Computers and the Humanities **35:** 37–54, 2001.

Representing Melodic Patterns as Networks of Elaborations

ALAN MARSDEN
Music Department, Lancaster University, Lancaster LA1 4YW, UK
(E-mail: A.Marsden@lancaster.ac.uk)

Abstract. Previous discussions of musical pattern have underlined difficulties in seeking pattern as a sequence of pitches, or of intervals or of other local and atomic features. This paper describes a manner of representing melodies through a hierarchical structure of elaboration, derived from concepts common in music theory (in particular, the concept of reduction found in the work of Schenker and of Lerdahl & Jackendoff). The fundamental structure is a planar directed acyclic graph, each node of which represents a musical note (not necessarily as it is present in the actual melody) and an elaboration which generates that note on the basis of two parents. These graph structures can be converted to trees, aiding processing and comparison, in two ways. Firstly, any graph can be transformed into a set of binary trees in which each node represents an interval between two notes and an elaboration of that interval. Secondly, in the planar graph, the link of a node to one of its parents often provides no useful information and can be disregarded, resulting in a reduction of the graph tending towards a set of trees. From this arises a new approach to the question of melodic segmentation. Examples of melodic fragments represented in this manner demonstrate how the representation makes explicit similarities between fragments which would not be found by an approach using sequences of features.

1. Background

Discussions of pattern often point out the difficulty of associating patterns with sequences of data referring to the sequences of notes in melodies. In some cases, matching data arises only when the pitches of notes are represented in a scale of seven pitches to the octave, and in some cases a scale of twelve pitches to the octave is required. Furthermore, some matches require the representation of intervals between pitches rather than the pitches themselves, and rhythmic factors such as duration and accent are often important also. The issue is aired in West, Howell and Cross (1991, pp. 5–7), and Cambouropoulos proposes a solution involving a sophisticated representation of intervals and multi-parametric matching (Cambouropoulos, 1996, pp. 246–248; 1998a, pp. 120–123). Selfridge-Field discusses the problem extensively but inconclusively (1998), and offers the “theme” of the second movement of Mozart’s piano sonata in D, K. 311, as an example of a case when a clearly audible musical pattern is manifested slightly differently at every occurrence. She suggests that recognition of the pattern among these different occurrences requires their reduction to a common “model” which underlies them all. The manner of representation described here allows such recognitions precisely because the reduction is inherent in each representation, and so matches can be discovered in the underlying layers of a network of elaborations even when the surface features of the music differ.

The idea of finding some kind of hierarchical generative structure in a melody is common in music theory. Heinrich Schenker’s highly influential theory has its basis in the idea that underlying any musical structure is a simpler one of which it is an elaboration (Schenker, 1935). In the case of melody, the processes of elaboration are precisely the processes used by performers for ornamentation. Lerdahl and Jackendoff (1983) attempted a partial formalisation of Schenker’s ideas, parsing musical structures into strict trees. A tree structure is also found in the representation scheme proposed by Deutsch and Feroe (1981) as a model for the cognitive representation of melodies. (See Deutsch, 1999, pp. 366–369 for a brief description of the scheme.) It is effectively present too in the various generative grammars for music (e.g. Baroni et al., 1992; Kippen and Bel, 1992; see Sundberg and Lindblom, 1991, for a survey).

While a number of features are held in common between these various approaches, notably the relation to procedures of ornamentation, there are important differences in the kind of hierarchical structure produced and the nature of information represented in nodes and links. Lerdahl and Jackendoff produce strict binary trees, with each node as a note (effectively). However, their links do not contain the information required to reconstruct the melody (i.e. the sequence of leaf nodes) on the basis of the head node and the structure of links. Nor does their theory fully specify a mechanism for deriving the tree structure from the melody. The scheme of Deutsch and Feroe produces trees with arbitrarily large branching (though commonly of an order of about five). The nodes are notes once again, and the links do carry sufficient information to allow reconstruction of the melody. However, their scheme allows an enormous number of different representations of the same melody, and no mechanism is suggested to derive a preferred representation. Baroni’s grammar allows reconstruction of melodies, as one might expect, but its nodes are effectively not simply notes. The production rules generally take a pair of notes on the left-hand side and produce on the right-hand side a sequence of three or more notes beginning and ending with the notes of the left-hand side. Thus if the generative structure is seen as a tree, its nodes must be regarded as *intervals between notes* rather than as notes themselves (see below). Schenker’s theory, because of its less formal nature, is not susceptible to such simple categorisation, but it is clear that elaborations are sometimes applied not to an individual note but are dependent on a pair of successive notes. His graphs are perhaps most easily read as directed acyclic graphs with the links only implicitly represented from level to level.

The system of representation described here is not unlike Baroni’s grammar in that elaborations are regarded as essentially dependent on a *pair* of notes.

However, it contains a set of elaborations which is fuller and richer in one sense, but more restricted in the sense that the elaborations are more tightly defined and the branching is always binary.

2. E-Graphs

The essential principle of the representation is to represent a melody as the product of successive elaborations of a simple outline. The outline is a sequence of notes and/or rests. In the most richly represented cases, it is just one or two notes (perhaps preceded and/or followed by rests as temporal "markers").

Each note has a pitch and a time. A note does not have a specified duration – this property emerges from the time of a note and the time of the following note. (An alternative formulation in which notes do have duration would be possible, but it appears to have no advantages.) Furthermore, other, contextual, information must be held for notes, giving the prevailing key, harmony and metre. Since these are not usually regarded as properties of notes, and since duration, normally regarded as a property of a note, is not taken as a primary property here, the term *place* will be used instead of note. This also allows for future extension of the representation to deal with fuller musical textures where a place does not correspond to a single note. For the purposes of this formulation, rests are defined to be notes with a silent pitch.

2.1. GRAPH STRUCTURE

The basic representation is called an E-graph, and consists of places, elaborations and links. The top level consists of a sequence of places (minimum two). Every other place is generated by an elaboration which applies to a pair of places, and generates that new place intermediate in time between the two parent places (except in the case of *accented elaborations*, discussed below). Thus a representation can be regarded as a directed acyclic graph which is planar in the sense that no links cross. The graph in Figure 1, for example, would represent the melody consisting of the notes represented at Place1, Place4, Place3, Place5, and Place2, in that order.

The representation could be converted to a tree by taking each node to represent the interval between two notes (or two places) rather than a single note, resulting in the representation shown in Figure 2. (It is important to note that, in this case, the intervals are not just "sizes" such as "perfect fourth", but intervals between a specific pair of pitches.) Although a tree structure is simpler than its equivalent graph, complications introduced by accented elaborations (see below) cause the graph to be more easily handled despite its more complex structure.

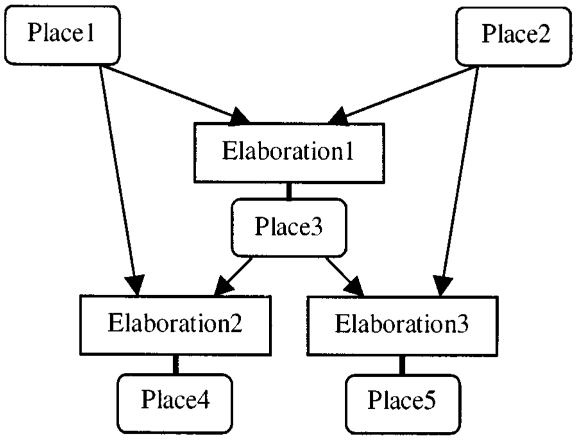

Figure 1. Basic E-graph structure.

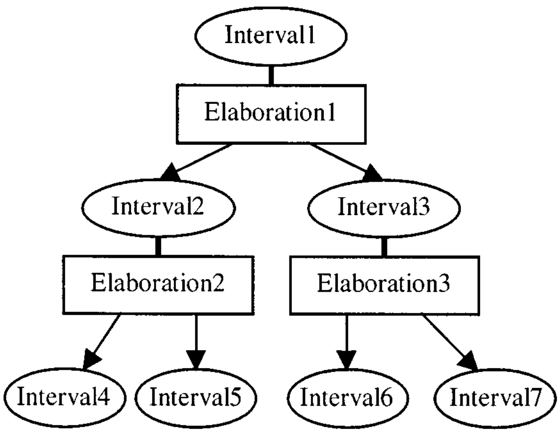

Figure 2. Tree structure based on E-graph of Figure 1.

2.2. PLACES

A place has properties of time, pitch and articulation. Times are expressed in terms of a particular metre, which is recorded with the time. A metre is a hierarchical set of divisions which, at each level, may be duple or triple. A metre of 3/4 has a triple division at the higher level and a duple division at the lower level, for example. A metre of 6/8, on the other hand, has a duple division at the higher level and triple at the lower level. These is no limit to the number of levels, and the highest level can correspond to a unit consisting of a number of bars (measures).

Pitches are represented in a manner which distinguishes between enharmonically equivalent pitches with different spellings. C sharp and D flat, for example, are represented as different pitches and can yield different results in the functions below. Just as times have a context of metre recorded with them, pitches have recorded contexts of key and harmony which effectively provide sequences of pitches in scales and arpeggios, respectively.

The property of **articulation** is intended to hold information about the manner in which a note is to be realised, but for the present it can have one of only two values: **tied** and **untied**. If the articulation is **tied**, and if a note of the same pitch is already sounding, then instead of stopping that note and restarting a note of the same pitch, the already sounding note is continued through the time occupied by the new note. (If the system for realising notes must know the duration of a note at its start, a look-ahead of one note is therefore required. If, however, the duration need only be known at the end of the note, as in simple MIDI, the look-ahead is zero.) The default is **untied**.

Certain functions are defined on times and pitches, used in the generation of new places by elaborations. For times, the essential function is **timeDivision(time1, time2)**, which yields a new time intermediate between **time1** and **time2**. The precise time yielded depends on both metres in a manner which need not be described in detail. Broadly speaking, the time coincides with the strongest beat between **time1** and **time2**, and, if there is more than one such beat, it is chosen to be approximately equidistant but with a bias for being later rather than earlier. A second function compares three times to determine whether the interval between the first and second is greater than, equal to or less than the interval between the second and the third.

For pitches, there are functions to yield new pitches which are a chromatic or scale step above or below a given pitch in the appropriate scale or a step above or below in the appropriate arpeggio, or an octave above or below. These functions, whose names are self-explanatory, are **chromaticUp(pitch)**, **chromaticDown(pitch)**, **stepUp(pitch)**, **stepDown(pitch)**, **arpeggioUp(pitch)**, **arpeggioDown(pitch)**, **octaveUp(pitch)**, and **octaveDown(pitch)**. In the case of **stepUp(pitch)** and **stepDown(pitch)**, the pitch must be a member of its key, and in the case of **arpeggioUp(pitch)** and **arpeggioDown(pitch)**, the pitch must be a member of its harmony. In no case can the pitch be the “null pitch” of a rest.

2.3. ELABORATION TYPES

An elaboration generates a new place on the basis of its two parent places. A musical language can be defined (to a degree) by the kinds of elaboration it allows. The set of elaborations described here is intended to be a partial definition of the language of melodies in the eighteenth and early nineteenth centuries. Considerably more refinement is required before it can be claimed that this language is adequately defined. At the current stage of research, the objective is to test the general framework as a system of representation rather than to accurately define any particular musical language.

Elaborations are distinguished by their characteristics in metre, time and pitch. By default, an elaboration is **neutral** in metre, which means that the new place is generated within the framework of the metre of whichever of the parents occurs at the lower metrical level, or the left parent if they occur at the same level. Specifying an elaboration as **duple** or **triple** allows duplets or triplets to be introduced where the normal division would be otherwise. If no division is specified either by the metre or the elaboration, it is assumed to be duple.

In the time domain, an elaboration is one of three types: **even**, **longShort** or **shortLong**. If the elaboration is **even** (which is the default), and if the times of its parent places are **time1** and **time2** respectively, the new place which it generates has a time equal to **timeDivision(time1, time2)**. If the elaboration is **longShort**, the time of the new place is the same as for **even** if the interval between **time1** and the new time is greater than the interval between the new time and **time2**. Otherwise the time of the new place is equal to **timeDivision(timeDivision(time1, time2), time2)**. For elaborations which are **shortLong**, the new time is similarly either the same as for **even** or **timeDivision(time1, timeDivision(time1, time2))**.

2.3.1. *Simple elaboration types*

A simple elaboration generates a new place at a time intermediate between its two parent places, in accordance with its specified temporal characteristic. Assuming that the pitches of the left and right parents are **pitch1** and **pitch2** respectively, the pitch of the new place is determined in accordance with the pitch characteristic of the elaboration, as in Table I. The table also specifies the context in which each elaboration type is valid. In the case of some, one pitch or the other must be in the specified harmony or scale. In the case of passing elaborations, there are constraints on how the children must be elaborated. (This is a complication arising from the insistence on binary branching in the graph. However, it seems a small price to pay for the advantage of having graphs which always have a comparable "shape".)

It is intended in future development to incorporate changes of harmony and key into elaborations, but for the present these are simply inherited by the newly generated place from one or both parents. The metre of the new place is similarly inherited, though it may be altered to take account of any explicit specification of a duple or triple division and/or extended to take account of any new metrical level. The articulation of the new place is **untied** in every case except **suspension** (see below).

The first four notes of *Frère Jacques* can now be represented as in Figure 3 which superimposes an E-graph on music notation showing the sequence of notes at each level. Boxes with rounded corners indicate places of the E-graph. Notes not in rounded boxes are included to complete the music notation and do not represent newly generated places. It will be seen that each such note has a pitch and time identical to a note directly above which does correspond to a place. The lower-level notes vary from those above only in their duration, but it will be recalled that duration is not a property of places but emerges from their sequence. The blank place at the end of the top level is intended to correspond to whatever immediately follows the extract, which in this case will be the first note of the next bar.

Table I. Simple elaboration types.

elaboration type	**new pitch**	**constraints**
shorten	null (new place is a rest)	none
repetition	pitch1	none
anticipation	pitch2	none
chromaticNeighbourAbove	chromaticStepUp(pitch2)	pitch2 must not be null
chromaticNeighbourBelow	chromaticStepDown(pitch2)	pitch2 must not be null
neighbourAbove	stepUp(pitch2)	pitch2 must be in key
neighbourBelow	stepDown(pitch2)	pitch2 must be in key
arpeggio1Above	arpeggioUp(pitch1)	pitch1 must be in harmony in key
arpeggio1Below	arpeggioDown(pitch1)	pitch1 must be in harmony in key
arpeggio2Above	arpeggioUp (arpeggioUp(pitch1))	pitch1 must be in harmony in key
arpeggio2Above	arpeggioDown (arpeggioDown(pitch1))	pitch1 must be in harmony in key
octaveAbove	octaveUp(pitch1)	pitch1 must not be null
octaveBelow	octaveDown(pitch1)	pitch2 must not be null
chromaticPassing1	if pitch1 $<$ pitch2: chromaticStepUp(pitch1) if pitch1 $>$ pitch2: chromaticStepDown(pitch1)	interval from pitch1 to pitch2 must be two semitones
chromaticPassing2	if pitch1 $<$ pitch2: chromaticStepUp (chromaticStepUp(pitch1)) if pitch1 $>$ pitch2: chromaticStepDown (chromaticStepDown(pitch1))	interval from pitch1 to pitch2 must be three semitones; there must be a chromaticPassing1 elaboration between the new place and the right parent
[other chromaticPassing*n* . . .]	. . .	. . .
passing1	if pitch1 $<$ pitch2: stepUp(pitch1) if pitch1 $>$ pitch2: stepDown(pitch1)	interval from pitch1 to pitch2 must be two steps in key
passing2	if pitch1 $<$ pitch2: stepUp(stepUp(pitch1)) if pitch1 $>$ pitch2: stepDown(stepDown(pitch1))	interval from pitch1 to pitch2 must be three steps in key; there must be a passing1 elaboration between the new place and the right parent
[other passing*n* . . .]	. . .	. . .

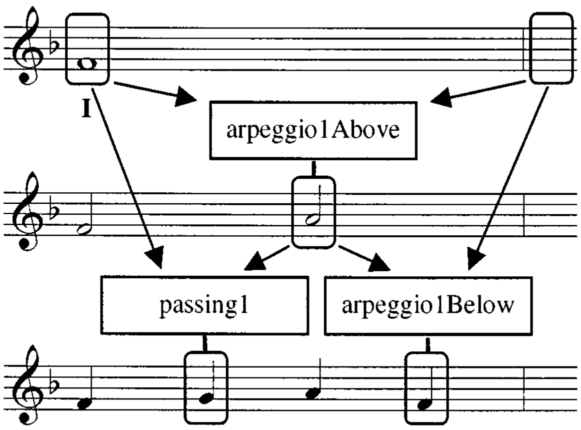

Figure 3. E-graph representation of the beginning of *Frère Jacques*.

Just as the graph in Figure 1 can be transformed into the tree in Figure 2, so the graph in Figure 3 can be transformed into a tree. If one is concerned only with the pitch pattern of these notes, and not with the actual pitches themselves, then the root interval can be omitted from the tree, and all the lower-level intervals which depend on it, leaving only a tree of elaborations. Such a tree can be notated using a structure of brackets of the form [<elaboration> <left child> <right child>]. A point will be used to indicate a child which is not elaborated. The first four notes of *Frère Jacques* then have the pattern [arpeggio1Above [passing1 . .] [arpeggio1Below . .]].

2.3.2. *Accented elaboration types*

In the case of some types of musical elaboration, a new note is not simply inserted between two existing notes, leaving the time of the original notes unchanged, but instead the result of the elaboration is a note at the time of the original note but with a different pitch, and a note with the original pitch but at a new, intermediate, time. The most obvious case is that of an appoggiatura, which is like a neighbour note in that it is one step above or below the note elaborated (usually above), but it starts at the original time of that note and shifts the original note to a later time. Elaborations like arpeggiations can also occur in this manner, where the note elaborated is shifted forward in time and replaced at its original time by a note above or below in the arpeggio of the prevailing harmony. The system of representation described here therefore includes accented versions of neighbour elaborations, called appoggiaturas, whose new pitches are as in Table I except that **pitch1** must be substituted for **pitch2**, and accented versions of arpeggio and octave elaborations. Table II lists other accented elaborations, the new pitches they yield, and their constraints. Note that the constraints for suspensions and accented passing notes are considerably more complex because they require examination of a neighbouring segment of the

Table II. Additional accented elaboration types.

elaboration type	**new pitch**	**constraints**
delay	null (new place is a rest)	none
chromaticSuspensionAbove	chromaticStepUp(pitch1) new articulation is **tied**	pitch0 must equal new pitch
chromaticSuspensionBelow	chromaticStepDown(pitch1) new articulation is **tied**	pitch0 must equal new pitch
suspensionAbove	stepUp(pitch1) new articulation is **tied**	pitch0 must equal new pitch
suspensionBelow	stepDown(pitch1) new articulation is **tied**	pitch0 must equal new pitch

accentedChromaticPassing	if pitch0 < pitch1:
	chromaticStepDown(pitch1)
	if pitch0 > pitch1:
	chromaticStepUp(pitch1)
		interval from pitch0 to pitch1
		must be two semitones

accentedPassing	if pitch0 < pitch1:
	stepDown(pitch1)
	if pitch0 > pitch1:
	stepUp(pitch1)
		interval from pitch0 to pitch1
		must be two steps in key

graph to determine whether or not the elaboration is valid, and **pitch0** refers to the pitch of the place preceding in sequence the left parent place. (See below for a definition of how the sequence of places is determined in a graph.)

Since accented elaborations generate *two* places, one of which effectively replaces the left parent, the manner in which elaborations in adjacent parts of a graph are connected to places becomes crucial, and different results can follow from different linkages. This is illustrated in Figure 4. The transformation to a tree representation with intervals instead of places is still possible, but a new kind of elaboration is required which yields just one interval whose first note is the same as the first note of its parent but whose second note is different. A tree representation of Figure 4b is then as in Figure 5. Since the yield of the elaboration required is always dependent on the corresponding elaboration in the adjacent branch of the tree, a hyphen symbol will be used for this interval-altering elaboration. These elaborations can be included in bracket notations (though they complicate their structure slightly by having only one child) and so render them unambiguous in the presence of accented elaborations. The patterns of Figures 4a and 4b are then [repetition [neighbourBelow . .] [accentedArpeggio1Above . .]] and [repetition [– [neighbourBelow . .]] [accentedArpeggio1Above . .]] respectively.

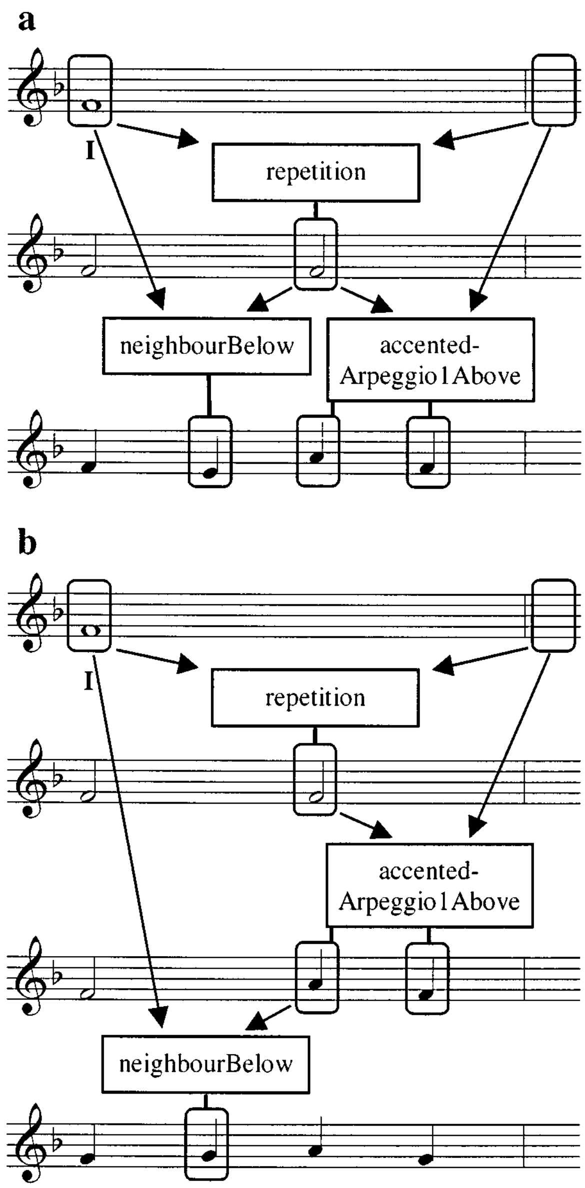

Figure 4. Two ways of attaching to an accented elaboration.

3. Interpreting E-Graphs

3.1. DERIVING MELODIES

To realise a melody from an E-graph, the elaborations can be ignored (except to the extent of identifying whether they are accented or not), and a sequence of places can be derived from the graph, representing the sequence of notes and rests in the

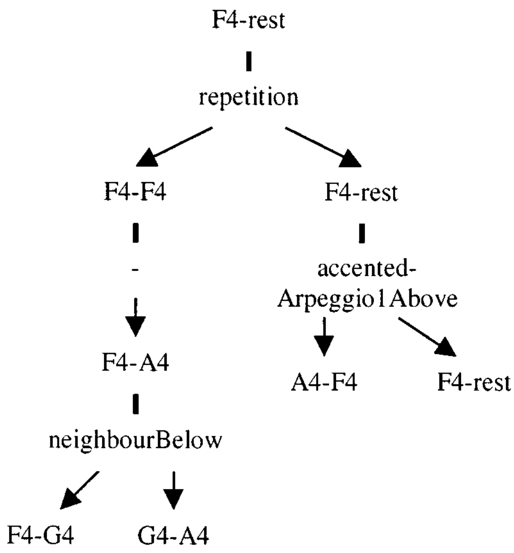

Figure 5. Tree representation of Figure 4b.

melody. In the absence of accented elaborations, this conversion of the graph to a sequence is simple, and every place in the E-graph occurs in the sequence. The first place is always the left-most top-level place. Thereafter, the next place in the sequence is always the lowest-level right child or, if there is none, the right parent, until a place is reached with no right child or parent.

In the presence of accented elaborations, however, the derivation of a sequence of places is more complex. Some places in the E-graph will not occur in the sequence and the elaborations cannot be ignored. As before, the first place is the left-most top-level place, and whenever there is a right child, the lowest-level such child is always the next place. However, if there is no right child, the next place is not necessarily the right parent because it might have been replaced in an accented elaboration. Thus the right-child elaborations of the right parent place must be checked for accented elaborations, and if one is found the next place in sequence is the first of the two places it generates, except that that place too must be recursively checked for further accented elaborations.

3.2. SEGMENTATION: S-TREES

It is common to segment melodies into groups of about five notes or less, and sometimes groups of groups, etc., generally resulting in a tree structure of segments. (See Lerdahl and Jackendoff, 1983 for example.) This segmentation is generally based on a bottom-up measure of intervals in pitch and time, generally placing boundaries between segments at large intervals, and also on the recurrence of patterns. Both factors are present in the "grouping rules" of Lerdahl and Jackendoff (1983, pp.

345–347), but only grouping by intervals is developed to any degree. Procedures for grouping by pattern have been proposed by Baker (1989) and Cambouropoulos (1998a, pp. 115–129; 1998b). It is proposed here that the structure of elaboration should also be a factor in determining segmentation, so that S-trees, which indicate segmentation, are based on E-graphs.

Although each elaboration in an E-graph is linked to two parents, in many cases information is taken from only one parent in determining the pitch of the new place. Although the determination of time takes information from both parents, the metre of one parent is often the same as for the other, and since the time intervals between the top-level places are commonly even, the time of one parent can often be inferred on the basis of the other and the context of elaborations. Thus **neighbourNote-** and **anticipation** elaborations take no essential information from their left parent, and **repetition**, **arpeggio-**, and **octave-** elaborations, together with most accented elaborations, take no essential information from the right parent. Thus, in these cases, it is possible to delete one link from the graph without losing any essential information for determining new pitches or the validity of elaborations. Information needs to be added to some remaining links to allow new times to be properly determined; essentially what is required is that a link specify the metrical level at which the new place is to be created. (In an E-graph, this can be determined from the context.) If all elaborations are one of these types, the resulting structure is a set of trees, each with a top-level place at its root. This tree structure implies a segmentation. The **suspension-**, **passing-** and **accentedPassing-** elaborations do not require information from both parents to determine their pitch, but their validity in a particular context can only be determined by reference to both parents in the case of passing notes and from other parts of the graph in the case of suspensions and accented passing notes. Therefore, while it would be possible to delete some links in these cases also, and arrive at a true tree structure, this structure would not suggest a segmentation in the same way.

Figure 6 shows E-graph and S-tree representations of the first two bars of melody from Mozart's piano sonata in C major, K. 545. Lines of the E-graph which carry no useful information and so can be safely omitted are shown dashed. The corresponding lines which carry all useful information from one parent are shown heavier. The lighter solid lines show where information is required from both parents. The S-tree then corresponds to the structure formed of solid lines. Note that the left-hand branch of this is a true tree whereas the right is not because of the accented-passing-note elaboration. The segmentation of this melody which the representation implies is shown below the music notation.

Just as the pattern embodied in an E-graph can be given a bracket notation which ignores places, a more complex bracket notation is also possible for S-trees, with the basic form {<elaboration(s)> <child1> <child2> . . . <child*n*>}. Each child is a place (rather than an interval as for E-graphs) and there can be more than one elaboration in the first item. It might be, for example, that a place is elaborated with branches on both the left and the right in an S-tree, as is the case in Figure 6 for the

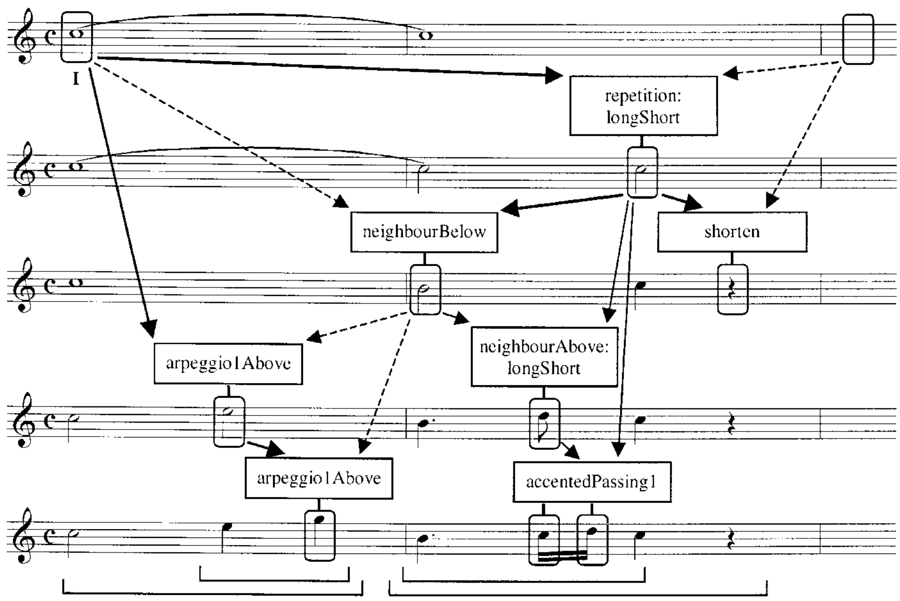

Figure 6. Transformation of an E-graph to an S-tree; based on a fragment of Mozart's piano sonata K. 545.

minim (half note) C. In such cases the two elaborations are shown in the first item, separated by a slash, and there are three children. In more complex cases (including Figure 6) where the S-tree is not a true tree, a portion of E-graph is incorporated into the elaboration, following an ampersand. The E-graph is applied to the interval between the two places which are the children of the elaboration preceding the ampersand. The number of places generated by the E-graph is therefore added to the number of children for this compound elaboration. The notation for the S-tree in Figure 6, for example, would be {repetition:longShort {arpeggio1Above . . {arpeggio1Above . .}} {neighbourBelow&[neighbourAbove:longShort [accentedPassing1 . .] .]/shorten . . . . .}.

It is not uncommon that a place could derive from more than one type of elaboration. Every **passing1** elaboration, for example, could be a **neighbour-** elaboration also. If the interval between two places is one scale step, then a **repetition** elaboration and a **neighbour-** elaboration would produce the same place. It is proposed that in cases like this, when the link to either the left parent or the right parent in an E-graph can be deleted in the construction of a set of S-trees, that the decision on which to delete should be based on the common segmentation factors mentioned above: the size of pitch and temporal interval, and the recurrence of pattern. Since pattern is here defined as emerging from an E-graph representation, there must be feedback in the parsing of a melody to produce a representation, and segmentation and parsing must proceed together.

The first four notes of *Frère Jacques*, for example, can be represented as [repetition:longShort [arpeggio1Above [passing1 . .] .] .] as well as in the two ways already indicated. However, the first representation is preferred because it allows matches with the pattern of the following two phrases. If an asterisk is used as a “wild card”, the first and second phrases share the pattern [arpeggio1Above [passing1 . .] *] and the first and third share the pattern [arpeggio1 * [* * *] [arpeggio1Below . .]].

4. Representing Pattern

Melodic patterns which are hard to demonstrate in representations using pitch or interval sequences can clearly emerge from E-graph representations. This is particularly true in two kinds of case of which examples are given below. The first is when the same pattern gives rise to different sequences of pitches or intervals because of differences of harmonic or pitch context. The second is when a pattern is seen not in the “surface” succession of notes but at an underlying level.

4.1. PATTERNS IN DIFFERENT CONTEXTS

Bars (measures) two to seven of Domenico Scarlatti’s sonata in A minor, K. 3, clearly contain a thrice-recurring pattern in the right hand. Yet the first interval is twice a fourth and once a third. The second interval is different each time. When shown in an E-graph representation, as in Figure 7, the three occurrences are shown to be identical in terms of the pattern of elaborations. Their differences arise purely from the different places at the top level. (In a full representation of the music, these places would be generated by other, higher-level elaborations.) In the first two occurrences the pitch of the second place is a semitone below the pitch of the first place, while it is a semitone above at the third occurrence, and the harmony varies at each place also, causing different intervals to arise from the arpeggio elaborations.

4.2. UNDERLYING PATTERNS

The extracts in Figure 8 are taken from a variation movement by Mozart, so it can be safely assumed that the composer intended these to be heard as different but related. The relation is most clearly heard in the rhythm of the second bar, but there are similarities in the pitch structure also which are made clear in an E-graph representation. (Three liberties have been taken in this representation. The first is in the change of articulation of the last A to **tied** in the first and third extracts, and the others are in the new harmonies for the quaver (eighth note) G at the beginning of the first extract and the triplet quaver A at the end of the second extract. Properly, these details should arise from elaborations, but that must await further development of the representation system. As before, the sequence

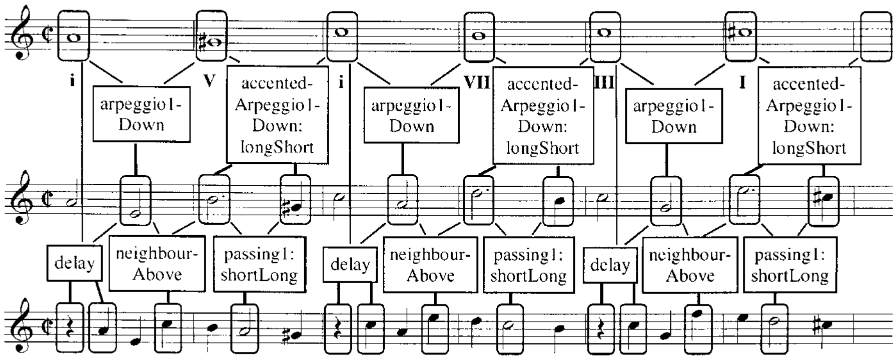

Figure 7. E-graph representation of the right hand of bars 2–7 of Domenico Scarlatti's sonata in A minor, K. 3.

of top-level places should, in a full representation, be generated by elaborations, but this requires both a richer language of elaborations and more context than these short extracts.) The first extract (which is from the theme) and the second extract have different top-level places, though their harmonies are the same and the last is an octave transposition of the corresponding place in the first extract. The underlying similarity between the two is made clear by the matching of portions of the E-graphs, which can be expressed in the common underlying pattern [* * *] [arpeggio1Above:longShort [accentedArpeggio1Below:shortLong . .] .] [appoggiaturaAbove * .]. The match is weak in the first part of this, but the underlying pattern does show that there should at be a note one quaver (eighth note) after the initial A, and perhaps others also.

The third extract has exactly the same top-level places as the first, and differs from it only in that an **accentedArpeggio** elaboration is replaced by a **neighbourAbove**, and the children of the initial passing elaboration are not further elaborated. Both extracts have the underlying pattern [passing1 * *] [arpeggio1Above:longShort [*:shortLong . .] .] [appoggiaturaAbove . .]. All three extracts thus share the pattern [* * *] [arpeggio1Above:longShort [*:shortLong . .] .] [appoggiaturaAbove * .].

5. Discussion and Application

The manner of representation described here has a number of potential uses, though further work is required before effective use can result.

5.1. FUTURE DEVELOPMENTS

The language of elaborations described here is based on the author's beliefs about eighteenth-century and early-nineteenth-century style and the elaborations which have appeared necessary to represent the examples and similar short extracts. A

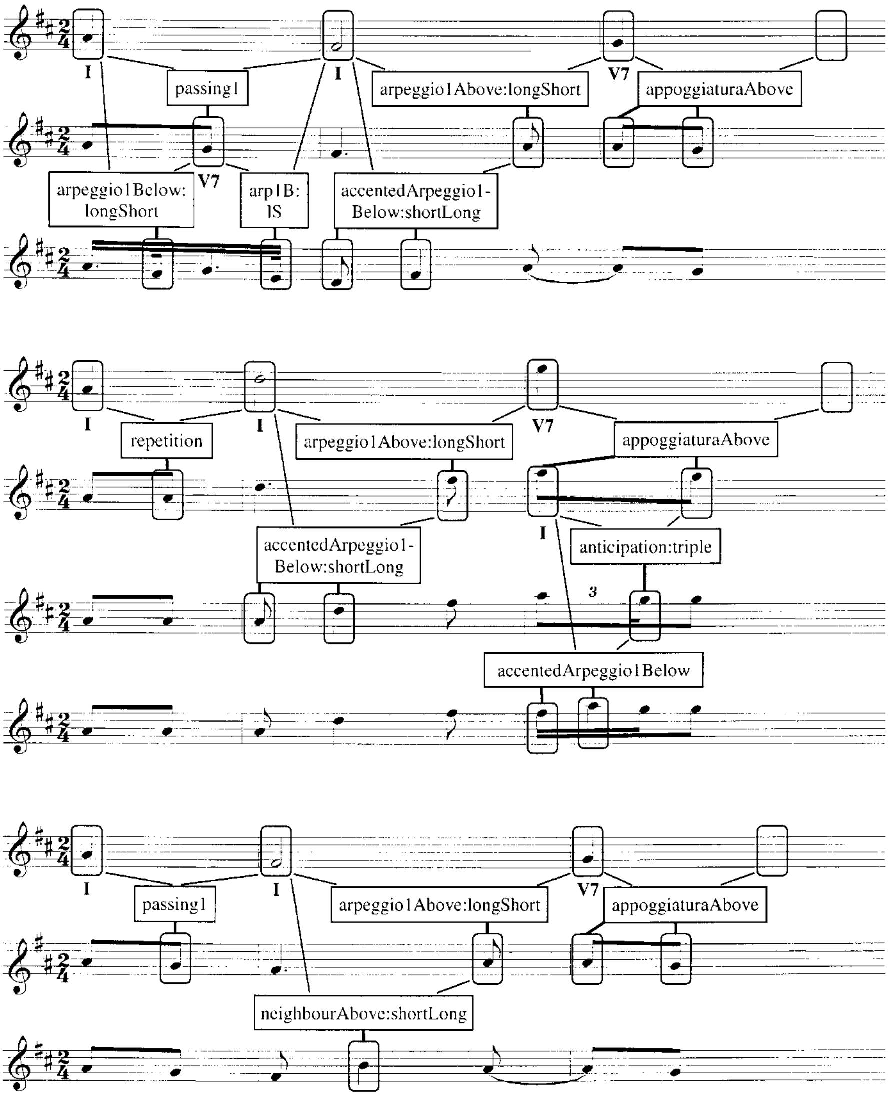

Figure 8. E-graph representations of extracts from the third movement of Mozart's string quartet in A major, K. 464.

more rigorous testing with significant corpuses of music is required to refine the set of elaborations and to define different musical languages.

Most music of the western tradition is polyphonic, yet the system of representation described here can represent only single-line melodies. To allow the representation of polyphonic music, one approach would be to represent separate voices with separate E-graphs, but there would have to be some co-ordination between the graphs to ensure agreement in harmony, metre and segmentation, where appropriate. An alternative approach would be to expand the definition of an E-graph and allow places to represent not just a single note but a number of simultaneous notes. It is expected that a combination of both approaches would be most successful. This issue is of importance in the representation of melodies also because many melodies appear to be best analysed as having an underlying structure which is polyphonic rather than monophonic. This is most obvious in so-called “pseudo-polyphony”, common in Baroque music, when one actual voice effectively jumps from one underlying voice to another in rapid figuration. However, the essential phenomenon underlies many less obvious case also.

Most importantly, a mechanism for parsing melodies to derive E-graphs is required. It has already been demonstrated that the same melody can have different representations, as in Figures 3 and 4b, for example. While the number of alternative elaborations at each point in an E-graph is small, the different possible combinations of such alternatives are likely to lead to the total number of possible E-graphs for a melody being exponentially related to the number of notes in that melody. A mechanism for determining the preferred interpretation in the course of deriving an E-graph is therefore required. The comments concerning segmentation suggest one strategy, but the feedback required in the discovery and application of pattern will be complex.

5.2. GENERATION OF MELODIES

Application in analysis is clearly possible, and it is hoped that more rigorous analyses of the kind found in Marsden (1987) may be undertaken. However, effective application in analysis must await the developments described above. For the generation of melodies, however, it is of less consequence that the language is imprecise or incomplete. An implementation of E-graphs in software allowing the user to create and manipulate graphs is under development and may be viewed at http://www.lancs.ac.uk/staff/marsdena/software/. One complex feature of this software is how it responds when the user copies a segment of an E-graph or S-tree to a position where it cannot validly fit. (The new context might have an interval too small for the number of passing notes, for example.) The software can make minimal amendments to the segment of E-graph or S-tree to allow it to fit, such as inserting or deleting extra **passing-** elaborations, or by changing a **neighbourAbove** elaboration to **neighbourBelow**. It is suggested that these procedures correspond to the specialised knowledge a composer brings to bear in using a melodic pattern in a particular context, though no particular claim is made for the appropriateness of the particular strategies followed by the software.

One planned application is to use this representation and these procedures in a system which generates melodies automatically on the basis of gestural input from the user. Other software will analyse the input for its level of activity, regularity, granularity, or other such global features. The system will then generate E-graphs which have similar global features. Regularity might be determined, for example, by the degree of recurrence of common patterns, activity by the depth of the E-graph, and granularity by the strength of segmentation in the corresponding S-tree. A very early version of such a system was implemented by the author in HARP (Camurri et al., 1992) at the University of Genoa in 1997, with promising results.

Acknowledgements

I gratefully acknowledge the assistance and hospitality of Prof. Antonio Camurri and his colleagues at the University of Genoa, where this work was begun, and the financial assistance of the Leverhulme Trust in enabling me to work there.

References

Baker, M. “A Computational Approach to Modeling Musical Grouping Structure”. *Contemporary Music Review*, 4 (1989), 311–325.

Baroni, M., R. Dalmonte and C. Jacoboni. “Theory and Analysis of European Melody”. In *Computer Representations and Models in Music*. Eds. A. Marsden and A. Pople, London: Academic Press, 1992, pp. 187–205.

Cambouropoulos, E. “A General Pitch Interval Representation: Theory and Applications”. *Journal of New Music Research*, 25 (1996), 231–251.

Cambouropoulos, E. *Towards a General Computational Theory of Musical Structure*. PhD thesis, University of Edinburgh, 1998a.

Cambouropoulos, E. “Musical Parallelism and Melodic Segmentation”. *Proceedings XII Colloquium on Musical Informatics*, Gorizia (1998b), 111–114.

Camurri, A., C. Canepa, M. Frixione and R. Zaccaria. “HARP: A System for Intelligent Composer’s Assistance”. In *Readings in Computer Generated Music*. Ed. D. Baggi, Los Alamitos, California: IEEE Computer Society Press, 1992, pp. 95–115.

Deutsch, D. “The Processing of Pitch Combinations”. In *The Psychology of Music (2nd edition)*. Ed. D. Deutsch, San Diego: Academic Press, 1999, pp. 349–411.

Deutsch, D. and J. Feroe. “The Internal Representation of Pitch Sequences in Tonal Music”. *Psychological Review*, 88 (1991), 503–522.

Kippen, J. and B. Bel. “Modelling Music with Grammars: Formal Language Representation in the Bol Processor”. In *Computer Representations and Models in Music*. Eds. A. Marsden and A. Pople, London: Academic Press, 1992, pp. 207–238.

Lerdahl, F. and R. Jackendoff. *A Generative Theory of Tonal Music*. Cambridge, Mass.: MIT Press, 1983.

Marsden, A. “A Study of Cognitive Demands in Listening to Mozart’s Quintet for Piano and Wind Instruments, K.452”. *Psychology of Music*, 15 (1987), 30–57.

Schenker, H. *Free Composition (Der freie Satz)*. Translated and edited by E. Oster. New York: Longman, 1979. (Original German publication, 1935.)

Selfridge-Field, E. “Conceptual and Representational Issues in Melodic Comparison”. In *Melodic Similarity: Concepts, Procedures and Applications (Computing in Musicology, 11)*. Eds. W. Hewlett and E. Selfridge-Field, Cambridge, Mass.: MIT Press, 1998, pp. 3–64.

Sundberg, J. and B. Lindblom. “Generative Theories for Describing Musical Structure”. In *Representing Musical Structure*. Eds. P. Howell, R. West and I. Cross, London: Academic Press, 1991, pp. 242–272.

West, R., P. Howell and I. Cross. “Musical Structure and Knowledge Representation”. In *Representing Musical Structure*. Eds. P. Howell, R. West and I. Cross, London: Academic Press, 1991, 1–30.