September 1991

Computational Accounts of Music Understanding

Daniel Hardt
University of Pennsylvania

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-91-66.

Computational Accounts of Music Understanding

Abstract

We examine various computational accounts of aspects of music understanding. These accounts involve programs which can notate melodies based on pitch and duration information. It is argued that this task involves significant musical intelligence. In particular, it requires an understanding of basic metric and harmonic relations implicit in the melody. We deal only with single-voice, tonal melodies. While the task is a limited one, and the programs give only partial solutions to this task, we argue that this represents a first step towards a computational realization of significant aspects of musical intelligence.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-91-66.

Computational Accounts of Music Understanding

MS-CIS-91-66
LINC LAB-207

Daniel Hardt

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

September 1991

Computational Accounts of Music Understanding

Daniel Hardt

October 18, 1989

Abstract

We examine various computational accounts of aspects of music understanding. These accounts involve programs which can notate melodies based on pitch and duration information. It is argued that this task involves significant musical intelligence. In particular, it requires an understanding of basic metric and harmonic relations implicit in the melody. We deal only with single-voice, tonal melodies. While the task is a limited one, and the programs give only partial solutions to this task, we argue that this represents a first step towards a computational realization of significant aspects of musical intelligence.

Contents

1 Introduction

Can musical understanding be achieved by a computer? To address this question, it is necessary to define a clear test of understanding. The test we use here is that of taking musical dictation: a melody is played, and the listener must write it down in standard musical notation. A version of this "dictation problem" is easily applied to a computer: the melody is input as a list of pitches and durations, and we define a form of output which is closely related to standard musical notation. We argue that such a test requires significant musical understanding. In particular, it requires an understanding of the basic metric and harmonic relations implicit in the melody.

Metric relations have to do with grouping notes in time: in deciding if a melody is a waltz or a march, one must decide between groups of three and four units of time. This is an example of grouping notes into time-units which are called "bars". It is also necessary to make smaller groupings, that is, groupings within bars. For example, if one decides that each bar contains six notes of a given length, one must further decide whether these six notes are grouped in two groups of three or three groups of two. For our purposes, we define the metric problem as determining these metric groupings at least up to the level of the bar.

Harmonic relations also involve grouping notes: not in time, but with respect to a key. A listener determines that a melody begins in a certain key, and hears each pitch with respect to that key, until the key is felt to change. A given pitch is notated differently depending on the current key. So the harmonic problem can be described as follows: the key must be identified, and notes are notated according to that key. Also, the system must be alert for the possibility that the key will change.

The "dictation problem" represents a very limited notion of musical intelligence. Even so, the programs which attempt to solve it are subject to a number of further limitations. First, they deal only with unaccompanied melodies. Also, they maintain an artificial separation between metric and harmonic information: with a few exceptions, they avoid the use of metric information by the harmonic component, or the use of harmonic information by the metric component. Despite such limitations, we argue that these programs are a first step towards a computational realization of significant aspects of musical intelligence.

In what follows, we consider the metric problem and the harmonic problem separately. We begin with a description of the *metric hierarchy*, which embodies some important constraints on the possible solutions to the metric problem. Then we look at several programs which attempt to solve various aspects to the metric problem. In considering the harmonic problem we again begin with an abstract description of the solution space, which we term the *harmonic space*. Then we look at various partial solutions to this problem. Finally we draw conclusions about possibilities for integrating these programs and expanding upon them, and make some general remarks about the investigation of musical intelligence.

2 Meter

We examine the problem of determining the meter of a melody. First, we describe the *metric hierarchy*, which embodies some strong constraints on the possible solutions to the metric problem. These constraints are often taken for granted by musicians, but are rarely made explicit. We present several programs which attempt to determine the meter based solely on the relative duration of notes. The first of these is only able to move down the metric hierarchy, after being given the beat and measure length. The next program is able to move up the metric hierarchy, and thus complements the approach in the first program. We look at another program which also moves up the metric hierarchy based on relative durations. Finally we examine an approach which infers metric groupings based on facts about about melodic repetition.

2.1 The Metric Hierarchy

The programs we will examine do not give output precisely in the form of music notation; instead, a hierarchy of metric groupings is given. This hierarchy can be expressed as a tree (Fig 2.1), where at each lower level the grouping is in shorter time-units. Defining the meter in this way is not entirely equivalent to standard musical notation for meter. Standard notation expresses no grouping larger than the measure, despite the fact that such groupings are often clearly definable. In this sense the hierarchical representation can contain more information. On the other hand, standard notation selects one level of metric grouping as the measure and another as the beat, while the hierarchical representation does not make any such distinctions among the levels. In this sense, it is the standard notation which contains more information.

It is assumed that the ratio of the metric units at adjacent levels has only two and three as prime factors. This is described as a "reasonable" ratio for a metric grouping. In general, when a possible metric grouping is being considered, it will be rejected if it is not "reasonable" in relation to the groupings already established. This assumption seems to be justified for baroque music; it may be necessary to relax this assumption for later music such as Brahms, where factors such as five and seven seem to appear in the metric hierarchy.

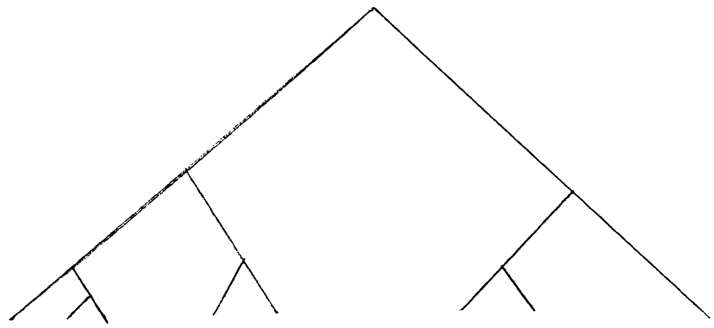
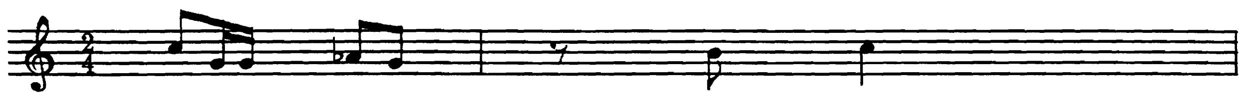


Fig. 2.1 The Metric Hierarchy



2.2 Relative Durations

2.2.1 Moving Down the Metric Hierarchy

We examine a program (LH76) that is able to infer levels of metric grouping below the beat level, based on relative durations of notes. The program requires that a bar's worth of beats be given before the melody; it is not able to identify the beats from the melody itself, as a musician can. On the other hand, it has the ability to tolerate slight variations in the durations of notes, as is inevitable in human performance.

The program begins by making the beat length the current metrical unit. Then, whenever the current metrical unit is interrupted by the onset of a note, the current metrical unit is subdivided into n units, where n is either 2 or 3, depending on whether the interruption was one/half or one/third of the way through the current metrical unit. In this way we build all the levels of the metric hierarchy which are below the beat level.

The system contains a constant called the "tolerance", which is the maximum temporal discrepancy it will disregard. For example, a given note would be considered an interruption of the current metrical unit only if its onset differed from the expected time by an amount greater than the tolerance. The tolerance is necessary because the system is dealing with "real" durations; the system must ignore the minor temporal discrepancies which are inevitable in a human performance. At the same time, the system can make use of such discrepancies to detect a gradual change in tempo. This would not be possible using "idealized" note lengths, as is done in MS73.

2.2.2 Moving up the Metric Hierarchy: Longuet-Higgins

Here we look at a program (LH82) which is meant to complement the program described above (LH 76). That program was only able to move down the metric hierarchy, and needed to be given the beat and measure length. This program addresses the problem of how to "get started", i.e., how to find the beat and other higher levels in the hierarchy. After identifying a metric grouping from the opening notes of a melody, it attempts to move up the metric hierarchy, at least up to the measure level. In principle it should be able to handle "real" note durations, as was possible in the 76 program.

The program begins by examining $t1$, the onset time of the first note, and $t2$, the onset time of the second note. The length l of note 1 is $t2 - t1$. The system predicts that the third note will occur at time $t3 = t2 + l$. If it does, the metrical hypothesis is confirmed. Then the routine CONFLATE is called, which moves $t2$ onto $t3$, thus doubling the current metric unit. If a note occurs between $t2$ and $t3$ which is longer than the note on $t2$, $t2$ is moved onto this note. This is the purpose of the STRETCH routine, which thus lengthens the current estimate of the beat length.

There are cases where the first note is an upbeat, where we want to move $t1$ to the second note. This is the purpose of the UPDATE routine: if we are near the beginning of a sequence and we encounter a note which is longer than any of its predecessors, we move $t1$ to this note. The UPDATE rule is, whenever the current note at $t2$ is longer the note at $t1$, move $t1$ to the current note, *except* if the long note at $t2$ is shorter than the current value of the beat. This allows the UPDATE rule to apply in Example A, but not in Example B (Fig 2.2).

The rules of the program are admirably simple–in some ways too simple. The program makes very quick decisions about the phase and period of metric units, and there is very little provision for undoing these decisions.

The period of the current metric unit is determined by the interaction between the CONFLATE and STRETCH rules. The CONFLATE rule assumes a binary grouping unless it is contradicted by the STRETCH rule. This causes the program to frequently mistake triplet groupings for binary groupings. For example, as long as notes of equal value are seen, CONFLATE continually builts up metric groupings which are twice the size of the previous unit. This would fail with an opening triplet passage, such as Fugue 4, book II of the WTC (Fig 2.3).

There are also important problems with determining the phase of metric units. Metric units at all levels begin at $t1$, and the only rule which can move $t1$ is the UPDATE rule. The UPDATE rule is very narrowly defined, however, and will in many cases fail to move $t1$ when required. For example, in the case of initial rests, higher level metric units will almost certainly be out of phase, since $t1$ can never be moved to the left, as it should be to handle initial rests. For example, in Fugue 1, Book I (Fig 2.4), the program would place $t1$ under the first note, and build up metric groupings from there. The metric units larger than eighth notes will be out of phase.

The program needs some ways other than UPDATE in which the phase of metric groupings can be changed. It is reasonable to assume that $t1$ marks the beginning of a metric group at *some* level: but except in the case of an upbeat, the program assumes that all higher levels of metric groupings will also begin there. Even in the case of an upbeat, UPDATE has problems. It would fail to notice an upbeat in a case such as the Beethoven Violin Sonata no 1, (Fig 2.5) where the upbeat is the same length as the downbeat. In this case, the phase of higher level metric groupings becomes obvious by the second measure, but by this time it is not possible for the UPDATE rule to apply. Also, the UPDATE rule causes the program to often identify an upbeat where one is not found, such as in Fig 2.6.

Thus this program is complementary to the work in LH76 in that it is able to find the beat and other higher levels in the metric hierarchy, while the earlier work was only able to move down the metric hierarchy after being given the beat. Both programs share the goal of handling "real" note durations, at least in principle. While the previous program used a constant "tolerance" to handle small variations in duration, there is no discussion of how such a mechanism would be incorporated into the present rules.

Fig. 2.2

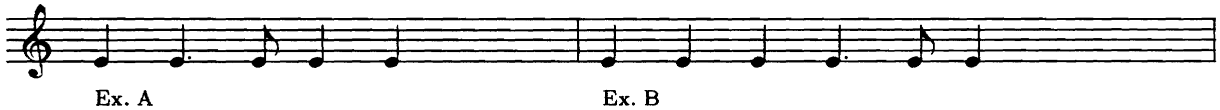


Fig. 2.3 WTC, Fugue 4, Book II

Fig. 2.4 WTC, Fugue I, Book I

Fig. 2.5 Beethoven Sonata for Violin

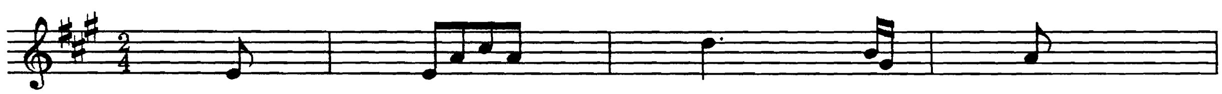
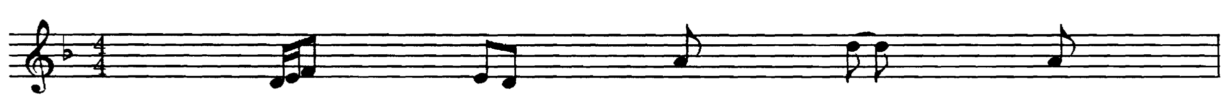


Fig. 2.6 Bach Harpsichord Concerto



2.2.3 Moving up the Metric Hierarchy: Steedman

We examine another program (MS73) which attempts to move up the metric hierarchy. This program infers metric groupings by comparing relative durations, as did the previous program. In addition, it defines some structured rhythmic events which it takes as evidence of metric groupings. As mentioned above, this program expects a melody whose notes have "idealized" durations, ie, in the exact ratios dictated by the meter. It has no "tolerance" for small variations in note-length, as had the program of LH76.

The program works as follows: the first note is taken to define the current metric unit, typically at a rather low level in the metric hierarchy. There are several rules for enlarging this metric unit and thus moving up in the rhythmic hierarchy. There are no rules for moving down the metric hierarchy, although it is not clear why the rules given could not serve this purpose as well. As in all the rhythmic algorithms we examine here, levels of the rhythmic hierarchy must be related by prime factors of 2 or 3, and no others.

The simplest evidence for moving up the metric hierarchy is a longer note; if its length is a "reasonable" multiple of the current metric unit, it is adopted as the new metric unit. In addition to such facts, Steedman defines two more structured rhythmic events which indicate a new metric grouping: the "dactyl" and the "isolated accent".

The dactyl is defined as follows: "the first three notes in a sequence of four, such that the second and third are equal in length and shorter than the first or the fourth." (p 39, MS73) If a dactyl is encountered, it defines a new metric unit, given that it is a reasonable multiple of the current metric unit. In Fugue 2 book 1, (Fig 2.7) Notes 5-7 comprise a dactyl which lasts one quarter note. Since the current metric unit is an eighth note, the new current metric unit becomes a quarter note.

An "isolated accent" is an isolated metrical unit which is "marked for accent", where "A metrical unit is marked for accent if a note or dactyl begins at the beginning of it and lasts throughout it" (p 43 MS73). Otherwise a metrical unit is "unmarked". Two marked units separated by several unmarked units is taken as evidence for a metric grouping, whose length is the period between the two marked units.

This allows, for example, the grouping of three eight notes in Fugue 15, Book I. (Fig 2.8) As with all the rules, the IAR will only establish a new metric grouping if it is a "reasonable" multiple of the current metric unit.

The rules can now be precisely stated as follows: (p 42, MS73)

1. Dactyl Rule: If the note is the first note of a dactyl, and the total length of the dactyl has a "reasonable" ratio to the current metric unit, this becomes the current metric unit. If the ratio is not a "reasonable" one, but the length of the first note minus the combined length of the next two does satisfy the "reasonableness" condition, then this length is taken to be the new current metric unit.

2. Dotted Note Rule: The note is followed by a single shorter note, followed by a longer note. take the length of the short note and subtract it from the first note. If this length is a "reasonable" one, it becomes the new current metric unit.

3. Long Note Rule: If the note length itself has a "reasonable" ratio to the current metric unit, it becomes the new cmu.

4. Isolated Accent Rule: If the note is marked for accent, or begins a dactyl which is marked for accent, and it is followed by several unmarked notes followed by another note/dactyl which is marked for accent, the period from the current note to the next marked note defines a metric grouping, if it is a reasonable multiple of the current metric unit.

5. If none of these rules apply, the current metric unit is retained.

Rules 2 and 3 are similar to the rules of the Longuet-Higgins algorithm considered above (LH82): they allow a step up in the metric hierarchy based on extremely simple facts about relative durations. As in the previous algorithm, the rules seem too quick to decide that a new metric grouping has been found. By rule 3, whenever we encounter a note which is twice the current metric unit, we define a new metric unit of that length. Although Rules 1 and 2 are meant to apply before Rule 3, this still causes the program to make simple mistakes in moving up in the metric hierarchy. For example, in the Minuet from Eine Kleine Nachtmusik (Fig 2.9), when the half note is encountered, Rule 3 is applied, causing the program to incorrectly infer a binary grouping of quarter notes.

The dactyl and isolated accent rules are different in that they define more complicated rhythmic structures which are taken to be evidence of metric groupings. The dactyl rule allows the program to make some musically plausible inferences of metric groupings, such as in the Bach Fugue Book 1 no 2 (Fig 2.7). However, it is not clear why the dactyl should be so narrowly defined. Why, for example, must it be four notes? A grouping of three or five notes, where the first and last are long, and the intervening notes of short, equal values, would seem equally plausible candidates for establishing metric groupings. The isolated accent rule seems open to the same type of questions that we raise below with respect to melodic repetition: while such events clearly seem to be evidence of salient musical groupings, it is not always clear that such groupings should be identified with metric groupings. In particular, the phase of such groupings might differ from that of the metric groupings.

In the next section we look at a program which infers metric groupings based on melodic repetition. Like the dactyl and isolated accent rules described above, this is a structured event which is taken as evidence for metric groupings. It is the first method we have seen which introduces information other than relative durations.

Fig. 2.7 WTC, Fugue 2, Book I

Fig. 2.8 WTC, Fugue 15, Book I

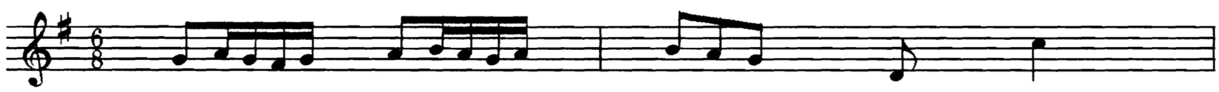
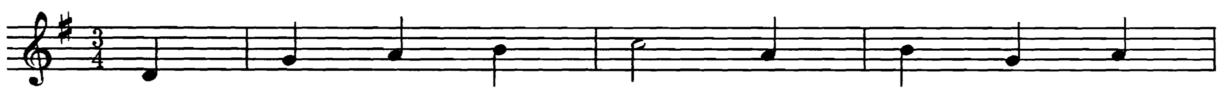


Fig. 2.9 Menuet, Eine Kleine Nactmusik



2.3 Melodic Repetition

Steedman (MS73, MS77) argues that information about the meter can be inferred from the repetition of a figure: a new level of metric group can be obtained as the duration between the start of a figure and its repetition. The repetition of a figure is defined in terms of the durations and melodic intervals between notes. For example in Fugue 4 book II (Fig 2.3), the first three notes are a figure that is repeated by notes 4-6. This repetition indicates a grouping of three notes. Notes 13-15 repeat the figure as well. This indicates in turn a higher level of grouping, of groups of four triplets. A simple repetition of a figure is defined as follows (MS77, p 560):

> A pair of sequences of three or more successive notes of a melody constitutes a simple repetition if all the notes before the last one are equal in duration, and if the corresponding intervals between the notes in the two sequences involve the same number of steps in the scale identified by the key signature, in the same direction.

An *interval* between two notes is defined as the number of scale steps between them. Thus intervals can only be determined once the key is known. The metric grouping to be inferred from a repetition is one which begins on the first note of the figure and lasts until the first note of the repeat. This grouping is accepted subject to the usual restrictions imposed by the metric hierarchy, i.e., the grouping must begin where the current metric group begins, and it must be a reasonable multiple of the current metric group. One other important restriction is that a repetition must be taken to involve the *earliest possible* figure: otherwise, in Fugue 4 book II (Fig 2.3), notes 13-15 might be taken as a repetition of notes 4-6, giving an absurd metric grouping.

The notion of "variant" repetition is introduced to account for repetitions that otherwise might give a grouping whose phase is incorrect. For example, in Fugue 20, book I (fig 2.10), if one infers a metric unit based on the simple repetition as defined above, the metric unit inferred would be from note 3 to note 8, which is incorrect. What is needed is a way of counting the figure as beginning at the beginning of the piece, and the repetition beginning with note 5. This is achieved by defining a notion of a"variant" repetition (p 563):

> Two sequences of notes in a melody constitute a variant repetition if the corresponding notes (except possibly the last) are

equal in duration, and the corresponding intervals are the same in direction, and the sequences end with a simple repetition.

The notion of variant repetition allows the program to move the metric grouping to the left, when the grouping might otherwise have incorrect phase. A variant repetition can be divided into two parts. The second part must be a simple repetition. The first part must match the first part of the figure, but the standard for matching is less strict. Intervals whose directions are the same but whose size are different are allowed to match, and intervals involving rests match any interval.

Scales and alternations must be dealt with as special cases. A scale would contain repetitions of many periods, so inferences based on a scale must be restricted as follows: any repetition which begins in the middle of a scale is ignored, unless the repetition continues after the scale passage. Alternations also must be dealt with in a special way. An alternation is a sequence where every other note is the same. There must be at least three successive occurrences of the repeating separated by notes all of which are either higher, or lower in pitch. Fugue 15, book II contains an alternation sequence (Fig 2.11), while Fugue 18 book II (Fig 2.12) does not. Although the underlined section includes three successive A's, the intervening notes are not all either above or below, as required. Alternation sequences are taken to indicate a binary grouping of notes, with the repeated note unaccented.

Some notion of figure and repetition is doubtless an essential part of an account of musical intelligence, and the current program represents an important attempt to incorporate such a notion. Indeed, this notion should be expanded in various ways. The criteria of what counts as repetition ought to be relaxed a bit. For example, in Fig 2.13 the repetition of the notes 1-5 and notes 6-10 is ruled out, since the first interval is a 4th in the figure, and a 3rd in the repetition. Also, while there is a very constrained notion of "variant" repetition, we need a broader notion of repetitions which resemble the figure in a salient way. For example, there is no concept of a question and answer, although this identifies groupings just as repetition does.

While it is definitely worthwhile to incorporate figure and repetition in an account of musical intelligence, the inference of metric grouping on this basis is somewhat problematic. Melodic repetition does seem to imply a grouping of some sort, but it is not always clear that a metric grouping is implied. They are groupings which are *related* to metric groupings but should not necessarily be identified with them. (See Lerdahl+Jackendoff, p 27) The relationship of these groupings to metric units needs further investigation.

It may be that the period of groupings generally coincides with that of metric groupings, while the phase does not.

Fig. 2.10 WTC, Fugue 20, Book I

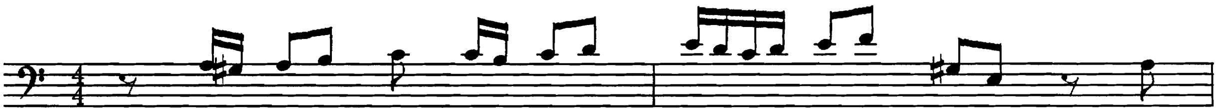


Fig. 2.11 WTC, Fugue 15, Book II

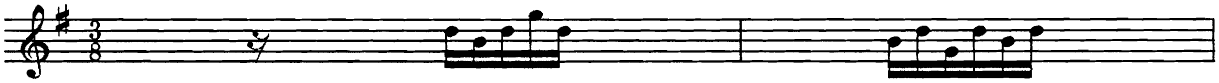


Fig. 2.12 WTC, Fugue 18, Book II

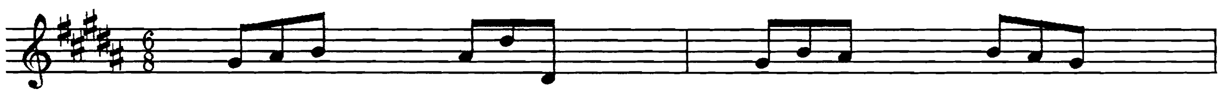


Fig. 2.13 Beethoven Sonata op 2 no 2 Scherzo

2.4 Idealized vs. Real Durations

As mentioned above, the Steedman programs expect "idealized" durations, so that only ratios which are exactly divisible by 2 or 3 will allow the inference of metric groupings. In real human performance such an expectation will rarely be satisfied. This is why Longuet-Higgins introduces the notion of a "tolerance", which allows the system to ignore discrepancies smaller than a certain value. Also, this makes it possible to detect a change in tempo based on such minor discrepancies, without problems arising with the metric groupings.

While the Steedman programs lack these virtues, it seems that the durations could be "normalized" in some fashion prior to the application of the current rules of the program: for example, all durations could be rounded to the nearest small ratio with respect to the first note, such that the ratio is divisible by 2 or 3.

It seems reasonable to abstract away from such issues in attempting to frame principles that are musically and cognitively interesting. Such idealization is common in other areas of AI, such as computational linguistics, where the input to syntactic and semantic rules is idealized in several ways. Indeed, we suspect that it is impossible to arrive at a viable theory of musical understanding unless one can abstract away from such low level issues.

3 Harmony

The harmonic problem is to determine the note names for a melody. To do this, it is necessary to determine the key of the melody, since a given tone is notated differently depending on the key. For example, in the key of F major, a certain tone would be notated B♭, while in F♯ major that same tone would be A♯. We begin with an abstract description of the *harmonic space*, formalizing the notion of a key and other important concepts. Next, we look at a very simple proposal for determining the key: the tonic- dominant rule. Then we look at an alternative to this simple rule. This alternative defines a syntax of melody, and determines the key based on the harmonic implications of the syntactic constituents. Finally we look at the problem of detecting a change in key.

3.1 The Harmonic Space

Longuet-Higgins (LH62a, LH62b) has given a formal theory of the *harmonic space* which serves as a basis for the harmonic algorithms we examine. A musical interval can be thought of as a ratio between two frequencies. An octave, for example, has a ratio of 2/1. A perfect fifth has the ratio of 3/2, and a major third has the ratio of 5/4. However, not all frequency ratios define musical intervals. For example, the ratio 7/1 is not one that appears in Western music. Indeed, the only frequency ratios which appear in Western music are those which are expressible as the product of the three prime factors 2,3, and 5. In other words, all musical intervals can be thought of as being composed of octaves, fifths, and major thirds.

To reflect this, Longuet-Higgins pictures the harmonic space as a three-dimensional array, where the x axis denotes intervals of perfect fifths, the y axis, major thirds, and the z axis, octaves. A note at point (x,y,z) is a distance of x fifths, y major thirds, and z octaves from the origin, which is arbitrarily defined as middle C. This harmonic space is pictured in Fig 3.1, with the z dimension left out. The box corresponds to the *extended key* of C; any other extended key can be similarly determined by drawing such a box with the key note occupying the position of C in the diagram. The extended key determines the correct note-name for any keyboard position. Thus if the key is known, the notation for any keyboard position can be found in the harmonic space. The *key frame* is the seven notes within the extended key which make up the scale of that key. The other five notes within the frame we term *accidentals*.

	−3	−2	−1	0	1	2	3	4
−3	*D*♯	*A*♯	*E*♯	*B*♯	*F*×	*C*χ	*G*×	*D*χ
−2	*B*	*F*♯	*C*♯	*G*♯	*D*♯	*A*♯	*E*♯	*B*♯
−1	*G*	*D*	*A*	*E*	*B*	*F*♯	*C*♯	*G*♯
0	*E*♭	*B*♭	*F*	*C*	*G*	*D*	*A*	*E*
1	*C*♭	*G*♭	*D*♭	*A*♭	*E*♭	*B*♭	*F*	*C*
2	*A*♭♭	*E*♭♭	*B*♭♭	*F*♭	*C*♭	*G*♭	*D*♭	*A*♭

Fig. 3.1 The Harmonic Space

By defining the harmonic space in this way, we are able to give precise definitions for some important musical concepts. First we can precisely quantify the *remoteness* of an interval. To do so, we define the *sharpness* of a note as the quantity $q = x + 4y$, where x,y are coordinates in the harmonic space. The remoteness of an interval then is defined as the difference in sharpness of the two notes of the interval. A further distinction needed is the traditional distinction between *chromatic* and *diatonic* intervals. We define diatonic intervals as those with remoteness less than 6, and chromatic intervals as those with remoteness greater than 6.

3.2 The Tonic-Dominant Rule

Longuet-Higgins (LH71,76) offers a simple solution to the problem of determining the initial key: it is based on the assumption that a melody will begin on either the tonic or the dominant, most likely the tonic. The rule is that one begins by assuming that the first note is the tonic. If the second note is to the right of the first or directly over it (in the harmonic space), the hypothesis is supported, otherwise, the first note is taken to be the dominant. This rule holds quite well for most Bach fugue subjects. However, there are many unremarkable melodies which start on a note other than the tonic or dominant, such as "I Dream of Jeannie with the Light Brown Hair" (Fig 3.2). Furthermore, even if the first note *is* either the tonic or dominant, the second note does not always allow one to determine which it is.

3.3 An Alternative to the Tonic-Dominant Rule

Because of the obvious inadequacy of the tonic-dominant rule, Steedman (MS73) proposes an alternative approach to key determination. The approach begins by allowing many different key hypotheses, and then successively rules out hypotheses that conflict with the harmonic implications of the melody. These implications are defined, not in terms of individual notes, but in terms of "syntactic constituents" of the melody. These constituents in turn are defined based on individual notes as notated according to a given key hypotheses. Thus the overall structure of the algorithm is as follows:

```
Initialize list of key hypotheses
For each note in melody
 For each key hypothesis
   notate current note
   determine syntactic constituent
   identify harmonic implications, if any
  eliminate any hypotheses that conflict with implications
```

The criteria for eliminating a key hypothesis are quite strict, so we find that nearly all key hypotheses are generally eliminated after the first few notes. When only one hypothesis remains, it is taken to be the key of the melody. In what follows we explain this algorithm in detail. First, we describe how a melody is notated according to a given key hypothesis. Next, we define the syntactic constituents and give the harmonic implications which are defined in terms of these constituents. Finally, we give the criteria for eliminating a given key hypothesis.

3.4 Notating the Melody

Since we are merely notating based on a given key hypothesis, the rules are rather simple. Whenever the notes fall within the scale of the key, the notation is obvious. We require special rules for accidentals and for semitone sequences. Often accidentals serve to signal a modulation, and in such cases these rules will fail, since the rules ignore the possibility of modulation.

The first rule is: if a note can be notated within the frame of the seven scale degrees, that is the notation adopted. (Note: for minor scales, we follow the melodic convention, ie, in ascending and descending minor scales we use the major sixth and minor seventh).

We define a special rule to handle sequences of semitones:

Semitone rule: Every note in a chromatic scale, except the first and last, must be involved in at least one diatonic semitone.

Finally, accidentals which are not part of semitone sequences are dealt with in the following rule.

Rule for accidentals: place it as close as possible to the previous note. The definition of *closeness* is that given above in the section on the harmonic space: it is the difference in *sharpness* of two notes. Except for one interval, a closest interpretation can always be found for an interval. There is one interval for which this rule does not give a unique answer: that of 6 semitones. It could be either the augmented fourth or the diminished fifth. Steedman gives no method for making a choice in this case.

The notation algorithm as a whole is as follows:

1. If interval is a semitone, notate according to semitone rule.

2. Otherwise, if interval is non-accidental, notate according to key frame.

3. Else (interval *is* accidental), choose interpretation closest to predecessor.

Fig. 3.2 I Dream of Jeannie

Fig. 3.3 WTC, Fugue 7, Book I

3.5 Defining a Syntax of Melody

Harmonic understanding, like rhythmic understanding, requires grouping notes into larger units. We describe a "syntax of melody" which defines musically salient constituents, such as turns, inflections, and runs. This allows the system to recognize to some intuitively obvious musical facts, that it otherwise would miss. For example, the subject of the E♭ major fugue (Fig 3.3) opens with the E♭ major triad; this is clearer if one thinks of the G-F-G as an "inflection" of the G.

Constituents are defined in terms of intervals, which can be determined based on the notation of the melody according to a given key hypothesis. There are two categories of constituents. The first, which concerns single notes or elaborations of a single note, we term *points.* The second concerns movements between points; these are called *transitions.*

Of points we define four kinds:

1. Note: a single note.
2. Repetition: a note that is repeated one or more times.
3. Inflection: a sequence of three intervals such that the first is an descending second, the second is an ascending second, and both have the same duration.
4. Turn: A sequence of four intervals such that the first two are descending seconds, the third is an ascending second, and all three have the same duration.

There are two kinds of transitions:

1. Run: a sequence of 2 or more seconds, all in the same direction
2. Jump: any transition which is not involved in a run.

There is some possible overlap among the definitions of run, turn and inflection. The run always takes precedence over the turn,i.e., a turn will not be found if part of it can be considered part of a run. In all cases except one, the run also takes precedence over the inflection. The exception, for a case like no 4 book II, (Fig 2.3) states that if an inflection is succeeded but not preceded by an ascending second, it is not to be considered a run but as an inflection. Finally, if there is ambiguity between a turn and an inflection, a turn is found.

3.6 Harmonic Implication

We define rules for determination of harmonic implication in terms of the syntactic constituents defined above. In particular, we look at the intervals of melodic transitions, that is, jumps and runs. The harmony is defined by its *triad*, which is made up of thirds and a perfect 5th. Thus the first rule is that

> "the thirds and the perfect fifth, (and their inversions), imply the two triads apiece which they take part in." (p 173, MS73)

This leaves seconds and sevenths, which are not in the triad. They are interpreted as implying thirds or fifths, which in turn imply triads. These implication of seconds and sevenths are found by examining their path through the harmonic space. For example the major second is two steps to the right in the harmonic space; it is taken to imply one step to the right: the perfect fifth. Thus the major second implies the triads the perfect fifth implies: the major and minor triads. Similarly, the dominant seventh, which is two steps to the left, implies the perfect fourth, one step to the left.

The minor second, two steps left and one step up, implies the perfect fourth, one step to the left. Its inverse, the minor seventh, implies a perfect fifth leading to the second note. The minor seventh can also imply two other intervals: the perfect fifth from the first note, and the perfect fifth from the note a third below the first note. The semitone is even more ambiguous; its implications can be similarly determined from the path in the harmonic space. Often the implied interval will have a different end- point than the actual interval. Such an implied end-point of a transition is called a *virtual note*: in this case the next transition is taken as beginning on the virtual end-point of its predecessor.

Harmonic implications are in general defined in terms of transitions. Apart from one exception, no "internal" implications are considered, i.e., the implications of the notes making up a point or transition are not considered. The exception is that we consider the harmonic implications of the first interval of a run. The motivation this is to handle examples such as Fugue 1, book 1 (Fig 2.4) If one merely took the implication of the run as a whole, the perfect 4th C - F would only allow the two triads containing F and C, ruling out the correct key. The first interval C - D implies the key of C major, allowing it to eventually be selected. It isn't clear that the first few notes of the C major subject *should* lead to an irrevocable conclusion that the key is C major. The same rule would cause an example such as Fig 3.4

to be incorrectly labeled as G major. This exception for the first interval of a run seems rather ad hoc. In principle it would seem that *all* internal implications should be considered, granting them perhaps less weight than "external" implications.

We have seen how a transition can be seen as implying a given triad. A transition which implies the triad of a given key is said to be *congruent* to that key. The notion of congruence to a key is an essential one for key determination.

3.7 Key Determination

We present the key determination algorithm(p 214, MS73):

1. Introduce all key hypotheses for which the 1st note is non-accidental.

2. For each transition, eliminate any hypotheses for which that transition is non-congruent. A transition is congruent to a key iff it implies the triad of that key.

3. If all hypotheses are non-congruent, retain those for which the current transition is non-accidental.

4. If the current transition is accidental for all hypotheses, retain them all.

Thus the current key hypotheses are examined with respect to the current transition based on two criteria: whether the interval is *accidental* and whether the interval is *congruent*. If no hypothesis matches the stricter test of congruence, then the weaker test of looking for accidentals is used. Normally, the first interval will rule out all but a very small number of key hypotheses.

This algorithm gives a much more sophisticated approach to key determination than the approach based on the tonic-dominant rule. Rather than considering just two key hypotheses, it gives consideration to many possible keys until it finds evidence to rule them out, based on the harmonic implications of the melody. These implications are defined in terms of syntactic melodic constituents. The notion of syntactic constituents, in addition to improving the performance of the present algorithm, is suggestive of ways in which the current approach could be extended to model some aspects of higher-level musical understanding.

Fig. 3.4

Fig. 3.5 Paganini Violin Concerto in D

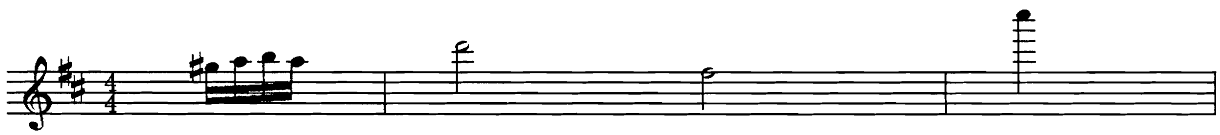


Fig. 3.6 WTC Fugue 5, Book II

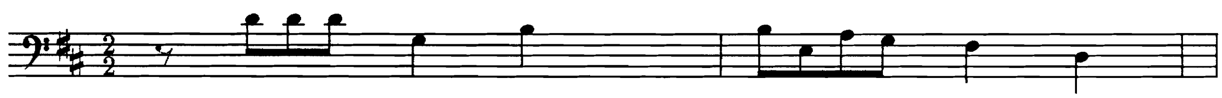
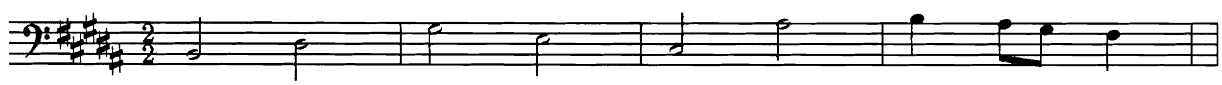


Fig. 3.7 WTC Fugue 23, Book II



3.8 Problems with the Key Determination Algorithm

Despite such virtues, this approach has some serious shortcomings. We mention first a rather minor problem relating to the initial list of key hypotheses. Then we discuss the more fundamental problem that the algorithm lacks any notion of harmonic context or progression.

The first rule eliminates any key for which the first note is accidental. This loses the benefit of having defined syntactic constituents, and will cause an error when the first note is part of a "point" such that, by itself, the first note is accidental to the key, while the point is non-accidental. For example, a melody which begins with a turn might easily begin with a note which is accidental. (See Fig 3.5.) This problem would be rectified if one merely began with *all* possible key hypotheses.

Steedman mentions two examples which illustrate a basic problem with this algorithm: it lacks any notion of harmonic context or progression. It always assumes that the first triad established is that of the tonic. Also, if a note *can* be interpreted in the current harmony then it *must* be. Thus in Fig 3.6 the first triad established is the subdominant, but the algorithm mistakenly interprets it as the tonic. This despite the fact that in the context of the two measures, the harmony is clearly IV-V-I. Similarly in Fig 3.7 the algorithm mistakenly finds the key to be G ♯ minor, because the first three notes are that triad. Again, from the context, the key of B major is obvious. What is required is some notion of harmonic progressions such as IV-V-I, and some notion of harmonic context.

3.9 Detecting a Change in Key

Steedman's syntax-based approach ignores the issue of detecting a change in key. It merely assumes that the first tonality detected is the key of the melody, and notates the entire melody according to this. Longuet-Higgins gives some simple rules for this problem.

The assumption is that, if the present key "requires the notes to jump about too violently in harmonic space..." the system "selects a new key in which the offending intervals give place to less remote ones". (LH79, p 319) We give the following rules about a possible change in key:

1. If L,M, and N are three successive notes, and intervals LM and MN are both chromatic, then M is assigned a new interpretation M' which makes both intervals diatonic, and the key changes so as to take in M'.

2. If K,L,M, and N are four successive notes, and the three intervals LM,KM and LN are all chromatic, the M is again reinterpreted, and the key is changed accordingly.

3. If N is a note a the major key and MN is a rising chromatic semitone, then M is reinterpreted so as to make M'N a diatonic semitone, but no change of key is precipitated.

Longuet-Higgins does not give a precise method for determining the new interpretation for a given note; nor does he describe how to select the new key to accommodate the reinterpreted note. Presumably one could identify, in each case, the choice which is nearest in the harmonic space.

4 Conclusions

4.1 Integrating the Programs

The programs we have presented could be integrated, removing some rather artificial boundaries between different parts of the dictation problem. For example, we have generally maintained a strict separation between the metric and harmonic components of the dictation problem. Each component would be improved if it had access to information from the other. We have seen only one example of this: the inference of metric groupings based on melodic repetition (MS73). Harmonic information could be used much more extensively as evidence for metric groupings. In general, "stronger" beats tend to coincide with more harmonic stability, where stability is defined in terms of proximity to the tonic. This correspondence could be reflected by rules in various ways. One simple example involves finding upbeats, which is necessary for determining the phase of metric groupings. An upbeat is rarely more stable harmonically than the downbeat. So a metric algorithm could require that it would only find an upbeat if it is less stable harmonically than the downbeat. Metric information could be used by the harmonic components in similar ways, by looking for a correspondence between stronger beats and more harmonic stability.

Another boundary involves moving up and down the metric hierarchy. We have only seen programs that can do one or the other. What is needed is a program that does both. Many of the rules which were described in the context of finding a larger metric grouping could be applied equally well to finding lower level groupings.

4.2 Extending the Programs

Various ways of extending these programs suggest themselves. One important way, which has been suggested by Steedman (MS73), involves expanding the harmonic context of key determination. Steedman's key determination program really just finds the first chord that is implied. The next step is to define a context such as a phrase, and describe the chord sequences that can make up a phrase. Once the phrase level is defined, it would be possible to look at higher levels of harmonic organzation, which might look very much like the phrase level. Thus the extremely local harmonic key-determination of MS73 could be retained as the lowest level in an expanded harmonic program. Phrase structure rules for chord sequences have been used by Steedman (MS84) for jazz sequences. Also, Winograd (68) and Lerhdahl and Jackendoff have suggested similar approaches.

It may also be possible to expand on the metric programs we have seen. None define methods for detecting a change in meter. Also, none of the programs give a method for determining which level of the metric hierarchy is the beat, and which is the measure. It can be argued that this is arbitrary. However, it appears that the metric hierarchy does not extend indefinitely upward. LH82 conjectured that one can only go a level or two above the measure level. Beyond that, metric groupings seems to become rather different, if they can indeed be called metric groupings at all. If this is true, it appears one does need to make some distinctions in levels in the metric hierarchy, at least to find the upper limit. Perhaps this upper limit also gives a basis for finding the measure and beat levels.

4.3 Investigating Musical Intelligence

The dictation problem has been useful for our purposes because it is testable, and because it requires building some of the basic structure required for music understanding. Although only the trained musician can actually perform such a task, it seems reasonable to assume that the inference of basic facts about key and meter underly a more universal musical understanding. Still, if the ultimate goal is a computational model of musical intelligence, it will be necessary to move on to other tasks.

The ability to notate a heard melody is at best a side-effect of musical understanding; one can certainly understand music without being able to perform this task. Nor is the ability to perform this task a sure sign of musical understanding. One could imagine a computer that could solve the "dictation problem" but that didn't really understand much about music. What is needed is a test which more directly reflects musical understanding. Such a test might be to recognize various structural facts about a piece: finding phrases, or recognizing a sonata form and its structural elements, for example. Another possibility might be recognizing relations between a theme and variations on that theme; the work cited above (MS73,MS77) on melodic repetition took some initial steps in that direction. Further progress towards a computational account of music understanding will require devising tests such as these.

References

1. Cooper, G. and Meyer, L.B. (1960) *The Rhythmic Structure of Music.* Chicago: University of Chicago Press.

2. Lerhdahl, and Jackendoff, R. *A Generative Approach to Tonal Music*, MIT Press.

3. Longuet-Higgins, H.C. (1962a) "Letter to a Musical Friend" Music Review 23, August, 244.

4. Longuet-Higgins, H.C. (1962b) "Second Letter to a Musical Friend" Music Review 23, November, 280.

5. Longuet-Higgins, H.C. (1976) "Perception of Melodies", Nature, Vol. 263, October 21, 1976.

6. Longuet-Higgins, H.C. (1979) "The Perception of Music", Proc. Roy. Soc. London B 205, 307-322.

7. Longuet-Higgins, H.C. and Steedman, M.J. (1971) "On Interpreting Bach", Machine Intelligence 6 (ed. Meltzer and Michie) Edinburgh: Edinburgh University Press.

8. Longuet-Higgins, H.C. and C.S. Lee (1982) "The Perception of Musical Rhythms", Perception, 1982.

9. Steedman, M.J. (1973) "The Formal Description of Musical Perception", Ph.D. thesis, 1973.

10. Steedman, M.J. (1977) "The Perception of Musical Rhythm and Meter", Perception, 1977.

11. Steedman, M.J. (1984) "A Generative Grammar for Jazz Chord Sequences", Music Perception, Fall, Vol. 2, No. 1, 52-77.

12. Winograd, T. (1968) "Linguistics and the Computer Analysis of Harmony" Journal of Music Theory, 12, 2.