,s],[d1,g1,a,f1]],[[dm],[e],[d1]],
              [[dm],[e],[c1]],[[g],[q],[b]],[[g7],[q],[r]],
              [[c,c],[s,s,s,s],[c1,f1,g,e1]],[[c],[e],[c1]],
              [[c],[e],[b]],[[f],[q],[a]],[[f7],[e],[r]],
              [[f7],[e],[c1]],[[bdim,bdim],[s,s,s,s],[b,e1,e1,
              d1]],
              [[bdim],[e],[b]],[[bdim],[e],[a]],[[e],[q],[gs]],
              [[e7],[q],[r]],[[am,am],[s,s,s,s],[a,d1,d1,c1]],
              [[am,am],[s,s,s,s],[e1,b,g,e]],[[f],[ds],[a]],
              [[f,f],[th,e],[gs,trill,a]],[[e7],[e],[r]],
              [[e7],[e],[d1]],[[am],[e],[c1]],
              [[am,e],[s,th,th,s,s,s],[b,b,a,gs,fs,gs]],
              [[e],[s],[a]],[[a],[e],[a]],[[bf],[q],[bf]],
              [[csdim,d],[e,e,q],[cs1,trill,d1]],
              [[dsdim],[e],[ds1]],[[em,em],[ts,s,e],
              [ds1,trill,e1]],[[e],[e],[r]],
              [[e,am],[th,th,th,th,e],[e,gs,f,e,c1]],
              [[bdim,e],[s,s,de],[b,f,gs]],[[e],[s],[a]],
              [[am,f],[t,t,t,h],[a,c1,f1,e1]],[[dm],[q],[d1]],
              [[e],[h],[e1]],[[e],[h],[r]]]
```

# Telemann - C2

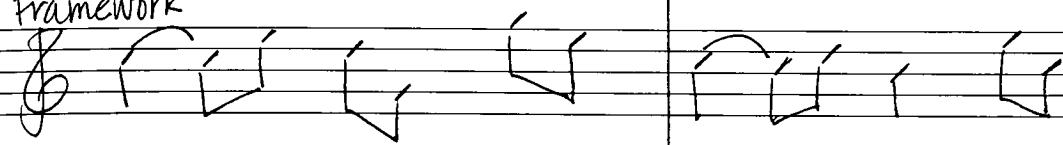
## Variation 1



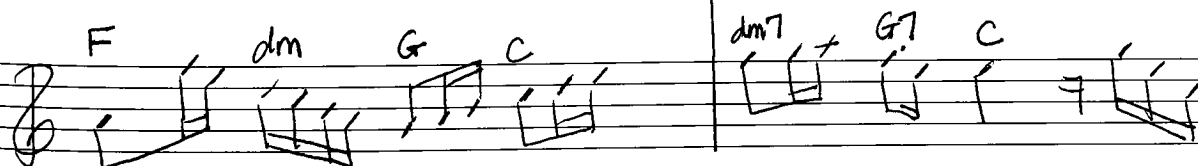
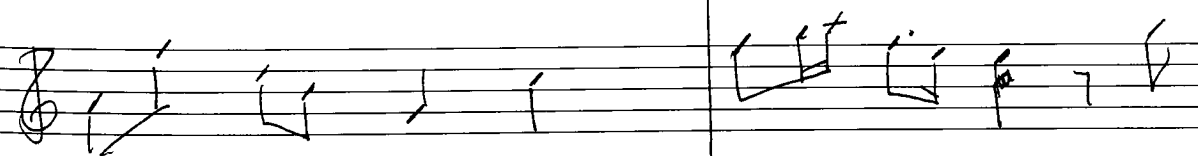
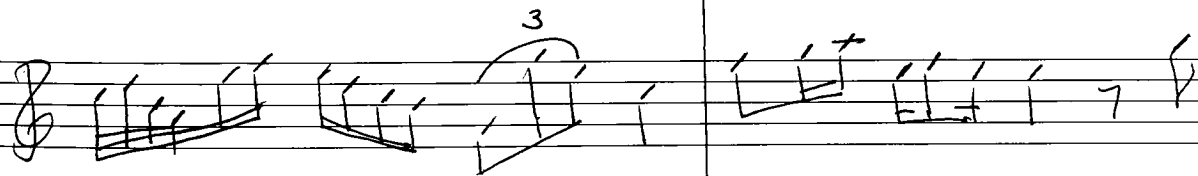
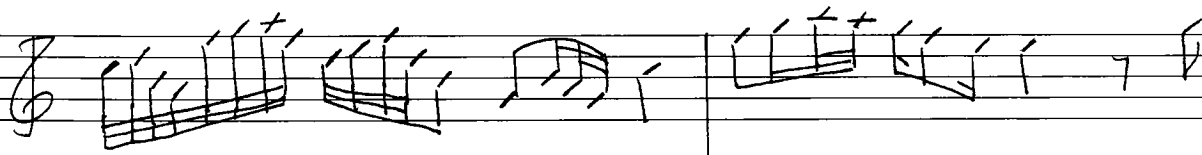
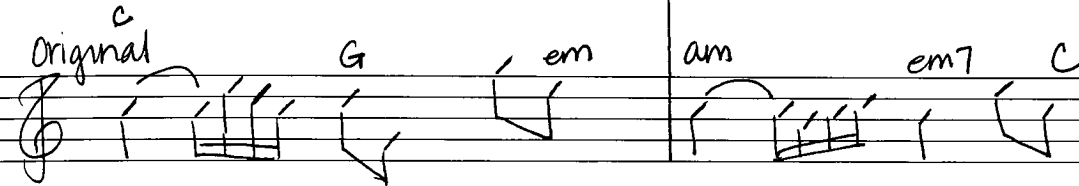
## Variation 2



## Framework



## Original



Handwritten musical notation for the first system, consisting of four staves. The first three staves contain melodic lines with triplets, trills, and slurs. The fourth staff contains a bass line with chords labeled D7, G, am7, dm, G, and D7.

Handwritten musical notation for the second system, consisting of four staves. The first three staves contain melodic lines with trills, triplets, and slurs. The fourth staff contains a bass line with chords labeled em7 C, f#dim, D, G, C, am, D, and G.

Handwritten musical notation for the first system, consisting of four staves. The first three staves contain melodic lines with various chords and accidentals. The fourth staff contains chord symbols: am7 G, D G, gm edmi, A, dm, em7, and A7.

Handwritten musical notation for the second system, consisting of four staves. The first three staves contain melodic lines with various chords and accidentals. The fourth staff contains chord symbols: dm, E, am, and a complex chord labeled "Gdim am" followed by G and am.

Handwritten musical notation for a four-staff system. The first three staves contain melodic lines with various chords and a 7-measure rest. The fourth staff contains a bass line with handwritten chord labels:  $B^b$ ,  $edim$ ,  $C$ ,  $F$ ,  $G$ ,  $C$ ,  $dm7$ , and  $G$ .

Handwritten musical notation for a four-staff system. The first three staves contain melodic lines. The fourth staff contains a bass line with handwritten chord labels:  $G$ ,  $G7$ ,  $em$ ,  $am$ ,  $em7$ , and  $C$ .

Handwritten musical notation for a 4-staff system. The first three staves contain melodic lines with various chords and accidentals. The fourth staff contains chord symbols: F, dm, G, C, dm7+, G7, and am.

Handwritten musical notation for a 4-staff system. The first three staves contain melodic lines with various chords and accidentals. The fourth staff contains chord symbols: bdim, gm, C7, F, G7, and C.

Handwritten musical score for four staves. The first three staves are treble clef, and the fourth is bass clef. The score is divided into two measures. The first measure contains chords:  $dm7$ ,  $G7$ ,  $C$ ,  $dm7$ ,  $G$ ,  $am$ . The second measure contains chords:  $C$ ,  $F$ ,  $C$ ,  $G$ ,  $C$ . There are also some handwritten notes and symbols like "tr" and "3".



```
[
[[c],[q],[cl]],[[c],[ts],[cl]],[[c],[s],[el]],[[c],[s],[dl]],[[c],[s],[cl]],
[[g],[e],[dl]],[[g],[e],[g]],[[g],[e],[gl]],[[em],[e],[el]],[[am],[q],[cl]],
[[am],[ts],[cl]],[[am],[s],[b]],[[am],[s],[cl]],[[am],[s],[dl]],
[[em7],[q],[b]],[[em7],[e],[el]],[[c],[e],[cl]],[[f],[e],[a]],[[f],[s],[fl]],
[[f],[s],[el]],[[dm],[s],[dl]],[[dm],[s],[cl]],[[dm],[s],[b]],[[dm],[s],[a]],
[[g],[e],[g]],[[g],[s],[a]],[[g],[s],[b]],[[c],[e],[cl]],[[c],[s],[dl]],
[[c],[s],[el]],[[dm7],[e],[fl]],[[dm7],[s],[gl]],[[dm7],[s],[a]],
[[g7],[de],[fl]],[[g7],[s],[el]],[[c],[q],[el]],[[c],[s],[r]],[[c],[s],[gl]],
[[c],[s],[el]],[[c],[s],[cl]],[[d7],[e],[dl]],[[d7],[s],[el]],[[s7],[s],[fs1]],
[[g],[e],[gl]],[[g],[e],[b]],[[am7],[q],[c]],[[dm],[s],[r]],[[dm],[s],[a]],
[[dm],[s],[b]],[[dm],[s],[cl]],[[g],[e],[b]],[[g],[s],[dl]],[[g],[s],[cl]],
[[g],[s],[b]],[[g],[s],[a]],[[g],[s],[b]],[[g],[s],[g]],[[d7],[e],[a]],
[[d7],[e],[d]],[[d7],[q],[dl]],[[em7],[e],[r]],[[c],[s],[cl]],[[c],[s],[b]],
[[c],[e],[cl]],[[c],[e],[cl]],[[fsdim],[q],[fs1]],[[d],[e],[r]],
[[g],[s],[gl]],[[g],[s],[dl]],[[c],[e],[el]],[[c],[e],[el]],[[c],[ts],[el]],
[[c],[s],[fs1]],[[am],[s],[gl]],[[am],[s],[a]],[[d],[e],[dl]],
[[d],[e],[dl]],[[g],[ts],[dl]],[[g],[s],[el]],[[g],[s],[fs1]],[[g],[s],[gl]],
[[am7],[e],[cl]],[[g],[s],[dl]],[[g],[s],[el]],[[d],[de],[a]],[[d],[s],[g]],
[[g],[q],[g]],[[gm],[e],[r]],[[edim],[s],[el]],[[edim],[s],[dl]],
[[a],[e],[cs1]],[[a],[s],[dl]],[[a],[s],[el]],[[dm],[e],[a]],[[dm],[e],[a]],
[[em7],[e],[r]],[[em7],[q],[gl]],[[a7],[s],[fl]],[[a7],[s],[el]],
[[dm],[s],[fl]],[[dm],[s],[gl]],[[dm],[s],[fl]],[[dm],[s],[el]],
[[dm],[s],[dl]],[[dm],[s],[cl]],[[dm],[s],[b]],[[dm],[s],[a]],[[e],[s],[gs]],
[[e],[s],[b]],[[e],[e],[el]],[[am],[s],[r]],[[am],[s],[a]],[[am],[s],[b]],
[[am],[s],[cl]],[[gsdim],[s],[d]],[[gsdim],[s],[fl]],[[am],[s],[el]],
[[am],[s],[dl]],[[e],[de],[b]],[[e],[s],[a]],[[am],[q],[a]],[[am],[e],[r]],
[[am],[e],[a]],[[bf],[q],[bf]],[[edim],[ts],[bf]],[[edim],[s],[a]],
[[c],[s],[bf]],[[c],[s],[cl]],[[f],[q],[a]],[[f],[e],[r]],[[f],[e],[cl]],
[[g],[e],[gl]],[[g],[q],[gl]],[[c],[s],[fl]],[[c],[s],[el]],[[dm7],[e],[fl]],
[[dm7],[q],[fl]],[[g],[s],[el]],[[g],[s],[dl]],[[c],[e],[el]],[[c],[e],[cl]],
[[c],[ts],[cl]],[[c],[s],[el]],[[c],[s],[dl]],[[c],[s],[cl]],[[g7],[e],[dl]],
[[g7],[e],[g]],[[g7],[e],[gl]],[[em],[e],[el]],[[am],[q],[cl]],[[am],[ts],[cl]],
[[am],[s],[b]],[[am],[s],[cl]],[[am],[s],[dl]],
[[em7],[q],[b]],[[em7],[e],[el]],[[c],[e],[cl]],[[f],[e],[a]],[[f],[s],[fl]],
[[f],[s],[el]],[[dm],[s],[dl]],[[dm],[s],[cl]],[[dm],[s],[b]],[[dm],[s],[a]],
[[g],[e],[g]],[[g],[s],[a]],[[g],[s],[b]],[[c],[e],[cl]],[[c],[s],[dl]],
[[c],[s],[el]],[[dm7],[e],[fl]],[[dm7],[s],[gl]],[[dm7],[s],[a]],
[[g7],[de],[b]],[[g7],[s],[cl]],[[am],[e],[cl]],[[am],[e],[cl]],[[em],[ts],[cl]],
[[am],[s],[b]],[[am],[s],[cl]],[[am],[s],[dl]],[[bdim],[e],[b]],
[[gm],[e],[bf]],[[gm],[ts],[bf]],[[gm],[s],[a]],[[c7],[s],[bf]],
[[c],[s],[cl]],[[f],[e],[a]],[[f],[s],[a]],[[f],[s],[gl]],[[f],[s],[fl]],
[[f],[s],[el]],[[f],[s],[dl]],[[f],[s],[cl]],[[g7],[q],[g]],[[g7],[q],[r]],
[[g7],[e],[r]],[[c],[s],[cl]],[[c],[s],[b]],[[c],[e],[cl]],[[c],[s],[dl]],
[[c],[s],[el]],[[dm7],[e],[fl]],[[dm7],[q],[fl]],[[g7],[s],[el]],
[[g7],[s],[dl]],[[c],[de],[el]],[[dm7],[s],[fl]],[[g],[de],[dl]],
[[am],[s],[cl]],[[c],[e],[gl]],[[f],[s],[a]],[[f],[s],[gl]],[[c],[e],[g]],
[[g],[e],[b]],[[c],[q],[cl]],[[c],[q],[r]]
].
```

```
[
[[c],[q],[cl]],[[c],[te],[cl]],[[c],[e],[el]],
[[g],[e],[dl]],[[g],[e],[g]],[[g],[e],[gl]],[[em],[e],[el]],[[am],[q],[cl]],
[[am],[te],[cl]],[[am],[e],[dl]],
[[em7],[q],[b]],[[em7],[e],[el]],[[c],[e],[cl]],[[f],[e],[a]],[[f],[e],[fl]],
[[dm],[e],[dl]],[[dm],[e],[b]],
[[g],[q],[g]],[[c],[q],[cl]],[[dm7],[e],[fl]],[[dm7],[s],[gl]],[[dm7],[s],[al]],
[[g7],[de],[fl]],[[g7],[s],[el]],[[c],[q],[el]],[[c],[e],[r]],[[c],[e],[gl]],
[[d7],[q],[dl]],
[[g],[e],[gl]],[[g],[e],[b]],[[am7],[q],[c]],[[dm],[s],[r]],[[dm],[s],[a]],
[[dm],[s],[b]],[[dm],[s],[cl]],[[g],[e],[b]],[[g],[e],[dl]],
[[g],[e],[b]],[[g],[e],[g]],[[d7],[e],[a]],
[[d7],[e],[d]],[[d7],[q],[dl]],[[em7],[e],[r]],[[c],[s],[cl]],[[c],[s],[b]],
[[c],[e],[cl]],[[c],[e],[cl]],[[fsdim],[q],[fs1]],[[d],[e],[r]],
[[g],[e],[gl]],[[c],[e],[el]],[[c],[e],[el]],[[c],[te],[el]],
[[am],[e],[a1]],[[d],[e],[dl]],
[[d],[e],[dl]],[[g],[te],[dl]],[[g],[e],[gl]],
[[am7],[e],[cl]],[[g],[s],[dl]],[[g],[s],[el]],[[d],[de],[a]],[[d],[s],[g]],
[[g],[q],[g]],[[gm],[e],[r]],[[edim],[e],[el]],
[[a],[e],[cs1]],[[a],[e],[el]],[[dm],[e],[a]],[[dm],[e],[a1]],
[[em7],[e],[r]],[[em7],[q],[gl]],[[a7],[e],[el]],
[[dm],[q],[fl]],
[[dm],[e],[dl]],[[dm],[e],[b]],[[e],[s],[gs]],
[[e],[s],[b]],[[e],[e],[el]],[[am],[s],[r]],[[am],[s],[a]],[[am],[s],[b]],
[[am],[s],[cl]],[[gsdim],[s],[d]],[[gsdim],[s],[fl]],[[am],[s],[el]],
[[am],[s],[dl]],[[e],[de],[b]],[[e],[s],[a]],[[am],[q],[a]],[[am],[e],[r]],
[[am],[e],[a]],[[bf],[q],[bf]],[[edim],[ts],[bf]],[[edim],[s],[a]],
[[c],[s],[bf]],[[c],[s],[cl]],[[f],[q],[a]],[[f],[e],[r]],[[f],[e],[cl]],
[[g],[e],[gl]],[[g],[q],[gl]],[[c],[e],[el]],[[dm7],[e],[fl]],
[[dm7],[q],[fl]],[[g],[e],[dl]],[[c],[e],[el]],[[c],[e],[cl]],
[[c],[te],[cl]],[[c],[e],[el]],[[g7],[e],[dl]],
[[g7],[e],[g]],[[g7],[e],[gl]],[[em],[e],[el]],[[am],[q],[cl]],[[am],[te],[cl]],
[[am],[e],[dl]],
[[em7],[q],[b]],[[em7],[e],[el]],[[c],[e],[cl]],[[f],[e],[a]],[[f],[e],[fl]],
[[dm],[e],[dl]],[[dm],[e],[b]],
[[g],[q],[g]],[[c],[q],[cl]],[[dm7],[e],[fl]],[[dm7],[s],[gl]],[[dm7],[s],[al]],
[[g7],[de],[b]],[[g7],[s],[cl]],[[am],[e],[cl]],[[am],[e],[cl]],[[em],[te],[cl]],
[[am],[e],[dl]],[[bdim],[e],[b]],
[[gm],[e],[bf]],[[gm],[te],[bf]],
[[c],[e],[cl]],[[f],[e],[a]],[[f],[e],[a1]],[[f],[e],[fl]],
[[f],[e],[dl]],[[g7],[q],[g]],[[g7],[q],[r]],
[[g7],[e],[r]],[[c],[s],[cl]],[[c],[s],[b]],[[c],[e],[cl]],[[c],[e],[dl]],
[[dm7],[e],[fl]],[[dm7],[q],[fl]],[[g7],[s],[el]],
[[g7],[s],[dl]],[[c],[de],[el]],[[dm7],[s],[fl]],[[g],[de],[dl]],
[[am],[s],[cl]],[[c],[e],[gl]],[[f],[s],[a1]],[[f],[s],[gl]],[[c],[e],[g]],
[[g],[e],[b]],[[c],[q],[cl]],[[c],[q],[r]]
].
```

?- v(Variation).

```
Variation = [[ [c], [q], [cl]], [[c,c], [s,s,s,s], [cl,fl,g,el]],
  [[g,g], [th,th,th,th,e], [dl,b,g,g,g]],
  [[g,em,am], [th,th,th,th,s,s,e,e], [gl,gl,al,fl,el,
    fl,dl,cl]],
  [[am], [te], [cl]], [[am,em7], [e,s,s,e], [el,dl,cl,b]],
  [[em7,c,f], [s,ts,s,s,th,th,th,th], [el,el,fl,dl,cl,
    dl,b,a]],
  [[f,dm,dm], [th,th,th,th,th,th,th,th,e],
    [fl,gl,al,fl,dl,el,fl,dl,b]],
  [[g,c], [e,s,th,th,q], [g,b,a,g,cl]], [[dm7], [e],
    [fl]],
  [[dm7], [s], [gl]], [[dm7,g7], [th,th,s,e], [bl,al,gl,
    fl]],
  [[g7], [s], [el]], [[c], [q], [el]], [[c], [e], [r]],
  [[c,d7], [e,s,s,e], [gl,dl,cl,dl]],
  [[g,g], [st,st,st,st,st,tht,tht], [gl,fl,el,dl,cl,
    dl,b]],
  [[am7], [q], [c]], [[dm], [s], [r]], [[dm], [s], [a]],
  [[dm,dm], [th,th,s], [b,trill,cl]],
  [[g,g], [s,s,s,s], [b,el,f,dl]],
  [[g,g], [th,th,th,th,s,s], [b,b,a,g,f,g]],
  [[d7,d7], [th,th,th,th,e], [a,fs,d,d,d]],
  [[d7], [q], [dl]], [[em7], [e], [r]], [[c], [s], [cl]],
  [[c,c], [s,s,s], [b,trill,cl]],
  [[c,fsdim], [s,s,t,t,t], [cl,el,dl,cl,fs1]],
  [[d], [e], [r]], [[g,c], [th,th,th,th,s,s],
    [gl,gl,fl,el,dl,el]], [[c], [e], [el]],
  [[c,am], [s,s,s,s], [el,e2,c2,al]], [[d], [e], [dl]],
  [[d], [e], [dl]], [[g,g], [s,s,s,s], [dl,d2,bl,gl]],
  [[am7], [e], [cl]], [[g], [s], [dl]], [[g,d], [s,s,e],
    [el,e,a]],
  [[d], [s], [g]], [[g], [q], [g]], [[gm], [e], [r]],
  [[edim,a], [s,s,s,s], [el,cs1,b,cs1]],
  [[a,dm], [s,s,e], [el,e,a]], [[dm], [e], [a]],
  [[em7], [e], [r]], [[em7,a7], [s,s,e,e], [gl,el,dl,el]],
  [[dm,dm,dm], [s,s,s,s,s,s,th,th,s], [fl,gl,al,fl,dl,
    el,fl,dl,b]],
  [[e,e], [th,th,th,th], [gs,cl,cl,b]], [[e], [e], [el]],
  [[am], [s], [r]], [[am], [s], [a]],
  [[am,am], [th,th,s], [b,trill,cl]], [[gsdim], [s], [d]],
  [[gsdim], [s], [fl]], [[am], [s], [el]],
  [[am,e], [s,s,e], [dl,a,b]], [[e], [s], [a]],
  [[am], [q], [a]], [[am], [e], [r]], [[am], [e], [a]],
  [[bf], [q], [bf]], [[edim], [ts], [bf]], [[edim], [s],
    [a]],
  [[c], [s], [bf]], [[c,f], [s,e,e], [cl,g,a]],
  [[f], [e], [r]], [[f,g], [th,th,th,th,e], [cl,dl,el,fl,
    gl]],
  [[g,c], [s,s,s,s,s,s], [gl,gl,fl,el,dl,el]],
  [[dm7], [e], [fl]], [[dm7,g], [s,s,s,s,s,s],
    [fl,fl,el,dl,cl,dl]],
  [[c,c], [s,s,e], [el,b,cl]], [[c,c], [s,s,s,s],
    [cl,fl,fl,el]],
  [[g7,g7], [th,th,th,th,th,th,th,th], [dl,b,a,b,g,b,
    a,g]],
  [[g7,em,am], [s,s,s,s,e,e], [fl,gl,fl,el,dl,cl]],
  [[am], [te], [cl]], [[am,em7], [s,s,q], [dl,a,b]],
  [[em7,c,f], [s,s,s,s,s,s], [dl,el,dl,cl,b,a]],
  [[f,dm,dm], [s,s,s,s,s,s], [el,fl,el,dl,cl,b]],
  [[g,c], [s,s,s,s,e,e], [g,cl,dl,cl,b,cl]],
  [[dm7], [e], [fl]], [[dm7], [s], [gl]],
  [[dm7,g7], [th,th,s,e], [al,gl,al,b]], [[g7], [s],
    [cl]],
```

[[am],[e],[cl]],[[am],[e],[cl]],[[em],[te],[cl]],  
[[am,bdim],[s,s,s,s],[dl,b,a,b]],[[gm],[e],[bf]],  
[[gm],[te],[bf]],[[c,f],[s,s,s,s],[cl,a,g,a]],  
[[f,f,f],[s,s,s,s,s,s],[gl,a1,gl,f1,e1,d1]],  
[[g7],[q],[g]],[[g7],[q],[r]],[[g7],[e],[r]],  
[[c],[s],[cl]],[[c,c],[s,s,s],[b,trill,cl]],  
[[c,dm7],[s,s,e],[dl,e1,f1]],[[dm7],[q],[f1]],  
[[g7],[s],[e1]],[[g7],[s],[dl]],  
[[c,dm7],[de,s],[e1,trill,f1]],[[g],[de],[dl]],  
[[am,c],[th,th,st,st,st],[cl,dl,e1,f1,gl]],  
[[f],[s],[a1]],[[f,c],[s,th,th,s],[gl,f1,gl,g]],  
[[g,c],[s,s,q],[b,trill,cl]],[[c],[q],[r]]]

?-

?- v(Variation).

```
Variation = [[c], [q], [cl]], [[c, c], [ts, s, e], [cl, dl, el]],
[[g, g], [th, th, th, th, e], [dl, b, g, g, g]],
[[g, em, am], [th, th, th, th, s, s, e, e], [gl, gl, al, fl, el,
fl, dl, cl]],
[[am], [te], [cl]], [[am, em7], [s, s, q], [dl, a, b]],
[[em7, c, f], [s, ts, s, s, th, th, th, th], [el, el, fl, dl, cl,
dl, b, a]],
[[f, dm, dm], [s, s, s, s, s, s], [el, fl, el, dl, cl, b]],
[[g, c], [t, t, t, q], [g, gl, el, cl]], [[dm7], [e], [fl]],
[[dm7], [s], [gl]], [[dm7, g7], [s, s, e], [al, el, fl]],
[[g7], [s], [el]], [[c], [q], [el]], [[c], [e], [r]],
[[c, d7], [e, s, s, e], [gl, dl, cl, dl]],
[[g, g], [st, st, st, st, st, tht, tht], [gl, fl, el, dl, cl,
dl, b]],
[[am7], [q], [c]], [[dm], [s], [r]], [[dm], [s], [a]],
[[dm, dm], [th, th, s], [b, trill, cl]],
[[g, g], [s, s, e], [b, cl, dl]], [[g, g], [s, s, s, s], [b, g, f,
g]],
[[d7, d7], [th, th, th, th, e], [a, fs, d, d, d]],
[[d7], [q], [dl]], [[em7], [e], [r]], [[c], [s], [cl]],
[[c, c], [s, s, s], [b, trill, cl]],
[[c, fsdim], [e, s, s, e], [cl, c2, al, fs1]], [[d], [e], [r]],
[[g, c], [s, s, s, s], [gl, el, dl, el]], [[c], [e], [el]],
[[c, am], [tth, th, th, th, s, s], [el, al, bl, al, gl, al]],
[[d], [e], [dl]], [[d], [e], [dl]],
[[g, g], [tth, th, th, th, s, s], [dl, gl, al, gl, fs1, gl]],
[[am7], [e], [cl]], [[g], [s], [dl]],
[[g, d], [sf, sf, sf, sf, s, th, th, s], [el, cl, b, cl, a, cl, b,
a]], [[d], [s], [g]],
[[g], [q], [g]], [[gm], [e], [r]],
[[edim, a], [s, s, s, s], [fl, el, dl, cs1]],
[[a, dm], [th, th, th, th, th, th, th, th], [el, cs1, b, cs1, a,
cs1, b, a]],
[[dm], [e], [al]], [[em7], [e], [r]],
[[em7, a7], [s, s, e, e], [al, gl, fl, el]],
[[dm, dm, dm], [s, e, s, s, s, e], [el, fl, el, dl, cl, b]],
[[e, e], [th, th, th, th], [gs, cl, d, b]], [[e], [e], [el]],
[[am], [s], [r]], [[am], [s], [a]],
[[am, am], [th, th, s], [b, trill, cl]], [[gsdim], [s], [d]],
[[gsdim], [s], [fl]], [[am], [s], [el]],
[[am, e], [sf, sf, sf, sf, s, e], [dl, dl, cl, b, a, b]],
[[e], [s], [a]], [[am], [q], [a]], [[am], [e], [r]],
[[am], [e], [a]], [[bf], [q], [bf]], [[edim], [ts], [bf]],
[[edim], [s], [a]], [[c], [s], [bf]],
[[c, f], [sf, sf, sf, sf, e, e], [cl, cl, b, a, g, a]],
[[f], [e], [r]], [[f, g], [th, th, th, th, e], [cl, dl, el, fl,
gl]],
[[g, c], [s, s, s, s, s, s], [gl, gl, fl, el, dl, el]],
[[dm7], [e], [fl]], [[dm7, g], [s, s, s, s, s, s],
[fl, fl, el, dl, cl, dl]],
[[c, c], [s, s, s, s], [fl, el, dl, cl]],
[[c, c], [s, s, s, s], [cl, fl, g, el]],
[[g7, g7], [s, s, e], [dl, d, g]],
[[g7, em, am], [th, th, th, th, th, th, th, th, q],
[gl, al, bl, gl, el, fl, gl, el, cl]], [[am], [te], [cl]],
[[am, em7], [e, s, s, e], [el, dl, cl, b]],
[[em7, c, f], [th, th, th, th, th, th, th, th, e],
[el, fl, gl, el, cl, dl, el, cl, a]],
[[f, dm, dm], [s, s, s, s, s, s], [el, fl, el, dl, cl, b]],
[[g, c], [s, s, s, s, q], [g, b, a, g, cl]], [[dm7], [e], [fl]],
[[dm7], [s], [gl]], [[dm7, g7], [th, th, s, e], [al, gl, al,
b]],
[[g7], [s], [cl]], [[am], [e], [cl]], [[am], [e], [cl]],
```

```
[[em],[te],[cl]],[[am,bdim],[s,s,s,s],[dl,b,a,b]],  
[[gm],[e],[bf]],[[gm],[te],[bf]],  
[[c,f],[s,s,s,s],[cl,a,g,a]],  
[[f,f,f],[s,s,s,s,s,s],[gl,a1,g1,f1,e1,d1]],  
[[g7],[q],[g]],[[g7],[q],[r]],[[g7],[e],[r]],  
[[c],[s],[cl]],[[c,c],[s,s,s],[b,trill,cl]],  
[[c,dm7],[s,s,s,s],[dl,g1,g1,f1]],[[dm7],[q],[f1]],  
[[g7],[s],[e1]],[[g7],[s],[d1]],  
[[c,dm7],[de,s],[e1,trill,f1]],[[g],[de],[d1]],  
[[am,c],[th,th,st,st,st],[cl,dl,e1,f1,g1]],  
[[f],[s],[a1]],[[f,c],[s,th,th,s],[g1,f1,g1,g]],  
[[g,c],[s,s,q],[b,trill,cl]],[[c],[q],[r]]]
```

?-

## 8. APPENDIX 3: PROGRAM LISTING

Following is a program listing.

```

/*                                KATHY MERCK
                                RIT MASTER'S THESIS PROGRAM
                                EXPERT SYSTEM:
                                MUSICAL VARIATIONS IN THE STYLE OF TELEMANN
*/
/*****
THE PROGRAM TAKES AS INPUT TWO FILES: THE "ORIGINAL" AND THE "FRAMEWORK".
EACH CONSISTS OF A LIST REPRESENTATION OF THE PIECE TO BE VARIED, WITH A SUBLIST
FOR EACH NOTE THAT APPEARS IN THE SCORE; IN OTHER WORDS, THE PIECE AS A WHOLE
IS A LIST OF NOTE LISTS. FOR EACH NOTE, THE NOTE LIST HOLDS THE CHORD
PLAYED WITH THAT NOTE, THE RHYTHM OF THE NOTE, AND THE NOTE'S PITCH. THE ORIGINAL
VERSION CONTAINS EVERY NOTE, WHILE THE FRAMEWORK SHOWS THOSE NOTES THAT PROVIDE
A LESS DETAILED AND MORE BASIC SENSE OF THE MOVEMENT OR UNDERLYING STRUCTURE
OF THE PIECE (MODELLED SOMEWHAT ON THE TRANSFORMATIONS OF SCHENKERIAN ANALYSIS).
A THIRD LIST, "VARIATION", IS BUILT AND MODIFIED BY CLAUSES OF THE PROGRAM TO
PRODUCE A VARIATION ON THE ORIGINAL USING THE FRAMEWORK AS THE MODEL/STARTING
POINT.
*****/

v(Variation) :-
    see("prolog:Telemann_cm"),
    read(Original),
    seen,
    see("prolog:Telemann_cm_framework"),
    read(Framework),
    seen,
    parse(Original,Framework,Variation).

/*****
FIRST, STEP THROUGH THE ORIGINAL LIST AND PRODUCE THE "OrigSeq" LIST WHICH
CONTAINS ALL ORIGINAL NOTES AND IN ADDITION BRACKETS AND LABELS MELODIC SEQUENCES
AS COMPOUND SUBLISTS OF NOTES IN THE SEQUENCE. SECOND, USING THAT LABELLED
VERSION AND THE FRAMEWORK, STEP THROUGH THE TWO IN PARALLEL, ANALYZING NOTE PATTERNS
IN THE FRAMEWORK AND ORIGINAL AND CREATING A NEW LIST "Variation1" WHICH HAS
MELODIC SUBSTITUTIONS MADE FROM THE MELODIC VOCABULARY FACTS IN THE DATABASE,
BRACKETING NOTES OF THESE MELODIC PATTERN GROUPS AS LISTS WITHIN THE PIECE.
THIRD, USING THIS MOST CURRENT LIST REPRESENTATION OF THE VARIATION IN
PROGRESS, FIND APPROPRIATE RHYTHMS FOR THE NEW MELODIC MATERIAL IN THE VARIATION.
FINALLY, CHECK THE NOTES OF THE COMPLETED VARIATION AGAINST THE CHORDS WHICH
ACCOMPANY THEM, CHANGING ACCIDENTALS OF NOTES WHICH "CLASH" WITH THEIR
RESPECTIVE CHORDS.
*****/

parse(Original, Framework, Variation) :-
    labelsequences(Original, OrigSeq),
    checkmelframework(OrigSeq,Framework,'off', Variation1),
    checkrhythframework(Variation1, Variation2),
    checkaccidentals(Variation2,Variation).

/*****
                                LABELLING MELODIC SEQUENCES
*****/
/*****
STEP THROUGH THE ORIGINAL MELODY SEARCHING FOR MELODIC SEQUENCES CONSISTING
OF THREE OCCURRENCES EACH OF WHICH IS BETWEEN 4 AND 9 NOTES LONG.
IN THE PROCESS, PRODUCE A NEW LIST CALLED OrigSeq WHICH
CONTAINS THE ORIGINAL MELODY WITH SEQUENCES BRACKETED AND LABELLED.
A MELODIC SEQUENCE BEGINS WITH A MOTIF OF NOTES BETWEEN WHICH ARE CERTAIN INTERVALS.
THIS INTERVAL PATTERN IS REPEATED (TWO TIMES, IMMEDIATELY FOLLOWING THE ORIGINAL, IN
THIS CASE), WITH EACH REPETITION BEGINNING ON A DIFFERENT PITCH.
A SEQUENCE IS IDENTIFIED BY PATTERN MATCHING ADJACENT NOTES AND THEIR INTERVALS,
AND IS REPLACED IN THE NEW LIST BY GROUPING THE NOTES OF EACH SECTION OF THE
SEQUENCE AND ALSO OF THE SEQUENCE AS A WHOLE INTO A BRACKETED LIST STRUCTURE.
*****/

```



IF NO SEQUENCE IS FOUND BEGINNING WITH THE CURRENT NOTE, THAT NOTE IS ADDED UNCHANGED TO THE NEW LABELLED LIST AND THE SAME SEARCHING/MATCHING PROCEDURE CONTINUES WITH THE REST OF THE ORIGINAL PIECE.

\*\*\*\*\*/

```
labelsequences([ [CA1,RA1,[NA1]], [CA2,RA2,[NA2]], [CA3,RA3,[NA3]], [CA4,RA4,[NA4]],
  [CA5,RA5,[NA5]], [CA6,RA6,[NA6]], [CA7,RA7,[NA7]], [CA8,RA8,[NA8]],
  [CA9,RA9,[NA9]],
  [CB1,RB1,[NB1]], [CB2,RB2,[NB2]], [CB3,RB3,[NB3]], [CB4,RB4,[NB4]],
  [CB5,RB5,[NB5]], [CB6,RB6,[NB6]], [CB7,RB7,[NB7]], [CB8,RB8,[NB8]],
  [CB9,RB9,[NB9]],
  [CC1,RC1,[NC1]], [CC2,RC2,[NC2]], [CC3,RC3,[NC3]], [CC4,RC4,[NC4]],
  [CC5,RC5,[NC5]], [CC6,RC6,[NC6]], [CC7,RC7,[NC7]], [CC8,RC8,[NC8]],
  [CC9,RC9,[NC9]]|Ls],
  [ ['melseq', [ ['seq1', [ [CA1,RA1,[NA1]], [CA2,RA2,[NA2]], [CA3,RA3,[NA3]], [CA4,
    [CA5,RA5,[NA5]], [CA6,RA6,[NA6]], [CA7,RA7,[NA7]], [CA8,RA8,[NA8]],
    [CA9,RA9,[NA9]]]],
    ['seq2', [ [CB1,RB1,[NB1]], [CB2,RB2,[NB2]], [CB3,RB3,[NB3]], [CB4,RB4,[NB4]],
    [CB5,RB5,[NB5]], [CB6,RB6,[NB6]], [CB7,RB7,[NB7]], [CB8,RB8,[NB8]],
    [CB9,RB9,[NB9]]]],
    ['seq3', [ [CC1,RC1,[NC1]], [CC2,RC2,[NC2]], [CC3,RC3,[NC3]], [CC4,RC4,[NC4]],
    [CC5,RC5,[NC5]], [CC6,RC6,[NC6]], [CC7,RC7,[NC7]], [CC8,RC8,[NC8]],
    [CC9,RC9,[NC9]]]]]|NewLs)) :-
  sameinterval(NA1,NA2,NB1,NB2,NC1,NC2),
  sameinterval(NA2,NA3,NB2,NB3,NC2,NC3),
  sameinterval(NA3,NA4,NB3,NB4,NC3,NC4),
  sameinterval(NA4,NA5,NB4,NB5,NC4,NC5),
  sameinterval(NA5,NA6,NB5,NB6,NC5,NC6),
  sameinterval(NA6,NA7,NB6,NB7,NC6,NC7),
  sameinterval(NA7,NA8,NB7,NB8,NC7,NC8),
  sameinterval(NA8,NA9,NB8,NB9,NC8,NC9),
  labelsequences(Ls,NewLs).
```

```
labelsequences([ [CA1,RA1,[NA1]], [CA2,RA2,[NA2]], [CA3,RA3,[NA3]], [CA4,RA4,[NA4]],
  [CA5,RA5,[NA5]], [CA6,RA6,[NA6]], [CA7,RA7,[NA7]], [CA8,RA8,[NA8]],
  [CB1,RB1,[NB1]], [CB2,RB2,[NB2]], [CB3,RB3,[NB3]], [CB4,RB4,[NB4]],
  [CB5,RB5,[NB5]], [CB6,RB6,[NB6]], [CB7,RB7,[NB7]], [CB8,RB8,[NB8]],
  [CC1,RC1,[NC1]], [CC2,RC2,[NC2]], [CC3,RC3,[NC3]], [CC4,RC4,[NC4]],
  [CC5,RC5,[NC5]], [CC6,RC6,[NC6]], [CC7,RC7,[NC7]], [CC8,RC8,[NC8]]|Ls],
  [ ['melseq', [ ['seq1', [ [CA1,RA1,[NA1]], [CA2,RA2,[NA2]], [CA3,RA3,[NA3]], [CA4,
    [CA5,RA5,[NA5]], [CA6,RA6,[NA6]], [CA7,RA7,[NA7]], [CA8,RA8,[NA8]]]],
    ['seq2', [ [CB1,RB1,[NB1]], [CB2,RB2,[NB2]], [CB3,RB3,[NB3]], [CB4,RB4,[NB4]],
    [CB5,RB5,[NB5]], [CB6,RB6,[NB6]], [CB7,RB7,[NB7]], [CB8,RB8,[NB8]]]],
    ['seq3', [ [CC1,RC1,[NC1]], [CC2,RC2,[NC2]], [CC3,RC3,[NC3]], [CC4,RC4,[NC4]],
    [CC5,RC5,[NC5]], [CC6,RC6,[NC6]], [CC7,RC7,[NC7]], [CC8,RC8,[NC8]]]]]|NewLs
  sameinterval(NA1,NA2,NB1,NB2,NC1,NC2),
  sameinterval(NA2,NA3,NB2,NB3,NC2,NC3),
  sameinterval(NA3,NA4,NB3,NB4,NC3,NC4),
  sameinterval(NA4,NA5,NB4,NB5,NC4,NC5),
  sameinterval(NA5,NA6,NB5,NB6,NC5,NC6),
  sameinterval(NA6,NA7,NB6,NB7,NC6,NC7),
  sameinterval(NA7,NA8,NB7,NB8,NC7,NC8),
  labelsequences(Ls,NewLs).
```

```
labelsequences([ [CA1,RA1,[NA1]], [CA2,RA2,[NA2]], [CA3,RA3,[NA3]], [CA4,RA4,[NA4]],
  [CA5,RA5,[NA5]], [CA6,RA6,[NA6]], [CA7,RA7,[NA7]],
  [CB1,RB1,[NB1]], [CB2,RB2,[NB2]], [CB3,RB3,[NB3]], [CB4,RB4,[NB4]],
  [CB5,RB5,[NB5]], [CB6,RB6,[NB6]], [CB7,RB7,[NB7]],
  [CC1,RC1,[NC1]], [CC2,RC2,[NC2]], [CC3,RC3,[NC3]], [CC4,RC4,[NC4]],
  [CC5,RC5,[NC5]], [CC6,RC6,[NC6]], [CC7,RC7,[NC7]]|Ls],
  [ ['melseq', [ ['seq1', [ [CA1,RA1,[NA1]], [CA2,RA2,[NA2]], [CA3,RA3,[NA3]], [CA4,
    [CA5,RA5,[NA5]], [CA6,RA6,[NA6]], [CA7,RA7,[NA7]]]],
    ['seq2', [ [CB1,RB1,[NB1]], [CB2,RB2,[NB2]], [CB3,RB3,[NB3]], [CB4,RB4,[NB4]],
    [CB5,RB5,[NB5]], [CB6,RB6,[NB6]], [CB7,RB7,[NB7]]]]],
```

```

    ['seq3', [[CC1,RC1,[NC1]], [CC2,RC2,[NC2]], [CC3,RC3,[NC3]], [CC4,RC4,[NC4]],
    [CC5,RC5,[NC5]], [CC6,RC6,[NC6]], [CC7,RC7,[NC7]]]]|NewLs)) :-
sameinterval (NA1,NA2,NB1,NB2,NC1,NC2),
sameinterval (NA2,NA3,NB2,NB3,NC2,NC3),
sameinterval (NA3,NA4,NB3,NB4,NC3,NC4),
sameinterval (NA4,NA5,NB4,NB5,NC4,NC5),
sameinterval (NA5,NA6,NB5,NB6,NC5,NC6),
sameinterval (NA6,NA7,NB6,NB7,NC6,NC7),
labelsequences (Ls,NewLs) .

```

```

labelsequences ([ [CA1,RA1,[NA1]], [CA2,RA2,[NA2]], [CA3,RA3,[NA3]], [CA4,RA4,[NA4]],
[CA5,RA5,[NA5]], [CA6,RA6,[NA6]],
[CB1,RB1,[NB1]], [CB2,RB2,[NB2]], [CB3,RB3,[NB3]], [CB4,RB4,[NB4]],
[CB5,RB5,[NB5]], [CB6,RB6,[NB6]],
[CC1,RC1,[NC1]], [CC2,RC2,[NC2]], [CC3,RC3,[NC3]], [CC4,RC4,[NC4]],
[CC5,RC5,[NC5]], [CC6,RC6,[NC6]]|Ls],
[ ['melseq', [['seq1', [[CA1,RA1,[NA1]], [CA2,RA2,[NA2]], [CA3,RA3,[NA3]], [CA4,
[CA5,RA5,[NA5]], [CA6,RA6,[NA6]]]],
['seq2', [[CB1,RB1,[NB1]], [CB2,RB2,[NB2]], [CB3,RB3,[NB3]], [CB4,RB4,[NB4]],
[CB5,RB5,[NB5]], [CB6,RB6,[NB6]]]],
['seq3', [[CC1,RC1,[NC1]], [CC2,RC2,[NC2]], [CC3,RC3,[NC3]], [CC4,RC4,[NC4]],
[CC5,RC5,[NC5]], [CC6,RC6,[NC6]]]]]]|NewLs)) :-
sameinterval (NA1,NA2,NB1,NB2,NC1,NC2),
sameinterval (NA2,NA3,NB2,NB3,NC2,NC3),
sameinterval (NA3,NA4,NB3,NB4,NC3,NC4),
sameinterval (NA4,NA5,NB4,NB5,NC4,NC5),
sameinterval (NA5,NA6,NB5,NB6,NC5,NC6),
labelsequences (Ls,NewLs) .

```

```

labelsequences ([ [CA1,RA1,[NA1]], [CA2,RA2,[NA2]], [CA3,RA3,[NA3]], [CA4,RA4,[NA4]],
[CA5,RA5,[NA5]],
[CB1,RB1,[NB1]], [CB2,RB2,[NB2]], [CB3,RB3,[NB3]], [CB4,RB4,[NB4]],
[CB5,RB5,[NB5]],
[CC1,RC1,[NC1]], [CC2,RC2,[NC2]], [CC3,RC3,[NC3]], [CC4,RC4,[NC4]],
[CC5,RC5,[NC5]]|Ls],
[ ['melseq', [['seq1', [[CA1,RA1,[NA1]], [CA2,RA2,[NA2]], [CA3,RA3,[NA3]], [CA4,
[CA5,RA5,[NA5]]]],
['seq2', [[CB1,RB1,[NB1]], [CB2,RB2,[NB2]], [CB3,RB3,[NB3]], [CB4,RB4,[NB4]],
[CB5,RB5,[NB5]]]],
['seq3', [[CC1,RC1,[NC1]], [CC2,RC2,[NC2]], [CC3,RC3,[NC3]], [CC4,RC4,[NC4]],
[CC5,RC5,[NC5]]]]]]|NewLs)) :-
sameinterval (NA1,NA2,NB1,NB2,NC1,NC2),
sameinterval (NA2,NA3,NB2,NB3,NC2,NC3),
sameinterval (NA3,NA4,NB3,NB4,NC3,NC4),
sameinterval (NA4,NA5,NB4,NB5,NC4,NC5),
labelsequences (Ls,NewLs) .

```

```

labelsequences ([ [CA1,RA1,[NA1]], [CA2,RA2,[NA2]], [CA3,RA3,[NA3]], [CA4,RA4,[NA4]],
[CB1,RB1,[NB1]], [CB2,RB2,[NB2]], [CB3,RB3,[NB3]], [CB4,RB4,[NB4]],
[CC1,RC1,[NC1]], [CC2,RC2,[NC2]], [CC3,RC3,[NC3]], [CC4,RC4,[NC4]]|Ls],
[ ['melseq', [['seq1', [[CA1,RA1,[NA1]], [CA2,RA2,[NA2]], [CA3,RA3,[NA3]],
[CA4,RA4,[NA4]]]],
['seq2', [[CB1,RB1,[NB1]], [CB2,RB2,[NB2]], [CB3,RB3,[NB3]],
[CB4,RB4,[NB4]]]],
['seq3', [[CC1,RC1,[NC1]], [CC2,RC2,[NC2]], [CC3,RC3,[NC3]],
[CC4,RC4,[NC4]]]]]]|NewLs)) :-
sameinterval (NA1,NA2,NB1,NB2,NC1,NC2),
sameinterval (NA2,NA3,NB2,NB3,NC2,NC3),
sameinterval (NA3,NA4,NB3,NB4,NC3,NC4),
labelsequences (Ls,NewLs) .

```

```

labelsequences ([A|Ls], [A|NewLs]) :-
labelsequences (Ls,NewLs) .

```

```
labelsequences([], []).
```

```

/*****
RULE USED IN DETERMINING WHETHER THREE PAIRS OF NOTES (POTENTIALLY PART OF
A SEQUENCE) ALL HAVE THE SAME MELODIC INTERVAL.
*****/

```

```

sameinterval(A1,A2,B1,B2,C1,C2) :-
    (unison(A1,A2), unison(B1,B2), unison(C1,C2));
    (asc_min_second(A1,A2), asc_min_second(B1,B2), asc_min_second(C1,C2));
    (asc_second(A1,A2), asc_second(B1,B2), asc_second(C1,C2));
    (asc_third(A1,A2), asc_third(B1,B2), asc_third(C1,C2));
    (asc_fourth(A1,A2), asc_fourth(B1,B2), asc_fourth(C1,C2));
    (asc_fifth(A1,A2), asc_fifth(B1,B2), asc_fifth(C1,C2));
    (asc_sixth(A1,A2), asc_sixth(B1,B2), asc_sixth(C1,C2));
    (asc_octave(A1,A2), asc_octave(B1,B2), asc_octave(C1,C2));
    (desc_second(A1,A2), desc_second(B1,B2), desc_second(C1,C2));
    (desc_third(A1,A2), desc_third(B1,B2), desc_third(C1,C2));
    (desc_fourth(A1,A2), desc_fourth(B1,B2), desc_fourth(C1,C2));
    (desc_fifth(A1,A2), desc_fifth(B1,B2), desc_fifth(C1,C2));
    (desc_sixth(A1,A2), desc_sixth(B1,B2), desc_sixth(C1,C2));
    (desc_octave(A1,A2), desc_octave(B1,B2), desc_octave(C1,C2)).

```

```

/*****
                                MELODIC VARIATION
*****/

```

```

/*****
THIS GROUP OF CLAUSES CREATES NewLs, A LIST OF THE PIECE WITH MELODIC
MOTIF VARIATIONS SUBSTITUTED FOR ORIGINAL MOTIVES. THIS NEW LIST IS
CREATED BY STEPPING THROUGH THE FRAMEWORK AND THE ORIGINAL MELODY SIMULTANEOUSLY,
CHOOSING NEW MELODIC MOTIVES BASED ON THE SEQUENCES FOUND IN THE ORIGINAL,
WHAT CHANGES HAVE BEEN MADE THUS FAR IN THE VARIATION, AND THOSE MELODIC
MOTIVES WHICH ARE VALID GIVEN THE CURRENT AND UPCOMING NOTES IN THE
FRAMEWORK.
*****/

```

```

/*****
IF WE FIND THROUGH MATCHING THAT, BEGINNING AT THE CURRENT NOTE, THE ORIGINAL MELODY
HAS A SEQUENCE, THAT SEQUENCE WILL BE SUBSTITUTED/DEVELOPED FOR THE VARIATION
IN SUCH A WAY AS TO ENSURE THAT EACH PART OF THE SEQUENCE IS DEVELOPED USING THE
SAME MOTIF, THUS MIRRORING THE SEQUENCE IN THE ORIGINAL. IN THIS CASE, THE FIRST
SEQUENCE PART SUBSTITUTION IS CALLED NewLs1 AND THE SECOND AND THIRD ARE NewLs2 AND
NewLs3 RESPECTIVELY. ONCE THIS IS ACCOMPLISHED THE SEARCH/SUBSTITUTION
PROCESS RECURSIVELY CONTINUES ON THE REST OF THE PIECE.
*****/

```

```

checkmelframework(['melseq', Seq|RestOrig], Framework, SeqFlag,
                  [NewLs1, NewLs2, NewLs3|NewLs]) :-
    changemelwithsequence(['melseq', Seq], Framework,
                          RestFramework, NewLs1, NewLs2, NewLs3),
    checkmelframework(RestOrig, RestFramework, SeqFlag, NewLs).

```

```

/*****
IF THERE IS NO SEQUENCE, PATTERNS OF NOTES (FROM GROUPS OF 5 NOTES TO 2 NOTES)
FROM THE FRAMEWORK STARTING AT THE CURRENT PITCH ARE CHECKED TO SEE IF THERE IS
A MOTIF IN THE DATABASE THAT MATCHES THAT PATTERN. IF ONE IS FOUND, WE MUST MOVE
AHEAD IN THE ORIGINAL MELODY AS WELL AS IN THE FRAMEWORK, SO updateOriginalpointer
FINDS HOW MUCH OF THE ORIGINAL MELODY CORRESPONDS TO THE NOTES USED
FROM THE FRAMEWORK (THERE COULD BE SEVERAL NOTES IN THE ORIGINAL REPRESENTED
BY ONE IN THE FRAMEWORK). THUS WHEN WE CONTINUE WITH THE DEVELOPMENT OF
THE VARIATION, BOTH THE FRAMEWORK AND ORIGINAL ARE LINED UP.

```

```

*****/
checkmelframework(Original, [U,V,W,X,Y|Ls], SeqFlag, [Z|NewLs]) :-
    nosequenceinOriginal(Original, [U,V,W,X,Y], TempOriginal),
    change5bmel(U,V,W,X, Y, Z, SeqFlag),
    updateOriginalpointer(Original,TempOriginal,RestOriginal),
    checkmelframework(RestOriginal, Ls, SeqFlag, NewLs).

checkmelframework(Original, [V,W,X,Y|Ls], SeqFlag, [Z|NewLs]) :-
    nosequenceinOriginal(Original, [V,W,X,Y], TempOriginal),
    change4bmel(V,W,X,Y,Z, SeqFlag),
    updateOriginalpointer(Original,TempOriginal,RestOriginal),
    checkmelframework(RestOriginal, Ls, SeqFlag, NewLs).

checkmelframework(Original, [W,X,Y|Ls], SeqFlag, [Z|NewLs]) :-
    nosequenceinOriginal(Original, [W,X,Y], TempOriginal),
    change3bmel(W,X,Y,Z, SeqFlag),
    updateOriginalpointer(Original,TempOriginal,RestOriginal),
    checkmelframework(RestOriginal, Ls, SeqFlag, NewLs).

checkmelframework(Original, [X,Y|Ls], SeqFlag, [Z|NewLs]) :-
    nosequenceinOriginal(Original, [X,Y], TempOriginal),
    change2bmel(X, Y, Z, SeqFlag),
    updateOriginalpointer(Original,TempOriginal,RestOriginal),
    checkmelframework(RestOriginal, Ls, SeqFlag, NewLs).

/*****
CASE WHERE WANT ONE FRAMEWORK NOTE TO PASS THROUGH (AND AS MANY NOTES IN
ORIGINAL AS THAT MIRRORS.
*****/
checkmelframework(Original,[X,Y|Ls], SeqFlag, [X|NewLs]) :-
    onenotenosequenceinOriginal(Original, [X,Y], TempOriginal),
    updateOriginalpointer(Original,TempOriginal, RestOriginal),
    checkmelframework(RestOriginal,[Y|Ls], SeqFlag, NewLs).

/*****
COVERS LAST NOTE OF FRAMEWORK IF IT IS ALSO THE LAST SINGLE NOTE OF
THE ORIGINAL.
*****/
checkmelframework([X],[X],_, [X]) :-
    checkmelframework(_,[],_, []).

/*****
SPECIAL CASE - WHEN VARYING THE FIRST PART OF A SEQUENCE, s FLAG SENT THROUGH
FOR "ORIGINAL" PARAMETER SINCE THE ORIGINAL IS UPDATED DIFFERENTLY IN THE
SEQUENTIAL CASE. THIS s WILL PASS THROUGH IN THE BEGINNING STAGES OF THE
checkmelframework RECURSION, BUT WHEN THERE IS NO LONGER A VARIABLE IN THAT
POSITION, BUT RATHER A LAST ELEMENT CASE, THE REGULAR checkmelframework WITH
[X] AS THE ORIGINAL LIST PARAMETER WILL NOT MATCH.
*****/
checkmelframework(s,[X],_, [X]) :-
    checkmelframework(_,[],_, []).

checkmelframework(_,[],_, []).

/*****
CHECKING AT CURRENT NOTE FOR SEQUENCE
*****/
/*****
GIVEN THE ORIGINAL AND A LIST OF FRAMEWORK NOTES, WANT TO SEE IF THERE IS A SEQUENCE
BEGINNING AT THAT FIRST FRAMEWORK/ORIGINAL NOTE. THIS IS CHECKED BY ATTEMPTING
TO MAP THE FRAMEWORK GROUP OF NOTES TO A PORTION OF THE ORIGINAL LIST,
UNDERSTANDING THAT THE ORIGINAL MAY HAVE SEVERAL NOTES BETWEEN ANY TWO THAT

```

ARE FOUND IN THE FRAMEWORK. IF THE ORIGINAL MAPS TO THE FRAMEWORK NOTES GIVEN, THEN THERE IS NOT A SEQUENCE, SINCE THE 'MELSEQ' LABEL IS NOT FOUND IN THE MAPPING CLAUSES AND SO, IF IT OCCURS IN THE ORIGINAL, THE nosequence CLAUSES WILL FAIL. CONVERSELY, IF THE MAPPING SUCCEEDS, THERE IS NO SEQUENCE, AND THE SECTION OF THE ORIGINAL MELODY THAT CORRESPONDS TO THE GIVEN FRAMEWORK NOTES IS SAVED IN TempOriginal.

onenote ALLOWS FOR CASES WHERE WE WANT ONE FRAMEWORK NOTE TO PASS THROUGH THE VARIATION AND NEED TO FIND THE CORRESPONDING ORIGINAL NOTES TO PLACE IN TempOriginal (IN ORDER TO BE CONSISTENT WITH OTHER SITUATIONS WHERE MORE THAN ONE FRAMEWORK NOTE IS SEEN), OF WHICH THERE MAY BE MORE THAN ONE.

\*\*\*\*\*/

```
nosequenceinOriginal(s,_,_) :- !.
```

```
nosequenceinOriginal(Original, FrameworkNotes, TempOriginal) :-
    mapFrameworkintervaltoOrig(Original, FrameworkNotes, TempOriginal),!.
```

```
onenotenosequenceinOriginal(s,_,_) :- !.
```

```
onenotenosequenceinOriginal(Original, FrameworkNotes, TempOriginal) :-
    onenotemapFrameworkintervaltoOrig(Original, FrameworkNotes, TempOriginal).
```

\*\*\*\*\*  
 MAPPING A GROUP OF FRAMEWORK NOTES TO THEIR CORRESPONDING NOTES IN THE ORIGINAL IS ACHIEVED BY RECURSIVELY FINDING THE ORIGINAL NOTES FOR EACH FRAMEWORK INTERVAL REPRESENTED IN THE FRAMEWORK GROUP OF NOTES BEING EXAMINED. THESE ORIGINAL NOTES FOR ONE FRAMEWORK INTERVAL COULD INCLUDE SITUATIONS WHERE: THE ORIGINAL HAS ONLY THE SAME TWO NOTES AS THE FRAMEWORK INTERVAL, THERE ARE UP TO 5 EXTRA ORIGINAL NOTES BETWEEN THE TWO FRAMEWORK NOTES, EITHER OF THE ABOVE BUT AT THE LAST FRAMEWORK INTERVAL IN THE LIST BEING EXAMINED, THE LAST ORIGINAL NOTE AND THE LAST FRAMEWORK NOTE (NO INTERVALS ARE LSFT) ARE ALL THAT REMAIN, OR THE ORIGINAL HAS UP TO 4 EXTRA NOTES BEFORE THE NEXT FRAMEWORK NOTE.

\*\*\*\*\*/

```
mapFrameworkintervaltoOrig([ [CA,RA,NA], [CB,RB,NB] | RestOrig],
    [ [_,_,NA], [_,_,NB] | RestFrame],
    [ [CA,RA,NA] | TempOrig]) :-
    mapFrameworkintervaltoOrig([ [CB,RB,NB] | RestOrig], [ [_,_,NB] | RestFrame], TempOrig).
```

```
mapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [CB,RB,NB] | RestOrig],
    [ [_,_,NA], [ $, $, NB] | RestFrame],
    [ [CA,RA,NA], [C1,R1,N1] | TempOrig]) :-
    mapFrameworkintervaltoOrig([ [CB,RB,NB] | RestOrig], [ [ $, $, NB] | RestFrame], TempOrig).
```

```
mapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [CB,RB,NB] | RestOrig],
    [ [_,_,NA], [_,_,NB] | RestFrame],
    [ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2] | TempOrig]) :-
    mapFrameworkintervaltoOrig([ [CB,RB,NB] | RestOrig], [ [_,_,NB] | RestFrame], TempOrig).
```

```
mapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [CB,RB,NB] | RestOrig],
    [ [_,_,NA], [_,_,NB] | RestFrame],
    [ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3] | TempOrig]) :-
    mapFrameworkintervaltoOrig([ [CB,RB,NB] | RestOrig], [ [_,_,NB] | RestFrame], TempOrig).
```

```
mapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4],
    [CB,RB,NB] | RestOrig],
    [ [_,_,NA], [_,_,NB] | RestFrame],
    [ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4] | TempOrig]) :-
    mapFrameworkintervaltoOrig([ [CB,RB,NB] | RestOrig], [ [_,_,NB] | RestFrame], TempOrig).
```

```
mapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4],
    [C5,R5,N5], [CB,RB,NB] | RestOrig],
```

```

    [ [_,_,NA], [_,_,NB] | RestFrame ],
    [ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4],
      [C5,R5,N5] | TempOrig ] ) :-
    mapFrameworkintervaltoOrig([ [CB,RB,NB] | RestOrig], [ [_,_,NB] | RestFrame], TempOrig).

mapFrameworkintervaltoOrig([ [CA,RA,NA], [CB,RB,NB] | _],
    [ [_,_,NA], [_,_,NB] ],
    [ [CA,RA,NA], [CB,RB,NB] ] ).

mapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [CB,RB,NB] | _],
    [ [_,_,NA], [_,_,NB] ],
    [ [CA,RA,NA], [C1,R1,N1], [CB,RB,NB] ] ).

mapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [CB,RB,NB] | _],
    [ [_,_,NA], [_,_,NB] ],
    [ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [CB,RB,NB] ] ).

mapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [CB,RB,NB] | _],
    [ [_,_,NA], [_,_,NB] ],
    [ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [CB,RB,NB] ] ).

mapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4],
    [CB,RB,NB] | _],
    [ [_,_,NA], [_,_,NB] ],
    [ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4],
      [CB,RB,NB] ] ).

mapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4],
    [C5,R5,N5], [CB,RB,NB] | _],
    [ [_,_,NA], [_,_,NB] ],
    [ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4],
      [C5,R5,N5], [CB,RB,NB] ] ).

mapFrameworkintervaltoOrig([ [CA,RA,NA] | _], [ [_,_,NA] ], [ [CA,RA,NA] ] ).

mapFrameworkintervaltoOrig([ [C1,R1,N1], [CA,RA,NA] | RestOrig],
    [ [_,_,NA] | RestFrame], [ [C1,R1,N1] | TempOrig ] ) :-
    mapFrameworkintervaltoOrig([ [CA,RA,NA] | RestOrig], [ [_,_,NA] | RestFrame], TempOrig).

mapFrameworkintervaltoOrig([ [C1,R1,N1], [C2,R2,N2], [CA,RA,NA] | RestOrig],
    [ [_,_,NA] | RestFrame], [ [C1,R1,N1], [C2,R2,N2] | TempOrig ] ) :-
    mapFrameworkintervaltoOrig([ [CA,RA,NA] | RestOrig], [ [_,_,NA] | RestFrame], TempOrig).

mapFrameworkintervaltoOrig([ [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [CA,RA,NA] | RestOrig],
    [ [_,_,NA] | RestFrame], [ [C1,R1,N1], [C2,R2,N2], [C3,R3,N3] | TempOrig ]
    mapFrameworkintervaltoOrig([ [CA,RA,NA] | RestOrig], [ [_,_,NA] | RestFrame], TempOrig).

mapFrameworkintervaltoOrig([ [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4], [CA,RA,NA] | RestOrig],
    [ [_,_,NA] | RestFrame], [ [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4] | TempOrig ]
    mapFrameworkintervaltoOrig([ [CA,RA,NA] | RestOrig], [ [_,_,NA] | RestFrame], TempOrig).

/*****
WHEN ALLOWING ONLY ONE FRAMEWORK NOTE THROUGH, MAPPING THE ORIGINAL TO A SINGLE
FRAMEWORK NOTE TAKES ORIGINAL NOTES UP TO BUT NOT INCLUDING THE NEXT FRAMEWORK
NOTE OR TO THE BEGINNING OF A MELODIC SEQUENCE.
*****/

onenotemapFrameworkintervaltoOrig([ [CA,RA,NA], [_,_,NB] | _],
    [ [_,_,NA], [_,_,NB] ],
    [ [CA,RA,NA] ] ).

onenotemapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [_,_,NB] | _],

```

$[[\_ ,\_ ,NA], [\_ ,\_ ,NB]],$   
 $[[CA,RA,NA], [C1,R1,N1]]).$

$onenotemapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [\_ ,\_ ,NB] | \_ ,$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,NB]],$   
 $[[CA,RA,NA], [C1,R1,N1], [C2,R2,N2]]).$

$onenotemapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [\_ ,\_ ,NB] | \_ ,$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,NB]],$   
 $[[CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3]]).$

$onenotemapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4]$   
 $[\_ ,\_ ,NB] | \_ ,$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,NB]],$   
 $[[CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4]]).$

$onenotemapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4]$   
 $[C5,R5,N5], [\_ ,\_ ,NB] | \_ ,$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,NB]],$   
 $[[CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4],$   
 $[C5,R5,N5]]).$

$onenotemapFrameworkintervaltoOrig([ [CA,RA,NA] | \_ , [\_ ,\_ ,NA]], [[CA,RA,NA]]).$

$onenotemapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4]$   
 $[C5,R5,N5], ['melseq' | \_ ] | \_ ,$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,\_ ]],$   
 $[[CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4],$   
 $[C5,R5,N5]]).$

$onenotemapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4]$   
 $['melseq' | \_ ] | \_ ,$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,\_ ]],$   
 $[[CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4]]).$

$onenotemapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], ['melseq' |$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,\_ ]],$   
 $[[CA,RA,NA], [C1,R1,N1], [C2,R2,N2], [C3,R3,N3]]).$

$onenotemapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], [C2,R2,N2], ['melseq' | \_ ] | \_ ,$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,\_ ]],$   
 $[[CA,RA,NA], [C1,R1,N1], [C2,R2,N2]]).$

$onenotemapFrameworkintervaltoOrig([ [CA,RA,NA], [C1,R1,N1], ['melseq' | \_ ] | \_ ,$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,\_ ]],$   
 $[[CA,RA,NA], [C1,R1,N1]]).$

$onenotemapFrameworkintervaltoOrig([ [CA,RA,NA], ['melseq' | \_ ] | \_ ,$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,\_ ]],$   
 $[[CA,RA,NA]]).$

$onenotemapFrameworkintervaltoOrig([ [C1,R1,N1], [CA,RA,NA] | \_ ,$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,\_ ]], [[C1,R1,N1], [CA,RA,NA]]).$

$onenotemapFrameworkintervaltoOrig([ [C1,R1,N1], [C2,R2,N2], [CA,RA,NA] | \_ ,$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,\_ ]], [[C1,R1,N1], [C2,R2,N2], [CA,RA,NA]]).$

$onenotemapFrameworkintervaltoOrig([ [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [CA,RA,NA] | \_ ,$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,\_ ]], [[C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [CA,RA,NA]]).$

$onenotemapFrameworkintervaltoOrig([ [C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4], [CA,RA,NA]$   
 $[\_ ,\_ ,NA], [\_ ,\_ ,\_ ]], [[C1,R1,N1], [C2,R2,N2], [C3,R3,N3], [C4,R4,N4]],$

```

/*****
                                CHOOSING A SUBSTITUTION
*****/

```

```

/*****
GIVEN A GROUP OF CONSECUTIVE NOTES FROM THE FRAMEWORK, THIS CLAUSE ATTEMPTS TO
FIND A MELODIC MOTIF IN THE DATABASE THAT MATCHES THESE NOTES (ACTUALLY,
THE INTERVALS BETWEEN THEM ARE USED TO DETERMINE A MATCH). IF THERE
ARE SEVERAL POSSIBLE SUBSTITUTIONS FOR A MOTIF THAT MATCHES, THE SYSTEM
RANDOMLY CHOOSES BETWEEN THESE VALID POSSIBILITIES, WITH AN ADDED CONSTRAINT
WHICH KEEPS TRACK OF THE NUMBER OF TIMES THAT PARTICULAR POSSIBILITY HAS
BEEN CHOSEN IN THE PAST AND SETS PRIORITIES TO FIRST FIND ONE THAT HAS BEEN USED
ONCE, THEN THREE TIMES, THEN NEVER, THEN TWICE, IN AN EFFORT TO MODEL
THE EXPERT'S ATTEMPT TO KEEP SYMMETRY WITHIN THE PIECE BUT TO ALSO AVOID
TOO MUCH REPETITION WHEN POSSIBLE.
FOR EFFICIENCY PURPOSES, BEFORE ENTERING THE RANDOM CHOICE SECTION, THE VOCAB
SECTION OF THE DATABASE IS CHECKED TO MAKE SURE THAT THERE IS AT LEAST ONE VALID
MATCH FOR THE NOTE PATTERN.
IF THESE CLAUSES ARE BEING USED IN THE MIDST OF DEVELOPING A SEQUENCE (THE
FIRST OCCURRENCE WITHIN THE SEQUENCE IS GENERATED RANDOMLY THE SAME AS
NON-SEQUENCE PATTERNS), A FACT IS ADDED TO THE DATABASE STATING THE ID NUMBER OF THE
MOTIVE THAT WAS USED TO DEVELOP A SPECIFIC INTERVAL SO THAT IN THE
SUBSTITUTION FOR THE OTHER SEQUENCE PARTS THE SAME MOTIF CAN BE USED BY CHECKING
THESE FACTS IN THE DATABASE.
*****/

```

```

change5bmel ([[C1],[R1],[N1]], [[C2],[R2],[N2]], [[C3],[R3],[N3]], [[C4],[R4],[N4]], [[C5],[R5]
    NewMel, SeqFlag) :-
    melvocab(Interval,_, [N1,N2,N3,N4,N5],_),
    (
        rndmchoice(0, [N1,N2,N3,N4,N5], 1, Interval, MelID, NewN);
        rndmchoice(0, [N1,N2,N3,N4,N5], 3, Interval, MelID, NewN);
        rndmchoice(0, [N1,N2,N3,N4,N5], 0, Interval, MelID, NewN);
        rndmchoice(0, [N1,N2,N3,N4,N5], 2, Interval, MelID, NewN);
        rndmchoice(0, [N1,N2,N3,N4,N5], _, Interval, MelID, NewN)
    ),
    (
        (SeqFlag = 'set',
         assert(seq(Interval, MelID))) ;
        true
    ),
    NewMel = [[C1,C2,C3,C4,C5], [R1,R2,R3,R4,R5], NewN].

```

```

change4bmel ([[C1],[R1],[N1]], [[C2],[R2],[N2]], [[C3],[R3],[N3]], [[C4],[R4],[N4]],
    NewMel, SeqFlag) :-
    melvocab(Interval,_, [N1,N2,N3,N4],_),
    (
        rndmchoice(0, [N1,N2,N3,N4], 1, Interval, MelID, NewN);
        rndmchoice(0, [N1,N2,N3,N4], 3, Interval, MelID, NewN);
        rndmchoice(0, [N1,N2,N3,N4], 0, Interval, MelID, NewN);
        rndmchoice(0, [N1,N2,N3,N4], 2, Interval, MelID, NewN);
        rndmchoice(0, [N1,N2,N3,N4], _, Interval, MelID, NewN)
    ),
    (
        (SeqFlag = 'set',
         assert(seq(Interval, MelID))) ;
        true
    ),
    NewMel = [[C1,C2,C3,C4], [R1,R2,R3,R4], NewN].

```

```

change3bmel ([[C1],[R1],[N1]], [[C2],[R2],[N2]], [[C3],[R3],[N3]], NewMel, SeqFlag) :-
    melvocab(Interval,_, [N1,N2,N3],_),
    (

```



```

    rndmchoice(0,[N1,N2,N3],1,Interval,MelID,NewN);
    rndmchoice(0,[N1,N2,N3],3,Interval,MelID,NewN);
    rndmchoice(0,[N1,N2,N3],0,Interval,MelID,NewN);
    rndmchoice(0,[N1,N2,N3],2,Interval,MelID,NewN);
    rndmchoice(0,[N1,N2,N3],_,Interval,MelID,NewN)
),
(
    (SeqFlag = 'set',
    assert(seq(Interval, MelID)));
    true
),
NewMel = [[C1,C2,C3],[R1,R2,R3],NewN].

change2bmel([[C1],[R1],[N1]], [[C2],[R2],[N2]], NewMel, SeqFlag) :-
    melvocab(Interval,_,[N1,N2],_),
    (
        rndmchoice(0,[N1,N2],1,Interval,MelID,NewN);
        rndmchoice(0,[N1,N2],3,Interval,MelID,NewN);
        rndmchoice(0,[N1,N2],0,Interval,MelID,NewN);
        rndmchoice(0,[N1,N2],2,Interval,MelID,NewN);
        rndmchoice(0,[N1,N2],_,Interval,MelID,NewN)
    ),
    (
        (SeqFlag = 'set',
        assert(seq(Interval, MelID)));
        true
    ),
    NewMel = [[C1,C2],[R1,R2],NewN].

/*****
USED TO CHOOSE BETWEEN SEVERAL VALID MELODIC VOCABULARY OPTIONS.  SINCE
CURRENTLY THERE ARE AT MOST 4 VALID OPTIONS FOR ANY INTERVAL, ONCE
ALL FOUR OPTIONS HAVE BEEN TRIED, IF NONE MEETS ALL OF THE CONSTRAINTS THEN
WE FAIL AND BACKTRACK TO ANOTHER FREQUENCY PRIORITY.  ADDING AN "ALREADYTRIED" FACT
TO THE DATABASE IF A CERTAIN OF THE OPTIONS HAS BEEN TRIED AND DID NOT
MEET FREQUENCY AND VOCABULARY CONSTRAINTS KEEPS IT FROM BEING TESTED AGAIN,
EVEN IF THE RANDOM NUMBER GENERATOR COMES UP WITH A NUMBER IN THE RANGE
ASSIGNED TO THAT OPTION.  ONCE AN OPTION IS FOUND THAT SATISFIES ALL
CONSTRAINTS, THE ALREADYTRIED FACTS ARE REMOVED FROM THE DATABASE AND
THE FREQUENCY FACT REGARDING THE NUMBER OF USES OF THIS PARTICULAR OPTION
IS UPDATED.
FOR EFFICIENCY PURPOSES, BEFORE ANY RANDOM CHOICE IS MADE, THE DATABASE IS
CHECKED TO MAKE SURE THERE IS AT LEAST ONE MELODIC VOCABULARY OPTION WITH
THE SPECIFIED FREQUENCY.  IF NOT, RATHER THAN PROCEEDING, THE RNDMCHOICE CLAUSE
WILL FAIL, CAUSING BACKTRACKING TO TRY ANOTHER FREQUENCY.
*****/

rndmchoice(Total, Notes, Freq, Interval,MelID, NewN) :-
    ((Total = 4; not(freq(Interval,_,Freq))),
    retractall(alreadytried($)),!,fail);
    (
        random(RndmNum),
        getID(TempMelID, RndmNum),
        ((alreadytried(TempMelID),
        !,rndmchoice(Total, Notes, Freq, Interval,MelID, NewN));
        (
            (freq(Interval,TempMelID,Freq),
            melvocab(Interval,TempMelID,Notes,NewN),
            retractall(alreadytried($)),
            MelID is TempMelID,
            retract(freq(Interval,MelID,Freq)),
            Freq1 is 1 + Freq,

```

```

    assert(freq(Interval,MelID,Freq1))
    );
    (assert(alreadytried(TempMelID)),
    Total1 is 1 + Total,
    !,rndmchoice(Total1, Notes, Freq, Interval,MelID, NewN)
    )
    )
    ).

/*****
GIVEN THE RANDOM NUMBER GENERATOR RANGE OF APPROX. -32000 TO 32000, AND ASSUMING
THAT THERE ARE FOUR POSSIBILITIES FROM WHICH TO MAKE A CHOICE, THE RANDOM
RANGE IS DIVIDED INTO FOUR SECTIONS, EACH RETURNING A SPECIFIC ID NUMBER, 1 TO 4.
*****/

getID(1,Rndm) :-
    Rndm < -16000.
getID(2,Rndm) :-
    Rndm < 0.
getID(3,Rndm) :-
    Rndm < 16000.
getID(4,_).

/*****
                        KEEPING LOCATION IN ORIGINAL PARALLEL TO FRAMEWORK
*****/
/*****
WHEN THERE IS NO SEQUENCE IN THE ORIGINAL, THE FRAMEWORK "POINTER" OF
WHERE WE ARE IN CREATING THE VARIATION IS AUTOMATICALLY UPDATED THROUGH
THE UNIFICATION PROCESS. THE ORIGINAL VERSION MUST ALSO MOVE PAST
THOSE NOTES WHICH CORRESPOND TO THOSE DEVELOPED IN THE FRAMEWORK.
SINCE "TempOriginal" CONTAINS THOSE NOTES WHICH MIRROR THE FRAMEWORK NOTES
WHICH HAVE BEEN USED, WE CAN ASSUME THAT "TempOriginal" IS AN INITIAL SUBLIST OF
"Original" AND THAT WHATEVER ELSE FOLLOWS THAT SUBLIST WILL BE THE REST OF THE
ORIGINAL.
IN THE SPECIAL CASE WHERE THE FIRST PART OF A SEQUENCE IS BEING DEVELOPED, THE
s FLAG SIGNALS THAT NO UPDATE NEED BE DONE HERE.
*****/

updateOriginalpointer(s,_,s).

updateOriginalpointer(Original, TempOriginal, RestOriginal) :-
    decompose(Original, TempOriginal, RestOriginal).

/*****
THE COROLLARY TO CONCATENATE, DECOMPOSE TAKES THE LIST GIVEN AS ITS FIRST
PARAMETER AND AN INITIAL SUBLIST OF THIS FIRST LIST GIVEN IN THE SECOND
PARAMETER, AND GIVES THE REST OF THE ORIGINAL LIST AS THE THIRD PARAMETER.
*****/

decompose(L, [],L).
decompose([X|L1],[X|L2],L3) :-
    decompose(L1,L2,L3).

/*****
                        CHANGING A SEQUENTIAL SECTION OF THE MELODY
*****/

```

```

/*****
GIVEN A SEQUENCE MARKED IN THE ORIGINAL VERSION OF THE MELODY, WE MUST
MAP THAT SEQUENCE ONTO THE FRAMEWORK SO THAT IT CAN BE VARIED FROM THAT
BASE.  ONCE THESE OCCURRENCES ARE FOUND IN THE FRAMEWORK, THE FIRST
IS VARIED AS A NON-SEQUENTIAL PATTERN WOULD BE (USING THE VOCABULARY AND
THE RANDOM FUNCTION WITH PRIORITIES).  THE SECOND AND THIRD OCCURRENCES
ARE THEN DEVELOPED TO MODEL THE FIRST, LEAVING THE SEQUENCE INTACT.  FACTS
ADDED TO THE DATABASE REGARDING THIS PARTICULAR SEQUENCE ARE REMOVED ONCE
THE SEQUENCE IS DEVELOPED.
*****/

```

```

changemelwithsequence(['melseq',[['seq1',OrigNotes1],['seq2',OrigNotes2],
                                   ['seq3',OrigNotes3]]], Framework, RestFramework,
                      NewLs1, NewLs2, NewLs3) :-
    findframeworkseqboundaries(OrigNotes1, OrigNotes2, OrigNotes3, Framework,
                               Framework1, Framework2, Framework3, RestFramework),
    checkmelframework(s,Framework1, 'set',NewLs1),
    finishseqmelframework(Framework2, NewLs2),
    finishseqmelframework(Framework3, NewLs3),
    retractall(seq($)).

```

```

/*****
TRACES THE GROUND COVERED BY THE SEQUENCE IN THE ORIGINAL VERSION IN
THE FRAMEWORK SO THAT THE ACTUAL MELODIC SUBSTITUTIONS CAN BE DONE ON
THE SEQUENCE AS IT OCCURS IN THE FRAMEWORK, THE PLACE WHERE THE VARIATIONS
ARE TYPICALLY BASED.  THE FRAMEWORK SEQUENCE IS STORED IN FRAMEWORK1 THROUGH
3 AND THE REMAINDER OF THE FRAMEWORK (WHICH WILL STILL NEED TO BE
DEVELOPED) IS STORED IN RESTFRAMEWORK.
*****/

```

```

findframeworkseqboundaries(OrigNotes1, OrigNotes2, OrigNotes3, Framework, Framework1,
                           Framework2, Framework3, RestFramework) :-
    findseqboundary(OrigNotes1, Framework, Framework1, RestFramework1),
    findseqboundary(OrigNotes2, RestFramework1, Framework2, RestFramework2),
    findseqboundary(OrigNotes3, RestFramework2, Framework3, RestFramework).

```

```

/*****
THE NOTES BETWEEN X AND Y ARE NOTES THAT ARE FOUND IN THE ORIGINAL VERSION
INTERVAL BUT ARE NOT IN THE FRAMEWORK OF THAT SAME PATTERN - ONLY THE
BOUNDARY NOTES X AND Y OCCUR IN BOTH.  THE FRAMEWORK NOTES ARE ADDED TO
THE TEMPORARY STORAGE OF THAT SEQUENCE OCCURRENCE.  IF THERE ARE EXTRA NOTES
IN THE ORIGINAL BEFORE THE FRAMEWORK NOTE WHICH MARKS THE BEGINNING OF THE
NEXT SEQUENCE OCCURRENCE, WE MUST MOVE PAST THEM.
*****/

```

```

findseqboundary([[_],[_],[X]],
                [[_],[_],[Y]]|RestOrig, [[C6],[R6],[X]],[[C7],[R7],[Y]]|Rest,
                [[C6],[R6],[X]]|Frmwrk,ReturnRest) :-
    findseqboundary([[_],[_],[Y]]|RestOrig, [[C7],[R7],[Y]]|Rest, Frmwrk, ReturnF

findseqboundary([[_],[_],[X]],[_],[_],
                [[_],[_],[Y]]|RestOrig, [[C6],[R6],[X]],[[C7],[R7],[Y]]|Rest,
                [[C6],[R6],[X]]|Frmwrk,ReturnRest) :-
    findseqboundary([[_],[_],[Y]]|RestOrig, [[C7],[R7],[Y]]|Rest, Frmwrk, ReturnF

findseqboundary([[_],[_],[X]],[_],[_],[_],[_],
                [[_],[_],[Y]]|RestOrig, [[C6],[R6],[X]],[[C7],[R7],[Y]]|Rest,
                [[C6],[R6],[X]]|Frmwrk,ReturnRest) :-
    findseqboundary([[_],[_],[Y]]|RestOrig, [[C7],[R7],[Y]]|Rest, Frmwrk, ReturnF

findseqboundary([[_],[_],[X]],[_],[_],[_],[_],[_],
                [[_],[_],[Y]]|RestOrig, [[C6],[R6],[X]],[[C7],[R7],[Y]]|Rest,
                [[_],[_],[Y]]|RestOrig, [[C6],[R6],[X]],[[C7],[R7],[Y]]|Rest,

```

```

        [[C6],[R6],[X]]|Frmwrk],ReturnRest) :-
    findseqboundary([[[_],[_],[Y]]|RestOrig], [[C7],[R7],[Y]]|Rest], Frmwrk, ReturnF

findseqboundary([[_],[_],[Y]]|RestOrig], [[C7],[R7],[Y]]|Rest],
    [[C7],[R7],[Y]]|Frmwrk],ReturnRest) :-
    findseqboundary(RestOrig, Rest, Frmwrk, ReturnRest).

findseqboundary([], Rest, [], Rest).

findseqboundary([[_],[_],[_]],[_],[_],[_]],[_],[_],[_]],
    [_],[_],[_]], Rest,
    [],Rest).

findseqboundary([[_],[_],[_]],[_],[_],[_]],[_],[_],[_]], Rest,
    [],Rest).

findseqboundary([[_],[_],[_]],[_],[_],[_]], Rest, [],Rest).

findseqboundary([[_],[_],[_]], Rest, [],Rest).

/*****
ONCE THE FIRST OCCURRENCE OF A SEQUENCE IS DEVELOPED, USE THE
SAME MELODIC SUBSTITUTIONS FOR THE SECOND AND THIRD OCCURRENCES.
*****/

finishseqmelframework([U,V,W,X,Y|Ls], [Z|NewLs]) :-
    change5bseqmel(U,V,W,X,Y,Z),
    finishseqmelframework(Ls, NewLs).

finishseqmelframework([V,W,X,Y|Ls], [Z|NewLs]) :-
    change4bseqmel(V,W,X,Y,Z),
    finishseqmelframework(Ls, NewLs).

finishseqmelframework([W,X,Y|Ls], [Z|NewLs]) :-
    change3bseqmel(W,X,Y,Z),
    finishseqmelframework(Ls, NewLs).

finishseqmelframework([X,Y|Ls], [Z|NewLs]) :-
    change2bseqmel(X,Y,Z),
    finishseqmelframework(Ls, NewLs).

finishseqmelframework([X|Ls], [X|NewLs]) :-
    finishseqmelframework(Ls, NewLs).

finishseqmelframework([], []).

/*****
APPROACH THE SUBSTITUTION OF MELODIC VOCAB WITHIN THE SEQUENCE BY FINDING AN
INTERVAL THAT MATCHES THE PATTERN AND THEN CHECKING TO BE SURE THAT THE
PARTICULAR VARIATION OF THAT ORIGINAL INTERVAL CHOSEN IS THE SAME ONE THAT WAS
USED IN CHANGING THE FIRST PART OF THE SEQUENCE (BY CHECKING THE DATABASE FOR
seq FACT ABOUT THAT MelID NUMBER.
*****/

change5bseqmel([C1],[R1],[N1]], [C2],[R2],[N2]], [C3],[R3],[N3]], [C4],[R4],[N4]],
    [C5],[R5],[N5]],NewMel) :-
    melvocab(Interval, MelID, [N1,N2,N3,N4,N5], NewN),
    seq(Interval, MelID),
    NewMel = [C1,C2,C3,C4,C5], [R1,R2,R3,R4,R5],NewN].

change4bseqmel([C1],[R1],[N1]], [C2],[R2],[N2]], [C3],[R3],[N3]], [C4],[R4],[N4]],
    NewMel) :-

```

```

melvocab(Interval, MelID, [N1,N2,N3,N4], NewN),
seq(Interval, MelID),
NewMel = [[C1,C2,C3,C4],[R1,R2,R3,R4],NewN].

```

```

change3bseqmel([[C1],[R1],[N1]],[[C2],[R2],[N2]],[[C3],[R3],[N3]],NewMel) :-
melvocab(Interval, MelID, [N1,N2,N3], NewN),
seq(Interval, MelID),
NewMel = [[C1,C2,C3],[R1,R2,R3],NewN].

```

```

change2bseqmel([[C1],[R1],[N1]],[[C2],[R2],[N2]],NewMel) :-
melvocab(Interval, MelID, [N1,N2], NewN),
seq(Interval, MelID),
NewMel = [[C1,C2],[R1,R2],NewN].

```

```

/*****
                                RHYTHMIC VARIATION
*****/

/*****
GIVEN A LIST WHICH REPRESENTS THE ORIGINAL AND FRAMEWORK WITH MELODIC VARIATION
INCLUDED, THE RHYTHMIC VARIATION SECTION DETERMINES WHETHER A SEQUENCE HAS BEEN
INVOLVED OR NOT AND MODIFIES EITHER A SUBLIST WHICH IS NON-SEQUENTIAL (WHICH
WOULD BE REPRESENTED AS THE FIRST ELEMENT OF THE LIST BEING ADDRESSED) OR A SUBLIST
WHICH DOES INCLUDE A SEQUENCE (IN WHICH THE FIRST ELEMENT OF THE LIST WOULD
BE A LIST ITSELF- EITHER OF ONE NOTE GROUP IF THERE WAS ONLY ONE INTERVAL
CHANGED WITHIN EACH PART OF THE SEQUENCE, OR TWO OR THREE NOTE GROUPS IF MORE
INTERVAL CHANGES WERE INVOLVED). THE MODIFICATION OCCURS BY FINDING THE NUMBER OF
NOTES IN THE NEW MELODIC MOTIF WHICH HAS BEEN SUBSTITUTED, AND LOOKING UP RHYTHMIC
VOCABULARY FACTS IN THE DATABASE WHICH SUPPLY SUBSTITUTIONS GIVEN THE ORIGINAL
RHYTHMIC MOTIF AND THE NUMBER OF NOTES IN THE NEW MOTIF.
IF SEVERAL RHYTHMIC SUBSTITUTIONS APPLY, A RANDOM CHOICE IS MADE IN A WAY SIMILAR
TO THAT IN THE MELODIC VOCABULARY SELECTION. UNLIKE THE MELODIC APPROACH TO
ASSURING THAT SEQUENCES CARRY OVER INTO THE VARIATION, HOWEVER, IT SEEMS TO
SUFFICE HERE TO USE ONLY THE FREQUENCY CONSTRAINTS TO ASSURE THE RHYTHMIC
ASPECT OF THE SEQUENCE.
*****/

```

```

checkrhythframework([], []).

```

```

checkrhythframework([ [W,X,Y] | Ls ], [WZ,XZ,YZ | NewLs ]) :-
  changeseqrhythm([W,X,Y], WZ,XZ,YZ),
  checkrhythframework(Ls, NewLs).

```

```

checkrhythframework([ [W,X] | Ls ], [WZ,XZ | NewLs ]) :-
  changeseqrhythm([W,X], WZ,XZ),
  checkrhythframework(Ls, NewLs).

```

```

checkrhythframework([X | Ls ], [Z | NewLs ]) :-
  changerhythm(X, Z),
  checkrhythframework(Ls, NewLs).

```

```

checkrhythframework([ [X] | Ls ], [Z | NewLs ]) :-
  changerhythm(X, Z),
  checkrhythframework(Ls, NewLs).

```

```

changeseqrhythm([ [C1,R1,N1], [C2,R2,N2], [C3,R3,N3] ], [C1,NewR1,N1], [C2,NewR2,N2], [C3,NewR3,
  changerhythm([C1,R1,N1], [C1,NewR1,N1]),
  changerhythm([C2,R2,N2], [C2,NewR2,N2]),
  changerhythm([C3,R3,N3], [C3,NewR3,N3])).

```

```

changeseqrhythm([ [C1,R1,N1], [C2,R2,N2] ], [C1,NewR1,N1], [C2,NewR2,N2]) :-
  changerhythm([C1,R1,N1], [C1,NewR1,N1]),

```

```

changerhythm([C2,R2,N2],[C2,NewR2,N2]).

changerhythm([C,R,N],[C,NewR,N]) :-
    length(N, NumPitches),
    (rhythrndmchoice(0,R,2,NumPitches,NewR);
    rhythrndmchoice(0,R,1,NumPitches,NewR);
    rhythrndmchoice(0,R,0,NumPitches,NewR);
    rhythrndmchoice(0,R,3,NumPitches,NewR);
    rhythrndmchoice(0,R,_,NumPitches,NewR)).

changerhythm([C,R,N],[C,R,N]).

rhythrndmchoice(Total,Rhythm,Freq,NumPitches,NewR) :-
    ((Total = 2; not(freq(Rhythm,NumPitches,_,Freq))),
    retractall(alreadytried($)),!,fail);
    (
    random(RndmNum),
    getRhythID(RhythID,RndmNum),
    ((alreadytried(RhythID),
    !,rhythrndmchoice(Total,Rhythm,Freq,NumPitches,NewR));
    (
    (freq(Rhythm,NumPitches,RhythID,Freq),
    rhythvocab(Rhythm,NumPitches,RhythID,NewR),
    retractall(alreadytried($)),
    retract(freq(Rhythm,NumPitches,RhythID,Freq)),
    Freq1 is 1 + Freq,
    assert(freq(Rhythm,NumPitches,RhythID,Freq1))
    );
    (assert(alreadytried(RhythID)),
    Totall is 1 + Total,
    !,rhythrndmchoice(Totall,Rhythm,Freq,NumPitches,NewR)
    )
    )
    ).

getRhythID(1,Rndm) :-
    Rndm < 0.

getRhythID(2,_).

/*****
CHECKING ACCIDENTALS
*****/
/*****
STEP THROUGH THE VARIATION THAT HAS BEEN PRODUCED, CHECKING EACH NOTE TO
MAKE SURE THAT IT FITS WITHIN THE CHORD STRUCTURE THAT UNDERLIES IT. SINCE THE
NOTES ARE GROUPED BY THE MELODIC AND RHYTHMIC SUBSTITUTIONS THAT WERE MADE TO
THE ORIGINAL AND FRAMEWORK, CHECKING EACH NOTE CONSISTS OF CHECKING EACH NOTE
WITHIN EACH SUBLIST (GROUP) OF NOTES AGAINST THE CHORDS OF THAT GROUP.
THE DATABASE CONTAINS FACTS WHICH STATE WHICH ACCIDENTALS SHOULD BE CHANGED FOR
EACH CHORD. IF A MATCH (OF A "BAD" NOTE) IS FOUND, IT IS REPLACED IN THE LIST BY
THE GIVEN ACCIDENTAL (THE SAME PITCH NAME BUT WITH OR WITHOUT A SHARP OR FLAT),
OTHERWISE THE NOTE IS FINE AND IS LEFT AS IS.
*****/

checkaccidentals([],[]).

checkaccidentals([[Chords,Rhythm,Notes]|Rest],[[Chords,Rhythm,NewNotes]|NewLs]) :-
    checkeachsublist(Notes,Chords,NewNotes),
    checkaccidentals(Rest,NewLs).

```

```
checkeachsublist([],_,[]).
```

```
checkeachsublist([N|Rest],Chords,[NewN|NewNotes]) :-
    changeonenote(N,Chords,NewN),
    checkeachsublist(Rest,Chords,NewNotes).
```

```
changeonenote(N,[],N).
```

```
changeonenote(N,[C1|RestChords],NewN) :-
    changeaccidental(N,C1,NewN);
    changeonenote(N,RestChords,NewN).
```

```

/*****
                                DATABASE FACTS
*****/

```

```

/*****
MELODIC VOCABULARY FACTS MATCH A GIVEN INTERVAL OR "RUN" (FOR EXAMPLE A3 MEANS
ASCENDING THIRD) WITH A VALID SUBSTITUTION FOR THAT MELODIC INTERVAL/MOTIF,
ASSIGNING ID NUMBERS TO DISTINGUISH BETWEEN DIFFERENT POSSIBLE
SUBSTITUTIONS FOR THE SAME INTERVAL/MOTIF.
*****/

```

```
melvocab(am2,1,[A,B],[A,trill,B]) :-
    asc_min_second(A,B).
```

```
melvocab(a3,1,[A,B],[A,X,X,B]) :-
    asc_third(A,B),
    asc_fourth(A,X),
    desc_second(X,B).
```

```
melvocab(a3,2,[A,B],[A,X,B]) :-
    asc_third(A,B),
    asc_second(A,X),
    asc_second(X,B).
```

```
melvocab(a3,3,[A,B],[A,X,Y,B]) :-
    asc_third(A,B),
    asc_fourth(A,X),
    desc_fourth(A,Y),
    asc_sixth(Y,B).
```

```
melvocab(d3,1,[A,B],[A,X,B]) :-
    desc_third(A,B),
    desc_fourth(A,X),
    asc_second(X,B).
```

```
melvocab(d3,2,[A,B],[X,A,Y,B]) :-
    desc_third(A,B),
    asc_second(A,X),
    desc_second(A,Y),
    desc_second(Y,B).
```

```
melvocab(d3,3,[A,B],[A,B,X,B]) :-
    desc_third(A,B),
    desc_second(B,X).
```

```
melvocab(d3,4,[A,B],[A,A,X,B,Y,B]) :-
    desc_third(A,B),
    desc_second(A,X),
```

```

        desc_second(X,B),
        desc_second(B,Y).

melvocab(a4,1,[A,B],[A,X,Y,A,B]) :-
    asc_fourth(A,B),
    asc_third(A,X),
    desc_second(X,Y),
    desc_second(Y,A).

melvocab(a4,2,[A,B],[A,B,X,B,Y,B]) :-
    asc_fourth(A,B),
    asc_second(B,X),
    asc_min_second(Y,B).

melvocab(a4,3,[A,B],[A,X,Y,B]) :-
    asc_fourth(A,B),
    asc_octave(A,X),
    desc_third(X,Y),
    desc_third(Y,B).

melvocab(d4,1,[A,B],[X,Y,A,X,A,Z,B]) :-
    desc_fourth(A,B),
    asc_second(X,A),
    desc_second(Y,A),
    desc_second(Z,B).

melvocab(d4,2,[A,B],[A,X,A,B]) :-
    desc_fourth(A,B),
    desc_second(A,X).

melvocab(d4,3,[A,B],[A,B,X,B]) :-
    desc_fourth(A,B),
    asc_min_second(X,B).

melvocab(a5,1,[A,B],[A,X,Y,Z,B]) :-
    asc_fifth(A,B),
    asc_second(A,X),
    asc_second(X,Y),
    asc_second(Y,Z),
    asc_second(Z,B).

melvocab(a5,2,[A,B],[A,X,A,Y,B]) :-
    asc_fifth(A,B),
    desc_fourth(A,X),
    asc_third(A,Y),
    asc_third(Y,B).

melvocab(a5,3,[A,B],[A,X,Y,B]) :-
    asc_fifth(A,B),
    asc_third(A,X),
    asc_fourth(X,Y),
    desc_second(Y,B).

melvocab(d5,1,[A,B],[A,X,B,B,B]) :-
    desc_fifth(A,B),
    desc_third(A,X),
    desc_third(X,B).

melvocab(d5,2,[A,B],[A,X,B]) :-
    desc_fifth(A,B),
    desc_octave(A,X),
    asc_fourth(X,B).

melvocab(d5,3,[A,B],[A,X,Y,X,B,X,Y,B]) :-

```



```

desc_fifth(A,B),
desc_third(A,X),
desc_second(X,Y).

melvocab(a6,1,[A,B],[X,Y,A,B]) :-
    asc_sixth(A,B),
    desc_third(X,A),
    desc_second(Y,A).

melvocab(a6,2,[A,B],[A,X,Y,A,B]) :-
    asc_sixth(A,B),
    asc_third(A,X),
    desc_second(X,Y),
    desc_second(Y,A).

melvocab(d6,1,[A,B],[A,X,Y,B]) :-
    desc_sixth(A,B),
    desc_fourth(A,X),
    desc_second(X,Y),
    desc_second(Y,B).

melvocab(d6,2,[A,B],[A,W,X,Y,Z,Y,B]) :-
    desc_sixth(A,B),
    desc_second(A,W),
    desc_second(W,X),
    desc_second(X,Y),
    desc_second(Y,Z),
    desc_third(Y,B).

melvocab(a7,1,[A,B],[A,X,Y,B]) :-
    asc_seventh(A,B),
    asc_third(A,X),
    asc_third(X,Y),
    asc_third(Y,B).

melvocab(d7,1,[A,B],[A,X,A,B]) :-
    desc_seventh(A,B),
    desc_second(A,X).

melvocab(a8,1,[A,B],[A,X,Y,B]) :-
    asc_octave(A,B),
    asc_third(A,X),
    asc_third(X,Y),
    asc_fourth(Y,B).

melvocab(a8,2,[A,B],[X,A,Y,B]) :-
    asc_octave(A,B),
    desc_second(X,A),
    asc_second(Y,B).

melvocab(d8,1,[A,B],[A,X,Y,B]) :-
    desc_octave(A,B),
    desc_fourth(A,X),
    desc_third(X,Y),
    desc_third(Y,B).

melvocab(d8,2,[A,B],[A,X,A,B]) :-
    desc_octave(A,B),
    desc_second(A,X).

melvocab(dtriad,1,[A,B,C],[A,X,Y,A,B,Z,A,B,C]) :-
    desc_third(A,B),
    desc_third(B,C),
    asc_second(A,X),

```

```

        asc_second(X,Y),
        asc_second(B,Z).

melvocab(dtriad,2,[A,B,C],[A,A,X,Y,B,Y,Z,C]) :-
    desc_third(A,B),
    desc_third(B,C),
    asc_second(A,X),
    desc_second(A,Y),
    desc_second(Z,C).

melvocab(dtriad,3,[A,B,C],[X,A,X,B,Y,C]) :-
    desc_third(A,B),
    desc_third(B,C),
    asc_second(X,A),
    desc_second(B,Y).

melvocab(a5run,1,[A,B,C,D,E],[A,X,A,B,C,A,C,D,E]) :-
    asc_second(A,B),
    asc_second(B,C),
    asc_second(C,D),
    asc_second(D,E),
    desc_fourth(A,X).

melvocab(a5run,2,[A,B,C,D,E],[A,X,Y,Z,A,B,C,D,E]) :-
    asc_second(A,B),
    asc_second(B,C),
    asc_second(C,D),
    asc_second(D,E),
    desc_fourth(A,X),
    asc_second(X,Y),
    asc_second(Y,Z),
    asc_second(Z,A).

melvocab(rmfsdrun,1,[A,B,C,D,E],[A,X,A,B,C,B,C,D,E]) :-
    asc_second(A,B),
    asc_second(B,C),
    asc_second(C,D),
    desc_fifth(D,E),
    desc_fourth(A,X).

melvocab(rmfsdrun,2,[A,B,C,D,E],[A,X,Y,E,A,B,C,D,E]) :-
    asc_second(A,B),
    asc_second(B,C),
    asc_second(C,D),
    desc_fifth(D,E),
    desc_fourth(A,X),
    asc_second(X,Y),
    asc_second(Y,E).

melvocab(d5run,1,[A,B,C,D,E],[A,X,B,C,B,D,E]) :-
    desc_second(A,B),
    desc_second(B,C),
    desc_second(C,D),
    desc_second(D,E),
    asc_second(A,X).

melvocab(d5run,2,[A,B,C,D,E],[A,B,C,D,E,D,E]) :-
    desc_second(A,B),
    desc_second(B,C),
    desc_second(C,D),
    desc_second(D,E).

melvocab(d5run,3,[A,B,C,D,E],[A,C,B,A,E,D,E]) :-
    desc_second(A,B),

```

```

desc_second(B,C),
desc_second(C,D),
desc_second(D,E).

```

```

melvocab(smfrmrn,1,[A,B,C,D,B],[A,B,A,C,D,C,B]) :-
    desc_third(A,B),
    asc_second(B,C),
    desc_third(C,D),
    asc_second(D,B).

```

```

melvocab(smfrmrn,2,[A,B,C,D,B],[A,C,B,C,B,D,B]) :-
    desc_third(A,B),
    asc_second(B,C),
    desc_third(C,D),
    asc_second(D,B).

```

```

melvocab(_,_,X,X) :- fail.

```

```

/*****
RHYTHMIC VOCABULARY FACTS MATCH AN OLD GROUP OF RHYTHMS AND A NUMBER OF NOTES IN
THE NEW MELODY WITH A NEW RHYTHM FOR THAT MELODY.
*****/

```

```

rhythvocab([dh,dh],4,1,[q,q,q,dh]).
rhythvocab([dh,dh],3,1,[h,q,dh]).
rhythvocab([dh,h],7,1,[e,e,e,e,q,q,q]).
rhythvocab([dh,h],6,1,[e,e,e,e,q,h]).
rhythvocab([dh,h],4,1,[q,q,q,h]).
rhythvocab([dh,q],6,1,[e,e,e,e,q,q]).
rhythvocab([dh,q],5,1,[dq,e,e,e,q]).
rhythvocab([dh,q],4,1,[q,q,q,q]).
rhythvocab([dh,q],3,1,[h,q,q]).
rhythvocab([th,h],4,1,[tq,e,e,h]).
rhythvocab([th,q],5,1,[tq,ts,s,s,q]).
rhythvocab([th,q],4,1,[tq,e,e,q]).
rhythvocab([h,dh],4,1,[q,q,q,h]).
rhythvocab([h,dh],3,1,[q,q,dh]).
rhythvocab([h,h,q],9,1,[e,e,e,e,e,e,e,e,q]).
rhythvocab([h,h,q],8,1,[er,e,tt,t,t,e,e,q,q]).
rhythvocab([h,h,q],6,1,[q,e,e,q,q,q]).
rhythvocab([h,h],5,1,[t,t,t,q,h]).
rhythvocab([h,h],4,1,[q,q,q,q]).
rhythvocab([h,h],3,1,[q,q,h]).
rhythvocab([h,dq],4,1,[s,e,s,q,qr,ter]).
rhythvocab([h,q,q,dq,e],7,1,[q,q,q,e,e,dq,e]).
rhythvocab([h,q,q],9,1,[e,e,e,e,s,s,s,q]).
rhythvocab([h,q,q],8,1,[er,e,tt,t,t,t,t,q]).
rhythvocab([h,q,q],6,1,[q,q,e,e,e,e]).
rhythvocab([h,q],8,1,[s,s,s,s,s,s,e,q]).
rhythvocab([h,q],7,1,[th,th,th,th,dq,s,de]).
rhythvocab([h,q],6,1,[e,e,e,e,e,e]).
rhythvocab([h,q],6,2,[q,s,s,s,s,q]).
rhythvocab([h,q],5,1,[t,t,t,q,q]).
rhythvocab([h,q],4,1,[e,e,q,q]).
rhythvocab([h,q],3,1,[q,q,q]).
rhythvocab([tdq,e],4,1,[e,e,e,e]).
rhythvocab([dq,e],8,1,[th,th,th,th,q,th,th,s]).
rhythvocab([dq,e],7,1,[th,th,th,th,q,th,ds]).
rhythvocab([dq,e],5,1,[e,e,t,t,t]).
rhythvocab([dq,e],4,1,[e,e,e,e]).
rhythvocab([dq,e],3,1,[q,e,e]).
rhythvocab([tq,h],8,1,[s,s,s,s,s,s,e,q]).

```

```

rhythvocab([tq,h],5,1,[ts,s,s,s,h]).
rhythvocab([tq,h],3,1,[te,e,h]).
rhythvocab([tq,q],8,1,[s,s,s,s,s,s,s,s]).
rhythvocab([tq,q],6,1,[ts,s,s,s,e,e]).
rhythvocab([tq,q],5,1,[te,s,th,th,q]).
rhythvocab([tq,q],4,1,[e,e,e,e]).
rhythvocab([tq,q],3,1,[te,e,q]).
rhythvocab([tq,e],6,1,[e,th,th,th,th,e]).
rhythvocab([tq,e],5,1,[te,tth,th,th,th,e]).
rhythvocab([tq,e],4,1,[ts,s,e,e]).
rhythvocab([q,dh],5,1,[s,s,s,s,dh]).
rhythvocab([q,dh],4,1,[q,q,q,q]).
rhythvocab([q,h],6,1,[q,s,s,s,s,q]).
rhythvocab([q,h],5,1,[s,s,s,s,h]).
rhythvocab([q,h],4,1,[t,t,t,h]).
rhythvocab([q,h],3,1,[q,q,q]).
rhythvocab([q,dq],6,1,[q,s,s,s,s,e]).
rhythvocab([q,dq],5,1,[e,tth,th,th,th,dq]).
rhythvocab([q,dq],4,1,[s,e,s,dq]).
rhythvocab([q,q,h],9,1,[s,s,s,s,s,s,s,s,h]).
rhythvocab([q,q,h],8,1,[er,e,tt,t,t,t,t,q]).
rhythvocab([q,q,h],6,1,[e,e,s,e,s,h]).
rhythvocab([q,q,q],9,1,[s,s,s,s,s,s,s,s,q]).
rhythvocab([q,q,q],8,1,[er,e,tst,st,st,st,st,q]).
rhythvocab([q,q,q],6,1,[e,e,s,e,s,q]).
rhythvocab([q,q,e],9,1,[s,s,s,s,s,s,s,s,e]).
rhythvocab([q,q,e],8,1,[er,e,tst,st,st,st,st,e]).
rhythvocab([q,q,e],6,1,[e,e,s,e,s,e]).
rhythvocab([q,q],7,1,[s,s,s,s,s,s,e]).
rhythvocab([q,q],6,1,[s,s,s,s,e,e]).
rhythvocab([q,q],5,1,[e,s,th,th,q]).
rhythvocab([q,q],5,2,[s,s,s,s,q]).
rhythvocab([q,q],4,1,[t,t,t,q]).
rhythvocab([q,q],3,1,[e,e,q]).
rhythvocab([q,de],8,1,[s,s,s,s,s,s,th,th]).
rhythvocab([q,de],7,1,[s,s,s,s,s,s,s]).
rhythvocab([q,de],5,1,[e,e,s,s,s]).
rhythvocab([q,de],4,1,[e,s,s,de]).
rhythvocab([q,de],3,1,[e,e,de]).
rhythvocab([q,e,q],9,1,[s,s,s,s,s,s,s,e]).
rhythvocab([q,e,q],8,1,[s,s,s,s,s,s,e,e]).
rhythvocab([q,e,q],6,1,[e,e,s,s,e,e]).
rhythvocab([q,e,e,e,e],9,1,[q,s,s,s,s,s,s,s,s]).
rhythvocab([q,e,e,e,e],9,2,[q,ds,th,ds,th,ds,th,ds,th]).
rhythvocab([q,e,e],9,1,[s,s,s,s,s,s,th,th,s]).
rhythvocab([q,e,e],8,1,[s,ts,s,s,s,s,s,s]).
rhythvocab([q,e,e],6,1,[s,e,s,s,s,e]).
rhythvocab([q,e],8,1,[e,th,th,th,th,th,th,s]).
rhythvocab([q,e],6,1,[s,s,s,s,s,s]).
rhythvocab([q,e],5,1,[e,tth,th,th,th,e]).
rhythvocab([q,e],4,1,[s,s,e,e]).
rhythvocab([q,e],3,1,[e,e,e]).
rhythvocab([q,s,s],9,1,[st,st,st,st,st,st,th,th,s]).
rhythvocab([q,s,s],8,1,[s,s,s,s,th,th,th,th]).
rhythvocab([q,s,s],6,1,[s,e,s,th,th,s]).
rhythvocab([de,s,ds,th,e],7,1,[ds,th,ds,th,ds,th,e]).
rhythvocab([te,e,q],9,1,[tth,th,th,th,th,th,th,q]).
rhythvocab([te,e,q],8,1,[s,th,th,th,th,th,th,q]).
rhythvocab([te,e,q],6,1,[s,s,s,s,e,e]).
rhythvocab([te,e,e],9,1,[tth,th,th,th,th,th,th,th,e]).
rhythvocab([te,e,e],8,1,[ts,s,s,s,th,th,th,th]).
rhythvocab([te,e,e],6,1,[ts,s,s,s,s,s]).
rhythvocab([te,e,s,s,e],7,1,[ts,s,s,s,s,s,e]).
rhythvocab([te,e],8,1,[tth,th,th,th,th,th,th,th]).

```

rhythvocab([te,e],7,1,[tst,st,st,st,st,tht,tht]).  
 rhythvocab([te,e],6,1,[tth,th,th,th,s,s]).  
 rhythvocab([te,e],5,1,[tth,th,th,th,e]).  
 rhythvocab([te,e],4,1,[s,s,s,s]).  
 rhythvocab([te,e],3,1,[ts,s,e]).  
 rhythvocab([e,h],6,1,[e,s,s,s,s,q]).  
 rhythvocab([e,h],5,1,[e,s,s,e,q]).  
 rhythvocab([e,h],4,1,[e,e,e,q]).  
 rhythvocab([e,dq,e,e,q],9,1,[s,s,e,s,s,e,e,e,q]).  
 rhythvocab([e,dq],6,1,[e,s,s,s,s,e]).  
 rhythvocab([e,dq],5,1,[s,s,s,s,q]).  
 rhythvocab([e,dq],4,1,[e,s,s,q]).  
 rhythvocab([e,dq],3,1,[e,e,q]).  
 rhythvocab([e,q,e],9,1,[s,th,th,s,s,s,s,s,s]).  
 rhythvocab([e,q,e],8,1,[s,ts,s,s,s,s,s,s]).  
 rhythvocab([e,q,e],6,1,[st,st,st,de,s,e]).  
 rhythvocab([e,q],8,1,[th,th,th,th,s,s,s,s]).  
 rhythvocab([e,q],7,1,[e,tst,st,st,st,st,tht,tht]).  
 rhythvocab([e,q],6,1,[s,s,s,s,s,s]).  
 rhythvocab([e,q],5,1,[s,s,t,t,t]).  
 rhythvocab([e,q],4,1,[e,s,s,e]).  
 rhythvocab([e,q],3,1,[s,s,q]).  
 rhythvocab([e,de],6,1,[s,th,th,s,s,s]).  
 rhythvocab([e,de],4,1,[s,s,s,e]).  
 rhythvocab([e,de],3,1,[s,s,de]).  
 rhythvocab([e,e,q,de,s],7,1,[s,s,s,s,q,de,s]).  
 rhythvocab([e,e,q],9,1,[th,th,th,th,th,th,th,th,q]).  
 rhythvocab([e,e,q],8,1,[th,th,th,th,s,s,e,e]).  
 rhythvocab([e,e,q],6,1,[s,s,s,s,e,e]).  
 rhythvocab([e,e,de],9,1,[th,th,th,th,th,th,th,th,de]).  
 rhythvocab([e,e,de],8,1,[s,s,s,s,th,th,s,s]).  
 rhythvocab([e,e,de],6,1,[s,s,s,s,s,e]).  
 rhythvocab([e,e,e,h,dh],7,1,[e,e,e,e,e,q,dh]).  
 rhythvocab([e,e,e,de,s],9,1,[s,s,s,s,e,s,s,s,s]).  
 rhythvocab([e,e,e,e,q],9,1,[s,s,s,s,s,s,s,q]).  
 rhythvocab([e,e,e,e,q],7,1,[e,s,s,e,s,s,q]).  
 rhythvocab([e,e,e,ds,th],7,1,[s,s,s,s,e,ds,th]).  
 rhythvocab([e,e,e,s,s],7,1,[s,s,s,s,e,s,s]).  
 rhythvocab([e,e,e],9,1,[th,th,th,th,th,th,th,th,e]).  
 rhythvocab([e,e,e],8,1,[s,ts,s,s,th,th,th,th]).  
 rhythvocab([e,e,e],6,1,[s,s,s,s,s,s]).  
 rhythvocab([e,e],8,1,[th,th,th,th,th,th,th,th]).  
 rhythvocab([e,e],7,1,[st,st,st,st,st,tht,tht]).  
 rhythvocab([e,e],6,1,[th,th,th,th,s,s]).  
 rhythvocab([e,e],5,1,[th,th,th,th,e]).  
 rhythvocab([e,e],4,1,[s,s,s,s]).  
 rhythvocab([e,e],3,1,[s,s,e]).  
 rhythvocab([e,ds],6,1,[th,th,th,sf,sf,ds]).  
 rhythvocab([e,ds],4,1,[s,th,th,ds]).  
 rhythvocab([e,s],6,1,[th,th,th,th,th,th,th]).  
 rhythvocab([e,s],4,1,[s,th,th,s]).  
 rhythvocab([e,s],3,1,[s,s,s]).  
 rhythvocab([ts,s],6,1,[tsf,sf,sf,sf,th,th]).  
 rhythvocab([ts,s],4,1,[tth,th,th,th]).  
 rhythvocab([ts,s],3,1,[tth,th,s]).  
 rhythvocab([s,q],8,1,[sf,sf,sf,sf,s,s,s,s]).  
 rhythvocab([s,q],6,1,[sf,sf,sf,sf,e,e]).  
 rhythvocab([s,q],4,1,[th,th,e,e]).  
 rhythvocab([s,q],3,1,[s,e,e]).  
 rhythvocab([s,de],8,1,[sf,sf,sf,sf,s,th,th,s]).  
 rhythvocab([s,de],6,1,[sf,sf,sf,sf,s,e]).  
 rhythvocab([s,de],4,1,[th,th,s,e]).  
 rhythvocab([s,de],3,1,[s,s,e]).  
 rhythvocab([s,e],6,1,[sf,sf,sf,sf,s,s]).

```
rhythvocab([s,e],5,1,[th,th,st,st,st]).
rhythvocab([s,e],4,1,[s,th,th,s]).
rhythvocab([s,e],3,1,[s,s,s]).
rhythvocab([s,s,e,e,q],7,1,[th,th,s,s,s,e,q]).
rhythvocab([s,s],8,1,[sf,sf,sf,sf,sf,sf,sf,sf]).
rhythvocab([s,s],6,1,[sf,sf,sf,sf,th,th]).
rhythvocab([s,s],5,1,[th,th,tht,tht,tht]).
rhythvocab([s,s],4,1,[th,th,th,th]).
rhythvocab([s,s],3,1,[th,th,s]).
```

```
/******
MELODIC FREQUENCY FACTS LIST THE INTERVAL, MELODIC ID NUMBER FOR THAT INTERVAL,
AND FREQUENCY OF USE FOR THAT OPTION IN THE VARIATION TO DATE.
*****/
```

```
freq(am2,1,0).
freq(a3,1,0).
freq(a3,2,0).
freq(a3,3,0).
freq(a4,1,0).
freq(a4,2,0).
freq(a4,3,0).
freq(a5,1,0).
freq(a5,2,0).
freq(a5,3,0).
freq(a6,1,0).
freq(a6,2,0).
freq(a7,1,0).
freq(a8,1,0).
freq(a8,2,0).
freq(d3,1,0).
freq(d3,2,0).
freq(d3,3,0).
freq(d3,4,0).
freq(d4,1,0).
freq(d4,2,0).
freq(d4,3,0).
freq(d5,1,0).
freq(d5,2,0).
freq(d5,3,0).
freq(d6,1,0).
freq(d6,2,0).
freq(d7,1,0).
freq(d8,1,0).
freq(d8,2,0).
freq(dtriad,1,0).
freq(dtriad,2,0).
freq(dtriad,3,0).
freq(a5run,1,0).
freq(a5run,2,0).
freq(rmfdrun,1,0).
freq(rmfdrun,2,0).
freq(d5run,1,0).
freq(d5run,2,0).
freq(d5run,3,0).
freq(smfrmrn,1,0).
freq(smfrmrn,2,0).
```

```
/******
RHYTHMIC FREQUENCY FACTS LIST THE ORIGINAL RHYTHM, NUMBER OF NOTES IN THE NEW
MELODIC PATTERN FOR THAT SAME RHYTHMIC DURATION, RHYTHMIC ID NUMBER FOR THAT
```

OLD RHYTHM/NUMBER OF NOTES, AND FREQUENCY OF USE FOR THAT OPTION IN THE VARIATION TO DATE.

\*\*\*\*\*/

```

freq([dh,dh],4,1,0).
freq([dh,dh],3,1,0).
freq([dh,h],7,1,0).
freq([dh,h],6,1,0).
freq([dh,h],4,1,0).
freq([dh,q],6,1,0).
freq([dh,q],5,1,0).
freq([dh,q],4,1,0).
freq([dh,q],3,1,0).
freq([th,h],4,1,0).
freq([th,q],5,1,0).
freq([th,q],4,1,0).
freq([h,dh],4,1,0).
freq([h,dh],3,1,0).
freq([h,h,q],9,1,0).
freq([h,h,q],8,1,0).
freq([h,h,q],6,1,0).
freq([h,h],5,1,0).
freq([h,h],4,1,0).
freq([h,h],3,1,0).
freq([h,dq],4,1,0).
freq([h,q,q,dq,e],7,1,0).
freq([h,q,q],9,1,0).
freq([h,q,q],8,1,0).
freq([h,q,q],6,1,0).
freq([h,q],8,1,0).
freq([h,q],7,1,0).
freq([h,q],6,1,0).
freq([h,q],6,2,0).
freq([h,q],5,1,0).
freq([h,q],4,1,0).
freq([h,q],3,1,0).
freq([tdq,e],4,1,0).
freq([dq,e],8,1,0).
freq([dq,e],7,1,0).
freq([dq,e],5,1,0).
freq([dq,e],4,1,0).
freq([dq,e],3,1,0).
freq([tq,h],8,1,0).
freq([tq,h],5,1,0).
freq([tq,h],3,1,0).
freq([tq,q],8,1,0).
freq([tq,q],6,1,0).
freq([tq,q],5,1,0).
freq([tq,q],4,1,0).
freq([tq,q],3,1,0).
freq([tq,e],6,1,0).
freq([tq,e],5,1,0).
freq([tq,e],4,1,0).
freq([q,dh],5,1,0).
freq([q,dh],4,1,0).
freq([q,h],6,1,0).
freq([q,h],5,1,0).
freq([q,h],4,1,0).
freq([q,h],3,1,0).
freq([q,dq],6,1,0).
freq([q,dq],5,1,0).
freq([q,dq],4,1,0).
freq([q,q,h],9,1,0).
freq([q,q,h],8,1,0).

```

```

freq([q,q,h],6,1,0).
freq([q,q,q],9,1,0).
freq([q,q,q],8,1,0).
freq([q,q,q],6,1,0).
freq([q,q,e],9,1,0).
freq([q,q,e],8,1,0).
freq([q,q,e],6,1,0).
freq([q,q],7,1,0).
freq([q,q],6,1,0).
freq([q,q],5,1,0).
freq([q,q],5,2,0).
freq([q,q],4,1,0).
freq([q,q],3,1,0).
freq([q,de],8,1,0).
freq([q,de],7,1,0).
freq([q,de],5,1,0).
freq([q,de],4,1,0).
freq([q,de],3,1,0).
freq([q,e,q],9,1,0).
freq([q,e,q],8,1,0).
freq([q,e,q],6,1,0).
freq([q,e,e,e,e],9,1,0).
freq([q,e,e,e,e],9,2,0).
freq([q,e,e],9,1,0).
freq([q,e,e],8,1,0).
freq([q,e,e],6,1,0).
freq([q,e],8,1,0).
freq([q,e],6,1,0).
freq([q,e],5,1,0).
freq([q,e],4,1,0).
freq([q,e],3,1,0).
freq([q,s,s],9,1,0).
freq([q,s,s],8,1,0).
freq([q,s,s],6,1,0).
freq([de,s,ds,th,e],7,1,0).
freq([te,e,q],9,1,0).
freq([te,e,q],8,1,0).
freq([te,e,q],6,1,0).
freq([te,e,e],9,1,0).
freq([te,e,e],8,1,0).
freq([te,e,e],6,1,0).
freq([te,e,s,s,e],7,1,0).
freq([te,e],8,1,0).
freq([te,e],7,1,0).
freq([te,e],6,1,0).
freq([te,e],5,1,0).
freq([te,e],4,1,0).
freq([te,e],3,1,0).
freq([e,h],6,1,0).
freq([e,h],5,1,0).
freq([e,h],4,1,0).
freq([e,dq,e,e,q],9,1,0).
freq([e,dq],6,1,0).
freq([e,dq],5,1,0).
freq([e,dq],4,1,0).
freq([e,dq],3,1,0).
freq([e,q,e],9,1,0).
freq([e,q,e],8,1,0).
freq([e,q,e],6,1,0).
freq([e,q],8,1,0).
freq([e,q],7,1,0).
freq([e,q],6,1,0).
freq([e,q],5,1,0).
freq([e,q],4,1,0).

```



```

freq([e,q],3,1,0).
freq([e,de],6,1,0).
freq([e,de],4,1,0).
freq([e,de],3,1,0).
freq([e,e,q,de,s],7,1,0).
freq([e,e,q],9,1,0).
freq([e,e,q],8,1,0).
freq([e,e,q],6,1,0).
freq([e,e,de],9,1,0).
freq([e,e,de],8,1,0).
freq([e,e,de],6,1,0).
freq([e,e,e,h,dh],7,1,0).
freq([e,e,e,de,s],9,1,0).
freq([e,e,e,e,q],9,1,0).
freq([e,e,e,e,q],7,1,0).
freq([e,e,e,ds,th],7,1,0).
freq([e,e,e,s,s],7,1,0).
freq([e,e,e],9,1,0).
freq([e,e,e],8,1,0).
freq([e,e,e],6,1,0).
freq([e,e],8,1,0).
freq([e,e],7,1,0).
freq([e,e],6,1,0).
freq([e,e],5,1,0).
freq([e,e],4,1,0).
freq([e,e],3,1,0).
freq([e,ds],6,1,0).
freq([e,ds],4,1,0).
freq([e,s],6,1,0).
freq([e,s],4,1,0).
freq([e,s],3,1,0).
freq([ts,s],6,1,0).
freq([ts,s],4,1,0).
freq([ts,s],3,1,0).
freq([s,q],8,1,0).
freq([s,q],6,1,0).
freq([s,q],4,1,0).
freq([s,q],3,1,0).
freq([s,de],8,1,0).
freq([s,de],6,1,0).
freq([s,de],4,1,0).
freq([s,de],3,1,0).
freq([s,e],6,1,0).
freq([s,e],5,1,0).
freq([s,e],4,1,0).
freq([s,e],3,1,0).
freq([s,s,e,e,q],7,1,0).
freq([s,s],8,1,0).
freq([s,s],6,1,0).
freq([s,s],5,1,0).
freq([s,s],4,1,0).
freq([s,s],3,1,0).

```

```

/*****
CHANGEACCIDENTAL FACTS LIST THE ORIGINAL PITCH AND THE CHORD AGAINST WHICH THAT
PITCH IS BEING CHECKED, AND THE NEW PITCH TO SUBSTITUTE FOR THE ORIGINAL IN THAT
SITUATION.
*****/

```

```

changeaccidental(cs,c,c).
changeaccidental(gs,c,g).
changeaccidental(cs,c7,c).
changeaccidental(gs,c7,g).

```

changeaccidental (b,c7,bf) .

changeaccidental (ds,d,d) .  
changeaccidental (f,d,fs) .  
changeaccidental (as,d,a) .  
changeaccidental (ds,d7,d) .  
changeaccidental (f,d7,fs) .  
changeaccidental (as,d7,a) .  
changeaccidental (cs,d7,c) .  
changeaccidental (ds,dm,d) .  
changeaccidental (fs,dm,f) .  
changeaccidental (as,dm,a) .  
changeaccidental (ds,dm7,d) .  
changeaccidental (fs,dm7,f) .  
changeaccidental (as,dm7,a) .  
changeaccidental (cs,dm7,c) .

changeaccidental (d,dsdim,ds) .  
changeaccidental (f,dsdim,fs) .  
changeaccidental (as,dsdim,a) .

changeaccidental (g,e,gs) .  
changeaccidental (g,e7,gs) .  
changeaccidental (ds,e7,d) .  
changeaccidental (gs,em,g) .  
changeaccidental (gs,em7,g) .  
changeaccidental (ds,em7,d) .

changeaccidental (fs,f,f) .  
changeaccidental (as,f,a) .  
changeaccidental (cs,f,c) .  
changeaccidental (fs,f7,f) .  
changeaccidental (as,f7,a) .  
changeaccidental (cs,f7,c) .  
changeaccidental (e,f7,ef) .

changeaccidental (f,fsdim,fs) .  
changeaccidental (as,fsdim,a) .  
changeaccidental (cs,fsdim,c) .

changeaccidental (gs,g,g) .  
changeaccidental (ds,g,d) .  
changeaccidental (gs,g7,g) .  
changeaccidental (ds,g7,d) .  
changeaccidental (fs,g7,f) .

changeaccidental (g,gsdim,gs) .  
changeaccidental (ds,gsdim,d) .

changeaccidental (as,a,a) .  
changeaccidental (c,a,cs) .  
changeaccidental (as,a7,a) .  
changeaccidental (c,a7,cs) .  
changeaccidental (gs,a7,g) .  
changeaccidental (as,am,a) .  
changeaccidental (cs,am,c) .  
changeaccidental (as,am7,a) .  
changeaccidental (cs,am7,c) .  
changeaccidental (gs,am7,g) .

changeaccidental (d,b,ds) .  
changeaccidental (f,b,fs) .  
changeaccidental (d,b7,ds) .  
changeaccidental (f,b7,fs) .

changeaccidental (as,b7,a) .  
changeaccidental (ds,bm,d) .  
changeaccidental (f,bm,fs) .  
changeaccidental (ds,bm7,d) .  
changeaccidental (f,bm7,fs) .  
changeaccidental (as,bm7,a) .  
changeaccidental (ds,bdim,d) .  
changeaccidental (fs,bdim,f) .

changeaccidental (cs1,c,c1) .  
changeaccidental (gs1,c,g1) .  
changeaccidental (cs1,c7,c1) .  
changeaccidental (gs1,c7,g1) .  
changeaccidental (b1,c7,bf1) .

changeaccidental (ds1,d,d1) .  
changeaccidental (f1,d,fs1) .  
changeaccidental (as1,d,a1) .  
changeaccidental (ds1,d7,d1) .  
changeaccidental (f1,d7,fs1) .  
changeaccidental (as1,d7,a1) .  
changeaccidental (cs1,d7,c1) .  
changeaccidental (ds1,dm,d1) .  
changeaccidental (fs1,dm,f1) .  
changeaccidental (as1,dm,a1) .  
changeaccidental (ds1,dm7,d1) .  
changeaccidental (fs1,dm7,f1) .  
changeaccidental (as1,dm7,a1) .  
changeaccidental (cs1,dm7,c1) .

changeaccidental (d1,dsdim,ds1) .  
changeaccidental (f1,dsdim,fs1) .  
changeaccidental (as1,dsdim,a1) .

changeaccidental (g1,e,gs1) .  
changeaccidental (g1,e7,gs1) .  
changeaccidental (ds1,e7,d1) .  
changeaccidental (gs1,em,g1) .  
changeaccidental (gs1,em7,g1) .  
changeaccidental (ds1,em7,d1) .

changeaccidental (fs1,f,f1) .  
changeaccidental (as1,f,a1) .  
changeaccidental (cs1,f,c1) .  
changeaccidental (fs1,f7,f1) .  
changeaccidental (as1,f7,a1) .  
changeaccidental (cs1,f7,c1) .  
changeaccidental (e1,f7,ef1) .

changeaccidental (f1,fsdim,fs1) .  
changeaccidental (as1,fsdim,a1) .  
changeaccidental (cs1,fsdim,c1) .

changeaccidental (gs1,g,g1) .  
changeaccidental (ds1,g,d1) .  
changeaccidental (gs1,g7,g1) .  
changeaccidental (ds1,g7,d1) .  
changeaccidental (fs1,g7,f1) .

changeaccidental (g1,gsdim,gs1) .  
changeaccidental (ds1,gsdim,d1) .

changeaccidental (as1,a,a1) .  
changeaccidental (c1,a,cs1) .

changeaccidental (a1,a7,a1) .  
changeaccidental (c1,a7,cs1) .  
changeaccidental (gs1,a7,g1) .  
changeaccidental (a1,am,a1) .  
changeaccidental (cs1,am,c1) .  
changeaccidental (a1,am7,a1) .  
changeaccidental (cs1,am7,c1) .  
changeaccidental (gs1,am7,g1) .

changeaccidental (d1,b,ds1) .  
changeaccidental (f1,b,fs1) .  
changeaccidental (d1,b7,ds1) .  
changeaccidental (f1,b7,fs1) .  
changeaccidental (a1,b7,a1) .  
changeaccidental (ds1,bm,d1) .  
changeaccidental (f1,bm,fs1) .  
changeaccidental (ds1,bm7,d1) .  
changeaccidental (f1,bm7,fs1) .  
changeaccidental (a1,bm7,a1) .  
changeaccidental (ds1,bdim,d1) .  
changeaccidental (fs1,bdim,fl) .

changeaccidental (cs2,c,c2) .  
changeaccidental (cs2,c7,c2) .

changeaccidental (ds2,d,d2) .  
changeaccidental (f2,d,fs2) .  
changeaccidental (ds2,d7,d2) .  
changeaccidental (f2,d7,fs2) .  
changeaccidental (cs2,d7,c2) .  
changeaccidental (ds2,dm,d2) .  
changeaccidental (fs2,dm,f2) .  
changeaccidental (ds2,dm7,d2) .  
changeaccidental (fs2,dm7,f2) .  
changeaccidental (cs2,dm7,c2) .

changeaccidental (d2,dsdim,ds2) .  
changeaccidental (f2,dsdim,fs2) .

changeaccidental (g2,e,gs2) .  
changeaccidental (g2,e7,gs2) .  
changeaccidental (ds2,e7,d2) .  
changeaccidental (ds2,em7,d2) .

changeaccidental (fs2,f,f2) .  
changeaccidental (cs2,f,c2) .  
changeaccidental (fs2,f7,f2) .  
changeaccidental (cs2,f7,c2) .  
changeaccidental (e2,f7,ef2) .

changeaccidental (f2,fsdim,fs2) .  
changeaccidental (cs2,fsdim,c2) .

changeaccidental (ds2,g,d2) .  
changeaccidental (ds2,g7,d2) .  
changeaccidental (fs2,g7,f2) .

changeaccidental (g2,gsdim,gs2) .  
changeaccidental (ds2,gsdim,d2) .

changeaccidental (c2,a,cs2) .  
changeaccidental (c2,a7,cs2) .  
changeaccidental (cs2,am,c2) .  
changeaccidental (cs2,am7,c2) .

```

changeaccidental (d2,b,ds2) .
changeaccidental (f2,b,fs2) .
changeaccidental (d2,b7,ds2) .
changeaccidental (f2,b7,fs2) .
changeaccidental (ds2,bm,d2) .
changeaccidental (f2,bm,fs2) .
changeaccidental (ds2,bm7,d2) .
changeaccidental (f2,bm7,fs2) .
changeaccidental (ds2,bdim,d2) .
changeaccidental (fs2,bdim,f2) .

```

```

/*****
BASIC MUSICAL INTERVAL FACTS
*****/

```

```

asc_octave(X,Y) :-
    asc_fifth(X,Z), asc_fourth(Z,Y) .

asc_seventh(X,Y) :-
    asc_fifth(X,Z), asc_third(Z,Y) .

asc_sixth(X,Y) :-
    asc_third(X,Z), asc_fourth(Z,Y) .

asc_fifth(X,Y) :-
    asc_third(X,Z), asc_third(Z,Y) .

asc_fourth(X,Y) :-
    asc_third(X,Z), asc_second(Z,Y) .

asc_third(X,Y) :-
    asc_second(X,Z), asc_second(Z,Y) .

asc_second(c,d) .
asc_second(c,ds) .
asc_second(cs,d) .
asc_second(cs,ds) .
asc_second(d,e) .
asc_second(ds,e) .
asc_second(e,f) .
asc_second(e,fs) .
asc_second(f,g) .
asc_second(f,gs) .
asc_second(fs,g) .
asc_second(fs,gs) .
asc_second(g,a) .
asc_second(g,as) .
asc_second(gs,a) .
asc_second(gs,as) .
asc_second(a,b) .
asc_second(as,b) .
asc_second(b,c1) .
asc_second(b,cs1) .
asc_second(c1,d1) .
asc_second(c1,ds1) .
asc_second(cs1,d1) .
asc_second(cs1,ds1) .
asc_second(d1,e1) .
asc_second(ds1,e1) .
asc_second(e1,f1) .
asc_second(e1,fs1) .
asc_second(f1,g1) .
asc_second(f1,gs1) .

```



```
desc_third(X,Z), desc_third(Z,Y).

desc_fourth(X,Y) :-
    desc_second(X,Z), desc_third(Z,Y).

desc_third(X,Y) :-
    desc_second(X,Z), desc_second(Z,Y).

desc_second(g2,f2).
desc_second(g2,fs2).
desc_second(fs2,e2).
desc_second(f2,e2).
desc_second(e2,d2).
desc_second(e2,ds2).
desc_second(ds2,cs2).
desc_second(ds2,c2).
desc_second(d2,c2).
desc_second(d2,cs2).
desc_second(cs2,b1).
desc_second(c2,b1).
desc_second(b1,a1).
desc_second(b1,as1).
desc_second(as1,gs1).
desc_second(as1,g1).
desc_second(a1,g1).
desc_second(a1,gs1).
desc_second(gs1,fs1).
desc_second(gs1,f1).
desc_second(g1,f1).
desc_second(g1,fs1).
desc_second(fs1,e1).
desc_second(f1,e1).
desc_second(e1,d1).
desc_second(e1,ds1).
desc_second(ds1,cs1).
desc_second(ds1,c1).
desc_second(d1,c1).
desc_second(d1,cs1).
desc_second(cs1,b).
desc_second(c1,b).
desc_second(b,a).
desc_second(b,as).
desc_second(as,gs).
desc_second(as,g).
desc_second(a,g).
desc_second(a,gs).
desc_second(gs,fs).
desc_second(gs,f).
desc_second(g,f).
desc_second(g,fs).
desc_second(fs,e).
desc_second(f,e).
desc_second(e,d).
desc_second(e,ds).
desc_second(ds,cs).
desc_second(ds,c).
desc_second(d,c).
desc_second(d,cs).

unison(c,c).
unison(cs,cs).
unison(d,d).
unison(ds,ds).
unison(e,e).
```

```
unison(f, f) .
unison(fs, fs) .
unison(g, g) .
unison(gs, gs) .
unison(a, a) .
unison(as, as) .
unison(b, b) .
unison(c1, c1) .
unison(cs1, cs1) .
unison(d1, d1) .
unison(ds1, ds1) .
unison(e1, e1) .
unison(f1, f1) .
unison(fs1, fs1) .
unison(g1, g1) .
unison(gs1, gs1) .
unison(a1, a1) .
unison(as1, as1) .
unison(b1, b1) .
unison(c2, c2) .
unison(cs2, cs2) .
unison(d2, d2) .
unison(ds2, ds2) .
unison(e2, e2) .
unison(f2, f2) .
unison(fs2, fs2) .
unison(g2, g2) .
```



## 9. BIBLIOGRAPHY

- [Ab1] Abbot, C., "Programming Languages for Computer Music Synthesis, Performance, and Composition," Computing Surveys, vol. 17, no. 2, June 1985, pp. 235 - 265.
- [Ab2] Abbot, C., "Guest Editor's Introduction to the Special Issue on Computer Music," Computing Surveys, vol. 17, no. 2, June 1985, pp. 147 - 151.
- [Al] Alphonse, B.H., "Music Analysis by Computer - A Field for Theory Formation," Computer Music Journal, vol. 4, no. 2, Summer 1980, pp. 26 - 35.
- [Ap] Apel, W., Harvard Dictionary of Music, Belknap Press of Harvard University Press, 1972.
- [Ba1] Balaban, M., "Western Tonal Music A New Domain for AI Research," Technical Report 85-10, Department of Computer Science, State University of New York at Albany, 1986.
- [Ba2] Balaban, M., "Knowledge Representation and Inferencing in a Musical Database," Technical Report 85-11, Department of Computer Science, State University of New York at Albany, 1986.
- [Ba3] Balaban, M., "The Generalized Concept Formalism - A Frame and Logic Based Representation," Technical Report 85-20, Department of Computer Science, State University of New York at Albany, 1986.
- [Ba4] Balaban, M., "A Formal Basis for Research in Theories of Western Tonal Music," Technical Report 85-21, Department of Computer Science, State University of New York at Albany, 1986.
- [Ba5] Balaban, M., "Foundations for Artificial Intelligence Research of Western Tonal Music," Technical Report 85-22, Department of Computer Science, State University of New York at Albany, 1986.
- [Ba6] Balaban, M., "CSM: An AI Approach to the Study of Western

Tonal Music," Technical Report 85-24, Department of Computer Science, State University of New York at Albany, 1986.

- [BM] Balaban, M. and Murray, N., "Machine Tongues X: Prolog," Computer Music Journal, vol. 9, no. 3, Fall 1985, pp. 7 - 12.
- [BB] Baroni, M., Brunetti, R., Callegari, L. and Jacobini, C., "Relationships Between Melody and Harmony," International Conference on Musical Grammars and Computer Analysis, Modena, 1982.
- [[BJ1] Baroni, M. and Jacobini, C., "Analysis and Generation of Bach's Chorale Melodies," Proceedings of the First International Congress on the Semiotics of Music, Pesaro, Italy, 1975.
- [BJ2] Baroni, M. and Jacobini, C., Proposal for a Grammar of Melody: The Bach Chorales, Presses de l'Universite de Montreal, 1978.
- [BJ3] Baroni, M. and Jacobini, C., "Computer Generation of Melodies: Further Proposals," Computers and the Humanities, vol. 17, no. 1, March, 1983, pp. 1 - 18.
- [BO] Bernstein L. and Olive, J., "Computers and the 16th Century Chanson, A Pilot Project at the University of Chicago," Computers and the Humanities, vol. 3, no. 3, 1969, pp. 153 - 161.
- [Bi] Biles, J. A., Adaptive Recognition and Synthesis of Transcribed Jazz Solos By Computer, Master's thesis, University of Kansas, 19??.
- [BR] Boody, C. and Riedel, J., "A Computer Aided Study of Ecuadorean Urban Music," Computers and the Humanities, vol. 15, no. 2, August 1981, pp. 61 - 74.
- [BB] Brender, M. and Brender, R., "Computer Transcriptions and Analysis of Mid-Thirteenth Century Musical Notation," Journal of Music Theory, vol. 11, no. 2, 1967, pp. 198 - 221.

- [Br1] Brinkman, A. R., "Representing Musical Scores for Computer Analysis," Journal of Music Theory, vol. 30, no. 2, Fall 1986, pp. 225 - 275.
- [Br2] Brinkman, A. R., "A Binomial Representation of Pitch for Computer Processing of Musical Data," Music Theory Spectrum, vol. 8, 1986, pp. 44 - 57.
- [Br3] Brook, B., Musicology and the Computer, City University of New York Press, 1970.
- [BR] Buxton, W., Reeves, W., Baecker, R. and Mezel, L., "The Use of Hierarchy and Instance in a Data Structure for Computer Music," Computer Music Journal, vol. 2, no. 4, December 1978, pp. 10 - 20.
- [Ch] Child, P., "Fred Lerdahl and Ray Jackendoff: A Generative Theory of Tonal Music," review in Computer Music Journal, vol. 8, no. 4, Winter 1984, pp. 56 - 64.
- [Co1] Cope, D., "An Expert System for Computer-Assisted Music Composition," Computer Music Journal, vol. 11, no. 4, Winter 1987, pp. 30 - 46.
- [Co2] Cope, D., "Music & Lisp," AI Expert, March 1988, pp. 26 - 34.
- [CF] Crane, F. and Fiehler, J., "Numerical Methods of Comparing Musical Styles," in The Computer and Music, ed. H. Lincoln, Cornell University Press, 1970.
- [Eb1] Ebcioglu, K., An Expert System for Harmonization of Chorales in the Style of J. S. Bach, PhD dissertation, State University of New York at Buffalo, 1986.
- [Eb2] Ebcioglu, K., "An Expert System for Chorale Harmonization," Engineering Applications, pp. 784 - 788.
- [Er1] Erickson, R., "A General-Purpose System for Computer Aided Musical Studies," Journal of Music Theory, vol. 13, no. 2, 1969, pp. 276 - 294.
- [Er2] Erickson, R., "MUSICOMP 76 and the State of DARMS," College Music Symposium, vol. 17, no. 1, Spring 1977, pp. 90 - 101.

- [Fo] Forte, A., "Schenker's Conception of Musical Structure," in Readings in Schenker Analysis and Other Approaches, ed. M. Yeston, Yale University Press, 1977.
- [FR] Frankel, R., Rosenshein, S. and Smoliar, S., "A LISP-Based System for the Study of Schenkerian Analysis," Computers and the Humanities, vol. 10, no. 1, January/February 1976, pp. 21 - 32.
- [Fr1] Fry, C., "Computer Improvisation," Computer Music Journal, vol. 4, no. 3, Fall 1980, pp. 48 - 58.
- [Fr2] Fry, C., "Flavors Band: A Language for Specifying Musical Style," Computer Music Journal, vol. 8, no. 4, Winter 1984, pp. 20 - 34.
- [Go] Gourlay, J., "A Language for Music Printing," Communications of the ACM, vol. 29, no. 5, May 1986.
- [Gr] Gross, D., "Computer Applications to Music Theory: A Retrospective," Computer Music Journal, vol. 8, no. 4, Winter 1984, pp. 35 - 42.
- [HB] Harris, C. and Brinkman, A., "A Unified Set of Software Tools for Computer-Assisted Set-Theoretic and Serial Analysis of Contemporary Music," International Computer Music Conference 1986 Proceedings, Royal Conservatory, The Hague, Netherlands, pp. 331 - 336.
- [HS] Hewlett, W. and Selfridge-Field, E., Directory of Computer Assisted Research in Musicology, Center for Computer Assisted Research in the Humanities, 1986.
- [Hi] Hiller, L., "Music Composed with Computers - A Historical Survey," in The Computer and Music, ed. H. Lincoln, Cornell University Press, 1970.
- [HI] Hiller, L.A. and Isaacson, L.M., Experimental Music: Composition with an Electronic Computer, McGraw-Hill, 1959.
- [Ho1] Holland, S., "Speculations About How Artificial Intelligence

Could Be Applied to Aid to the Learning of Music Composition Skills," AISB Quarterly, no. 55-56, Autumn/Winter 1985.

- [Ho2] Holtzman, S. R., "Using Generative Grammars for Music Composition," Computer Music Journal, vol. 5, no. 1, Spring 1981, pp. 51 - 64.
- [JL] Jackendoff, R. and Lerdahl, F., "Discovery Procedures vs. Rules of a Musical Grammar in a Generative Music Theory," Perspectives of New Music, vol. 18, no. 1-2, 1980, pp. 503 - 510.
- [Jo] Jones, K., "Compositional Applications of Stochastic Processes," Computer Music Journal, vol. 5, no. 2, pp. 45 - 61.
- [La] Laske, O., "Keiith: A Rule System for Making Music-Analytical Discoveries," International Conference on Musical Grammars and Computer Analysis, Modena, 1972.
- [LJ1] Lerdahl, F. and Jackendoff, R., "Toward a Formal Theory of Tonal Music," Journal of Music Theory, vol. 21, no.1, 1977, pp. 111 - 172.
- [LJ2] Lerdahl, F. and Jackendoff, R., A Generative Theory of Tonal Music, MIT Press, 1983.
- [Le1] Levitt, D., A Melody Description System for Jazz Improvisation, Master's thesis, Massachusetts Institute of Technology, 1981.
- [Le2] Levitt, D., "Machine Tongues. X. Constraint Languages," Computer Music Journal, vol. 8, no. 1, Spring 1984, pp. 9 - 21.
- [LG] Lidov, D. and Gabura, J., "A Melody Writing Algorithm Using A Formal Language Model," Computer Studies in the Humanities, vol. 3, no. 4, 1973, pp. 138 - 148.
- [Li1] Lieberman, F., "Computer-Aided Analysis of Javanese Music," in The Computer and Music, ed. H. Lincoln, Cornell University Press, 1970.

- [Li2] Lieberman, H., "Machine Tongues. IX. Object-Oriented Programming," Computer Music Journal, vol. 6, no. 3, Fall 1982, pp. 8 - 21.
- [Li3] Lincoln, H., The Computer and Music, Cornell University Press, 1970.
- [Li4] Lincoln, H., "Uses of the Computer in Music Composition and Research," in Advances in Computers, vol. 12, ed. M. Rubinoff, Academic Press, 1972.
- [LA] Loy, G. and Abbott, C., "Programming Languages for Computer Music Synthesis, Performance, and Composition," Computing Surveys, vol. 17, no. 2, June 1985, pp. 235 - 265.
- [LS] Longuet-Higgins, H. and Steedman, M., "On Interpreting Bach," Machine Intelligence, vol. 6, 1971.
- [Me1] Meehan, J. R., "An Artificial Intelligence Approach to Tonal Music Theory," Computer Music Journal, vol. 4, no. 2, Summer 1980, pp. 60 - 65.
- [Me2] Merck, E., Extemporaneous Variation in the Baroque: A Study of Telemann's Methodial Sonatas, Master's thesis, Wesleyan University, 1976.
- [Mo] Moorer, J. A., "Music and Computer Composition," Communications of the ACM, vol. 15, no. 2, February 1972, pp. 104 - 113.
- [My] Myers, K. J., A System for the Generation of Tonal Music Based on Transformations, Master's Thesis, Rochester Institute of Technology, 1988.
- [NI] Nakamura, Y. and Inokuchi, S., "Music Information Processing System and Its Applications to Comparative Musicology," Proceedings of the Sixth International Joint Conference on Artificial Intelligence, Tokyo, 1979, pp. 633 - 635.
- [Na] Narmour, E., Beyond Schenkerism, University of Chicago Press, 1977.

- [Om] O'Maidin, D., "Computer Analysis of Irish and Scottish Jigs,"  
Musical Grammars and Computer Analysis,  
International Conference on Musical Grammars and  
Computer Analysis, Modena, 1982, pp. 329 - 336.
- [Pe1] Pelinski, R., "A Generative Grammar of Personal Eskimo  
Songs," International Conference on Musical Grammars  
and Computer Analysis, Modena, 1982, pp. 273- 286.
- [Pe2] Pennycook, B., "Computer-Music Interfaces: A Survey,"  
Computing Surveys, vol. 17, no. 2, June 1985, pp. 267 -  
289.
- [Po] Pope, S.T., "Building Smalltalk-80-based Computer Music  
Tools," Journal of Object-Oriented Programming,  
April/May 1988, pp. 6 11.
- [Qu] Quantz, J., On Playing the Flute, trans. by E. Reilly, Schirmer  
Books, 1966.
- [Ra1] Rader, G., "A Method for Composing Simple Traditional Music  
by Computer," Communications of the ACM, vol. 17,  
no. 11, 1974, pp. 631 - 642.
- [Ra2] Rahn, J., "On Some Computational Models of Music Theory,"  
Computer Music Journal, vol. 4, no. 2, Summer 1980,  
pp. 66 - 72.
- [Re] Reilly, E. R., Quantz and His Versuch, American Musicological  
Society, 1971.
- [Ro1] Roads, C., "Artificial Intelligence and Music," Computer Music  
Journal, vol. 4, no. 2, Summer 1980, pp. 13 - 25.
- [Ro2] Roads, C., "Grammars as Representations for Music," in  
Foundations of Computer Music, ed. J. Strawn, MIT  
Press, 1985.
- [Ro3] Roads, C., "Research in Music and Artificial Intelligence,"  
Computing Surveys, vol. 17, no. 2, June, 1985, pp. 163 -  
190.

- [Ro4] Roads, C., "An Overview of Music Representations," in Musical Grammars and Computer Analysis, ed. L. Callegari and M. Baroni, Societa Italiana de Musicologia, Firenze, Italy, 1986 (presented at International Conference on Musical Grammars and Computer Analysis, Modena, 1982.
- [Ro5] Rothgeb, J., "Simulating Musical Skills By Digital Computer," Computer Music Journal, vol. 4, no. 2, Summer 1980, pp. 36 - 40.
- [Sc] Schenker, H., Free Composition (Der Freie Satz), translated by E. Oster, Longman Press, 1979.
- [Sm1] Smoliar, S., "An Approach to Music Theory Through Computational Linguistics," Journal of Music Theory, vol. 20, 1976, pp. 105 - 131.
- [Sm2] Smoliar, S., "A Computer Aid for Schenkerian Analysis," Computer Music Journal, vol. 4, no. 2, Summer 1980, pp. 41 - 59.
- [St1] Stech, D., "A Computer-Assisted Approach to Micro-Analysis of Melodic Lines," Computers and the Humanities, vol. 15, no. 4, December 1981, pp. 211 - 221.
- [St2] Steinbeck, W., "The Use of the Computer in the Analysis of German Folksongs," Computers and the Humanities, vol. 10, no. 5, September/October 1976, pp. 287- 296.
- [Te] Telemann, G. P., Zwölf Methodische Sonaten, ed. by Max Seiffert, Basel: Barenreiter Kassel, 1955.
- [To] Tommasini, A., "A Computer Program that Really Knows the Score," The Boston Globe, August 28, 1983, p. A3.
- [Tr] Truax, B., "The POD System of Interactive Composition Programs," Computer Music Journal, vol. 1, no. 3, 1977, pp. 30- 39.
- [TI] Tsuboi, K. and Ishizuka, M., "Melody Analysis of Japanese Folk Songs Based on Production System," IEEE, 1984, pp. 714 - 716.



- [Ul] Ulrich, W., "The Analysis and Synthesis of Jazz by Computer,"  
Proceedings of the Fifth International Joint Conference  
on Artificial Intelligence, vol. 2, 1977, pp. 865 - 872.
- [Wi] Winograd, T., "Linguistics and the Computer Analysis of Tonal  
Harmony," Journal of Music Theory, vol. 12, no. 1,  
pp. 2 - 49.
- [Xe] Xenakis, I., Formalized Music, Indiana University Press, 1971.