Chapter 11
Automated Motivic Analysis: An Exhaustive Approach Based on Closed and Cyclic Pattern Mining in Multidimensional Parametric Spaces

Olivier Lartillot

Abstract Motivic analysis provides very detailed understanding of musical compositions, but is also particularly difficult to formalize and systematize. A computational automation of the discovery of motivic patterns cannot be reduced to a mere extraction of all possible sequences of descriptions. The systematic approach inexorably leads to a proliferation of redundant structures that needs to be addressed properly. Global filtering techniques cause a drastic elimination of interesting structures that damages the quality of the analysis. On the other hand, a selection of closed patterns allows for lossless compression. The structural complexity resulting from successive repetitions of patterns can be controlled through a simple modelling of cycles. Generally, motivic patterns cannot always be defined solely as sequences of descriptions in a fixed set of dimensions: throughout the descriptions of the successive notes and intervals, various sets of musical parameters may be invoked. In this chapter, a method is presented that allows for these *heterogeneous* patterns to be discovered. Motivic repetition with local ornamentation is detected by reconstructing, on top of "surface-level" monodic voices, longer-term relations between non-adjacent notes related to deeper structures, and by tracking motives on the resulting syntagmatic network. These principles are integrated into a computational framework, the *MiningSuite*, developed in Matlab.

11.1 Systematic and Automated Motivic Analysis

One main dimension of music analysis is related to the repetition of particular sequences and their development throughout pieces of music. These sequential repetitions form core theoretical aspects of music, whose actual denomination (motive, theme, cell, etc.) depends on their specificity within genres, the way they repeat within pieces and their sizes (cells and motives typically correspond to very short patterns,

Olivier Lartillot
Department of Architecture, Design and Media Technology, Aalborg University, Aalborg, Denmark
e-mail: ol@create.aau.dk

whereas themes are typically several bars in length). Studying these repetitions is an important part of the analytical characterization of particular pieces of music, as well as a means of deciphering the stylistic particularities of pieces in the context of other related works (e.g., other works by the same composer or in the same genre or from the same period).

Throughout the long tradition of motivic analysis (understood in a broad sense to include thematic analysis), one can appreciate the value but also the possible limitations of analyses carried out entirely by hand by musicologists, essentially driven by musical intuition. Music reveals itself as a hugely rich and complex structure, that cannot be easily grasped by human endeavour, no matter how expert it might be. Twentieth-century musicology, nurtured by the advent of linguistics, structural anthropology, semiology and cognitive science (Nattiez, 1990; Ruwet, 1987), proclaimed the need for more formalized and systematic analytical processes. Yet no working methodology towards that aim has yet been initiated, due to the huge structural complexity underlying this issue. Systematic attempts have been hampered by the underlying complexity of possible strategies and structures. Formalized descriptions of segmentation strategies offered by psychological and cognitive studies might be utilized as a means of guiding and making discovery processes explicit.

Computational modelling seems to be the most natural way of formalizing and automating thematic and motivic analysis, allowing for exhaustive analyses of sizeable pieces of music to be carried out. Yet, it is still a struggle to control the combinatorial explosion of structures and to offer musically relevant analyses. The main question of how to model such analytical processes remains open. This chapter presents a model that may offer an answer to these problems. Section 11.2 introduces the general principles of the approach that can discover motives defined along multiple parametric dimensions, such that each successive note of the motivic pattern can be defined on different parameters. Section 11.3 explains the issues related to structural proliferation of redundant structures. We show how these questions have been addressed in previous work and how a solution based on closed and cyclic patterns allows for a compact, but detailed structural analysis. Section 11.4 generalizes the model to the study of ornamented patterns.

11.2 Heterogeneous Multiparametric Pattern Mining

11.2.1 Why Heterogeneous Patterns?

What is a *motive*? At first sight, it could be considered a sequence of notes that is repeated throughout a piece or a corpus of pieces of music. However, motives can be transformed in various ways (e.g., transposed), so that what is repeated are not the notes themselves, but more generally musical *descriptions*. So a motive is a *pattern* representing a sequence of descriptions that is repeated at least once in a piece of music.[1] Each successive description represented by a pattern will be called the *pattern's individual description*. Each sequence of notes in the score whose sequence of descriptions is compatible with the sequence of descriptions defined by a pattern forms an *occurrence* of that pattern.

Music is a complex signal that conveys a lot of information:

1. Scores contain many notes, and, at each instant, several of them can sound together to form chords along multiple instrumental voices.
2. Each note conveys data along several parameters, related to diverse musical dimensions (pitch, rhythm, accentuation, etc.) and to the musical context: a metrical grid determines metrical positions, tonal/modal scales govern diatonic representations of pitches, pitch intervals and gross contours can be defined in particular with the previous note, rhythmic values are defined with respect to the next note and so on.
3. A large range of structures can be inferred from those configurations, based on motivic repetition, local (dis)continuities along the various parametric dimensions and so on.

In this chapter, the focus is on monodic music, consisting of a succession of notes without superpositions. Hence we ignore point 1 above, but still have to consider points 2 and 3. Concerning point 3, this chapter will mainly address the question of motivic repetition. However, in Sect. 11.4.2, local (dis)continuities are discussed, particularly with respect to how they interact with motivic repetition.

According to point 2, the successive descriptions that form a motive can be defined along various musical dimensions. In previous studies this has been addressed either by the notion of "viewpoints" (Conklin and Anagnostopoulou, 2001) (see also Chap. 15, this volume) or through representing pieces of music as sets of points in multi-dimensional spaces (Meredith et al., 2002) (see also Chaps. 13 and 17, this volume). These previous studies do not encompass the entire set of possible motivic descriptions: often motives can be characterized by particular parametric descriptions that are applied to particular notes and intervals of the motives, but not all (Conklin and Bergeron, 2008; Lartillot, 2005). In the following sections, some concrete examples of what can be called *heterogeneous patterns* will be developed.

11.2.2 Musical Parameters

Figure 11.1 shows the hierarchical parametric space presented in this chapter. Additional parameters can be integrated if required. This figure should be read as follows:

- Parameters indicated in bold letters indicate the detailed information composing each note description: for instance for the second note in Fig. 11.2 (pitch F♯4 at

[1] In this chapter, a "*pattern*" represents a sequence of descriptions that is repeated at least once. If a sequence is not repeated, it is not considered to be a pattern.

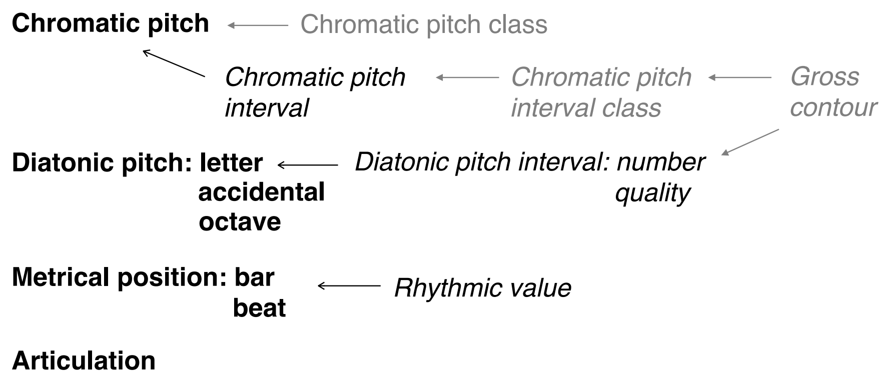

Fig. 11.1 Dimensions of the hierarchical parametric space currently used in the proposed model. See the text for an explanation of the schema

note 1-2),[2] the chromatic pitch value is the MIDI pitch value 66, the diatonic pitch letter is F, accidental is sharp, octave is 4, metrical position is bar #1, beat #2. Other information could be considered as well, such as articulation (e.g., staccato, marcato).

- Parameters shown in grey are directly deduced from more *specific* information: for our previous example, chromatic pitch class is 6 and is obtained from the chromatic pitch information, by simply performing a modulo 12 operation.
- Parameters written in italics describe intervals between successive notes, and are therefore obtained directly as a difference of the parameters associated with each note. To continue our previous example, the second note is preceded by the first note 1-1 (pitch D4 and a rhythmic value of quarter note): the interval between the two notes has chromatic pitch interval +4 semi-tones, diatonic pitch interval number +2 (an increasing third interval, i.e., 2 degrees upwards) and quality ‘major’, and rhythmic value 1 (a quarter note being the unit in 3/4 metre).
- Interval descriptions can themselves accept more *general* information also indicated in grey: here, chromatic pitch interval class, which is chromatic pitch interval modulo 12, is 4 and gross-contour is ‘increasing’.

11.2.3 Pattern Prefix Tree

Following a common straightforward convention, all the motives (i.e., patterns) extracted in the analysis are stored in a single prefix-tree (or *trie*), such as the one shown in Fig. 11.3. Each pattern is represented by a chain from the root, $\emptyset$, of the tree to one particular node N_L. This chain $\emptyset \rightarrow N_1 \rightarrow \ldots \rightarrow N_L$ represents the successive

[2] Throughout this chapter, each note is identified with a label of the form i-j, where i is the bar number and j the beat index in the bar.

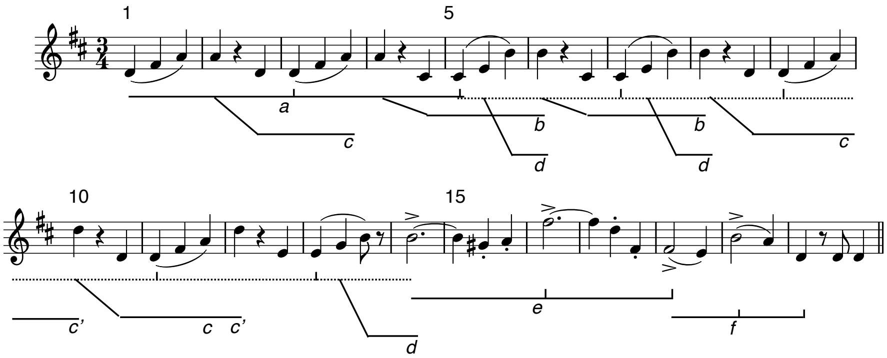

Fig. 11.2 Reduced and simplified version of the beginning of the first waltz theme in Johann Strauss' op. 314: "*An der schöne blauen Donau*". Pattern occurrences are shown below each stave. Patterns are described in Fig. 11.3. The figure is further discussed in Sect. 11.2.4. The horizontal lines with graduation notches correspond to cyclic patterns, as described in Sects. 11.3.3. Patterns branching out from cyclic patterns, such as pattern *d*, are figure against cyclic background, as further explained in Sect. 11.3.4

prefixes of the pattern. The whole pattern can therefore be identified with the final node N_L. The length of pattern N_L is the number of nodes in that chain, root excluded, i.e., L. In our illustration, pattern *d* is of length 2, pattern *f* of length 3, and pattern *a* is of length 6.

Patterns are commonly constructed incrementally, by progressively adding new nodes in the pattern tree, which initially is represented by only one node, i.e., the root of the tree. Two cases should be considered:

- Adding a node as a direct child of the root of the tree corresponds to the discovery of the repetition of a single note description.
- In other cases, the parent of that node already corresponds to a given pattern, so the new node indicates that several occurrences of that pattern are immediately followed by the same description. Notes that immediately follow occurrences of a pattern are called the *continuations* of that pattern.

Each occurrence of a pattern in the musical sequence is a structure connected to particular notes of the sequence and, at the same time, aligned with the corresponding pattern. Such a structure interconnecting the concrete score with the abstract pattern dictionary will be called a *construct*.

11.2.4 Illustration

The concepts developed in this section are illustrated on a particular example: a reduced and simplified version of the beginning of the first waltz theme in Johann Strauss' op. 314: "*An der schöne blauen Donau*", shown in Fig. 11.2. The prefix-tree is shown in Fig. 11.3. Motive *e*, for instance, is shown on the sixth branch

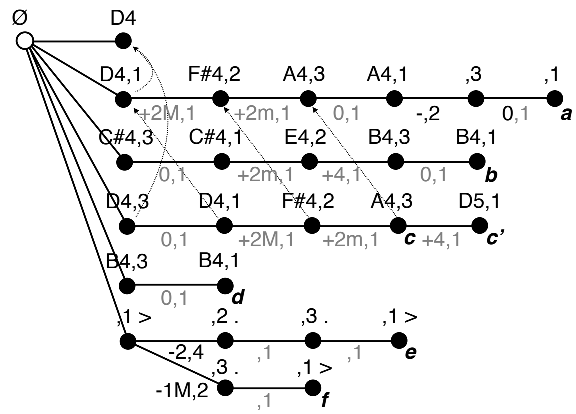

Fig. 11.3 Pattern trie related to the motivic analysis of the score shown in Fig. 11.2. The white disk represents the root, $\emptyset$, of the tree, while other nodes are shown with black disks. Each note description is written next to its node; interval descriptions are shown along the edges between nodes. See the text for further explanation. Arrows link more specific to less specific patterns, as explained in Sect. 11.3.1

starting from the top, ending at the node labelled '*e*'. The successive descriptions of the notes composing the motive are shown on each successive node in the branch: '$,1>$' represents a note on the first beat of a bar (1) that is accented ($>$); '$,2.$' represents a note on the second beat that is staccato (.), and so on for '$,3.$' and '$,1>$'. Further description of the motive is also associated with the intervals between successive notes, and represented on the edge between successive notes. In the example, there is an interval description between the two first nodes, indicated '$\overrightarrow{-2,4}$',[3] which represents a decreasing third interval ('-2' indicating two degrees down) and a 4-beat rhythmic value ('4'). The whole pattern can thus be described as "$,1 > \overrightarrow{-2,4}\ ,2.\ \overrightarrow{,1}\ ,3.\ \overrightarrow{,1}\ ,1>$".[4]

We can see that the descriptions of the successive notes and intervals of these motives do not always use the same types of musical parameter. Heterogeneous patterns cannot be defined solely as sequences of descriptions in a fixed set of dimensions.

[3] Parameters related to the interval between successive notes are indicated with an arrow above the parameter descriptions.

[4] The interval descriptions related to the two subsequent intervals, both "-1", do not convey additional information: they directly result from the difference of the parametric description of the corresponding pair of notes. For that reason, this information is shown in grey in Fig. 11.3.

11.2.5 Heterogeneous Pattern Construction

The method presented in this section is capable of discovering heterogeneous patterns. Instead of first decomposing the music into a superposition of viewpoints on which motivic analysis is performed, each note is associated with its entire set of related descriptions, and the analysis is performed on the whole set of descriptions at once. The description of a given note consists not only of the descriptions associated specifically with that note, but also includes the descriptions of the interval with its previous note.

Focus on particular dimensions is made adaptively for each successive step of the incremental construction of patterns. If, for instance, two notes share the same pitch value, D4, but have different other parameters, they become an occurrence of the pattern described solely by the pitch parameter “D4”, shown as the upper child of the root in Fig. 11.3. If both notes are also on beat 1 of a bar, that beat representation is indicated as well in the pattern (“D4,1”). If what is repeated is just an accented note on the first beat of a bar, we obtain the pattern “,1$>$”.

A given pattern can be extended by a new description in the same way. To continue our example, notes 1-1 and 3-1 are occurrences of the pattern “D4,1”. Each occurrence is continued with the same pitch F$\sharp$4 on notes 1-2 and 3-2. Hence pattern “D4,1” can be extended with a new node related to the note-description “$\mathrm{D4}, 1 \overrightarrow{+2\mathrm{M}, 1}\ \mathrm{F}\sharp 4, 2$”. The interval-description “$\overrightarrow{+2\mathrm{M}, 1}$” indicates the pitch interval of an ascending major third (two degrees upwards, in major mode) and the rhythmic value of one quarter note.

11.2.6 One-Pass Approach

The classical method for pattern construction (Cambouropoulos, 2006; Pasquier et al., 1999; Wang et al., 2007) would be to first find all patterns of length 1, then all patterns of length 2, and so on. To extend a given pattern, all its continuations are compared: any repetition of a given parameter in at least two of these continuations leads to the inference of a new hypothetical pattern extension.

In contrast, the pattern mining approach adopted here is based on an entirely chronological approach, where, for each successive note in the score, all possible patterns are inferred. This *one-pass* approach allows for redundant information to be discarded more easily, as explained in Sect. 11.3.2.

11.2.6.1 Pattern Continuation Memory and Its Use

In the one-pass approach, the detection of new patterns is made possible through the use of *associative tables*, one table for each parameter, that store, for each parameter value, the set of notes in the score that share that particular value. For instance, the associative table in the top left-hand corner of Fig. 11.4 is related to pitch, and stores, for each pitch value (D4, F♯4, etc.), the set of notes where this pitch occurs.

For each pattern, P, all its possible continuations are stored in a set of dedicated associative tables called *continuation memory*, associated with pattern P. In Fig. 11.4, continuation memories related to patterns in the pattern tree introduced in Fig. 11.3 are shown.

- The root of the tree $\emptyset$ stores the descriptions of all the notes in the score in a dedicated continuation memory. In the example in Fig. 11.4, it is composed solely of one associative table storing notes according to their pitches. This is precisely the aforementioned table in the top left-hand corner of the figure.
- Each child of $\emptyset$ corresponds to patterns made of one single note. For instance, the topmost child of $\emptyset$ in Fig. 11.4 is related to pattern "D4". Its continuation memory stores all the notes that immediately follow occurrences of "D4". This continuation memory is made of four different associative tables. From left to right, they are related, respectively, to pitch, inter-pitch interval, beat and rhythm. For instance, the pitch table shows occurrences of "D4" continued by either F♯4 or D4.
- This associative table allows us to detect that several occurrences of "D4" are followed by F♯4. This triggers the detection of a new pattern, that we could describe simply for the moment as "D4 F♯4". It is a child of the node related to pattern "D4" and is displayed with a grey disk in Fig. 11.4. We will see in Sect. 11.2.6.2 that the pattern would be described more precisely as "D4 $\overrightarrow{+2\text{M},1}$ F♯4, 2". We will see in Sect. 11.3 that this pattern is actually not interesting, because it is already described by another more specific pattern, "D4, 1 $\overrightarrow{+2\text{M},1}$ F♯4, 2", shown also in the figure.
- That more specific pattern "D4, 1 $\overrightarrow{+2\text{M},1}$ F♯4, 2" also features a continuation memory, decomposed into associative tables related to parameters pitch ("P."), inter-pitch interval ("I.p.") and beat ("Bt"). The tables show that several occurrences of the pattern are followed by the same description, leading to an extension of the pattern into "D4, 1 $\overrightarrow{+2\text{M},1}$ F♯4, 2 $\overrightarrow{+2\text{m},1}$ A4, 3".

11.2.6.2 Description of the One-Pass Approach

The one-pass approach ensures an exhaustive heterogeneous pattern mining by proceeding as follows: for each successive note n_i in the piece of music to be analysed, the previous note n_{i-1} has already been linked to its corresponding patterns, forming a certain number of constructs ended by that note. Constructs attached to the previous note n_{i-1} are considered one after each other. Each construct is related to a particular pattern p.

For instance, in Fig. 11.5, let us consider note 5-2. The previous note, 5-1, ends two constructs: one occurrence of pattern "D4", and one occurrence of pattern "D4,1". Since the first construct is more general than the second construct, it is redundant and therefore indicated in grey.

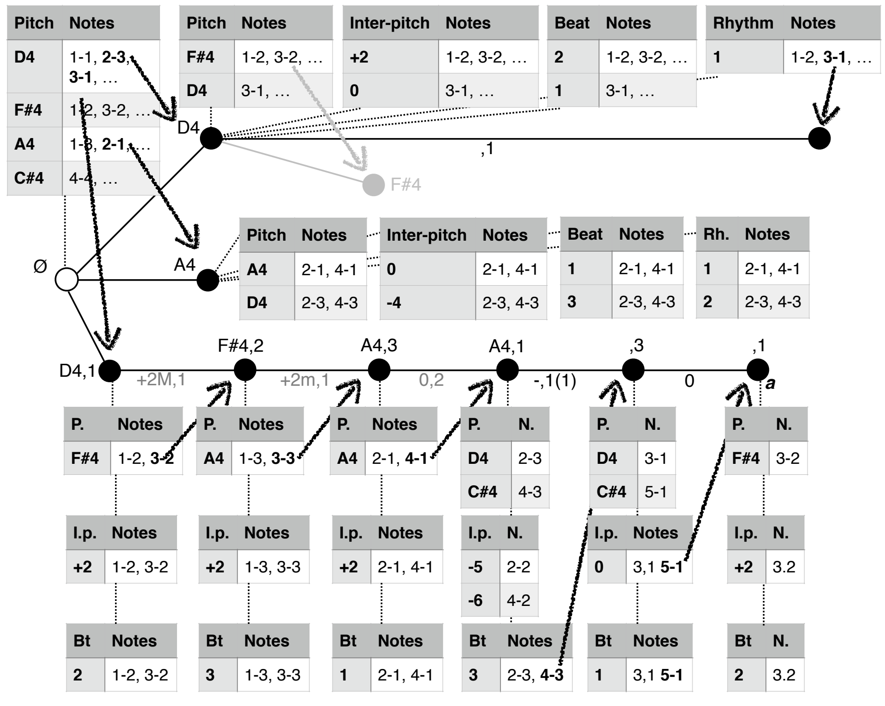

Fig. 11.4 Part of the pattern trie shown and described in Fig. 11.3. Associative tables are connected to their corresponding pattern nodes with dotted lines. In each table are stored the notes (right column) associated with each parameter value (left column). Notes memorized in the associative tables that trigger the inference of new patterns are shown in bold and arrows point to the new nodes. The node shown in grey is an example of a non-closed pattern that is actually not inferred by the model, as mentioned in Sect. 11.2.6.1 and further explained in Sect. 11.3.2

For each of these constructs, three tests are carried out:

1. **Pattern recognition** If the description of the new note n_i and of the interval between the two notes n_{i-1} and n_i is *entirely compatible*[5] with the description of one possible child c of pattern p, the construct is extended accordingly.
   For instance, in Fig. 11.5, the description of 5-2 and the interval between 5-1 and 5-2 corresponds exactly to the child "D4, 1 $\xrightarrow{+2\text{M},1}$ F♯4, 2" of pattern "D4,1". This pattern already exists because it was discovered while analysing note 3-2 (cf. below). Consequently, the construct is extended into an occurrence of that child.
2. **Pattern generalization** Else if that description is *partially compatible*, the *common description between the child's description and the description of the current note n_i and its interval with n_{i-1}* is computed. If it does not exist already,

[5] Expressions in italics are explained in more detail in Sect. 11.2.7.

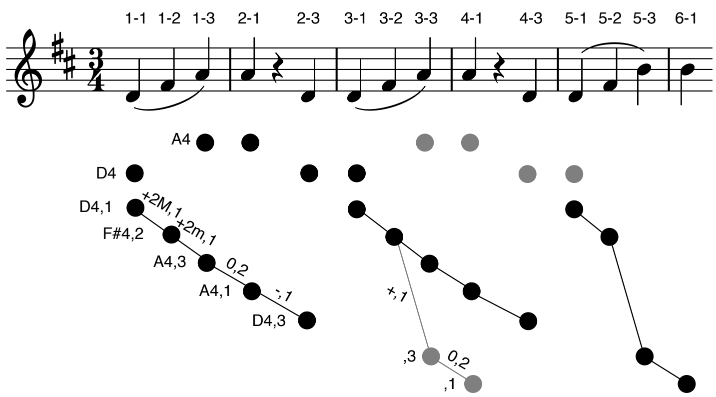

Fig. 11.5 Analysis of a variant of the beginning of the waltz melody in Fig. 11.2 (the pitches of the notes at 4-3 and 5-1 have been changed from C♯4 to D4). Constructs are shown by disks below the notes. In each line, all constructs are occurrences of the same pattern. The first 3 lines are related respectively to the elementary patterns "A4", "D4" and "D4,1". Other lines below correspond to successive prefixes of the two branches of a simple pattern tree

a new child of pattern p is created, related to the common description. The construct is extended accordingly.

For instance, in Fig. 11.5, the description of 5-3 and the interval between 5-2 and 5-3 corresponds only partially to the child "$\mathrm{D4},1\ \overrightarrow{+2\mathrm{M},1}\ \mathrm{F}\sharp 4,2\ \overrightarrow{+2\mathrm{m},1}\ \mathrm{A4},2$" of pattern "$\mathrm{D4},1\ \overrightarrow{+2\mathrm{M},1}\ \mathrm{F}\sharp 4,2$". The common description between the child and the current context defined by the succession 5-2 and 5-3 is an interval of ascending pitch in the same rhythmical description, i.e., "$\overrightarrow{+,1}\ ,3$". This leads therefore to the next pattern "$\mathrm{D4},1\ \overrightarrow{+2\mathrm{M},1}\ \mathrm{F}\sharp 4,2\ \overrightarrow{+,1}\ ,3$".

3. **Pattern discovery** The current context (i.e., the construct and the new note's description) is stored in the continuation memory of pattern p. If there already exists a previous context with identical value along a particular parameter (i.e., if a previous context is already stored for the same parameter value as key on one or several parametric dimensions), then the *common description between these two contexts*, d, is computed. If there does not exist a child of pattern p with that description d, then

   a. a new child c of pattern p is created, related to that description d;
   b. the current construct is extended into a new construct that is an occurrence of c; and
   c. the retrieved construct[6] is also extended into a new construct. Apart from this, the note that follows that extended retrieved construct is also stored in the continuation memory of the new pattern c.

[6] In the context of closed pattern mining, as developed in Sect. 11.3, there can actually be several retrieved constructs. Each construct is extended accordingly.

For instance, let us return to note 3-2 in the analysis in Fig. 11.5. The previous note is 3-1, with two associated constructs, which are occurrences of elementary patterns "D4" and "D4,1". The new note 3-2 is stored in the continuation memory of pattern "D4,1" (as in Fig. 11.4). In these associative tables, we detect that a previous context, the occurrence of "D4,1" at note 1-1 followed by note 1-2, already shares same description along at least one parameter (pitch, but also inter-pitch interval, beat and rhythm). The common description between these two contexts is "$\overrightarrow{+2\mathrm{M},1}$ F♯4, 2". This leads to the inference of a new child of pattern "D4,1", of description "D4, 1 $\overrightarrow{+2\mathrm{M},1}$ F♯4, 2". Its two occurrences at notes 1-2 and 3-2 are also created.

The three tests described above are also carried out in exactly the same way for the root, $\emptyset$, of the pattern tree:

1. **Pattern recognition** If the description of the new note n_i is *entirely compatible* with the description of one possible child c of the root, a new construct indicates that n_i is an occurrence of c. For instance, let us return to note 3-1 in the analysis in Fig. 11.5. Note 3-1 can be recognized as forming an occurrence of the elementary pattern "D4" that was created at an earlier stage (for note 2-3).
2. **Pattern generalization** Else if that description is *partially compatible*, the *common description between the child's description and the description of the current note* n_i is computed. If it does not exist already, a new child c of the root is created, related to the common description. A new construct indicates that n_i is an occurrence of c.
3. **Pattern discovery** The new note's description is stored in the root's continuation memory. If there exists already a previous note with identical value along a particular parameter (i.e., if a previous note is already stored for the same parameter value as key on one or several parametric dimensions), the *common description between these two notes*, d, is computed. If there does not exist a child of the root with that description d, then

   a. a new child c of the root is created, related to that description d;
   b. a new construct indicates that n_i is an occurrence of c; and
   c. another construct indicates that the retrieved note[7] is also an occurrence of c. Besides, the note that follows that construct is stored in the continuation memory of the new pattern c.

   For instance, let us return to note 2-3 in the analysis in Fig. 11.5. This note is stored in all the different associative tables (each one related to a different musical dimension) that form the continuation memory of the root ($\emptyset$) pattern (as in Fig. 11.4). In particular, the associative table related to pitch shows that there is already a previous context with the same pitch value D4 at note 1-1. The common description between these two contexts is "D4". The elementary pattern "D4" is created and its two occurrences at notes 1-1 and 2-3 are constructed.

[7] Here also, in the context of closed pattern mining, there can actually be several retrieved notes. Each of them forms a construct accordingly.

11.2.7 Comparing Musical Descriptions

To understand more precisely how musical descriptions are compared, it is important to distinguish between two types of description:

- The description *of a given note* (and its previous interval) in the score is specified entirely by the most specific parameters represented in bold letters in Fig. 11.1. All other parameters can be deduced directly from the most specific parameters.
- In contrast, a *pattern's individual description* is defined on particular parameters in that space, not necessarily the most specific ones.

For instance, as we saw in the previous example, pattern "D4" only specifies note pitch information. In motive *e* (see Figs. 11.2 and 11.3), "$, 1 >$" only indicates beat and articulation; whereas, "$\overrightarrow{-2, 4}\ , 2.$" specifies diatonic interval number, rhythmic value, beat position and articulation.

Recognizing a new occurrence of an already-discovered pattern requires testing whether or not the description of a given note (and its previous interval) in the score is *totally compatible* with a pattern's individual description. This implies the following.

- If the pattern description contains any specific note- or interval-related parametric description (along parametric dimensions indicated in black in Fig. 11.1: chromatic and diatonic pitch and pitch interval, metrical position and rhythmic value, articulation), the given note (with its previous interval) should feature the same parametric description.
- If the pattern description contains any more general parametric description (along parametric dimensions indicated in grey in Fig. 11.1: chromatic pitch class and pitch interval class, gross contour), the description of the given note (with its previous interval) should be congruent. For instance, if the pattern contains an ascending pitch contour, the interval description should also contain an ascending pitch contour.

Detecting a new pattern more general than another pre-existing pattern requires testing whether or not the description of a given note (and its previous interval) in the score is *partially compatible* with a pattern's individual description. This means that the two descriptions are compared along all the different individual parameters, and if there is any identity between the two descriptions along any parameter, they are partially compatible. In such cases, the *common description* between these two descriptions is the set of all these common parametric descriptions.

Finally, detecting a new pattern extension requires detecting an identity between two different contexts (notes and their previous intervals) in the score. Once any identity along any parameter is detected using the continuation memories, the *common description* between these two contexts is the set of all common parametric descriptions.

11.3 Richness and Concision of Pattern Description

At first sight, finding motivic patterns consists simply of extracting all possible sequence of descriptions in the piece of music. But it turns out that the problem is more complex, due to the proliferation of a large number of redundant structures (Cambouropoulos, 1998).

In order to reduce that structural proliferation at the pattern extraction phase, filtering heuristics are generally added that select a sub-class of the result based on global criteria such as pattern length, pattern frequency (within a piece or among different pieces), etc. (Cambouropoulos, 2006). Similarly, Conklin and Anagnostopoulou (2001) base the search for patterns on general statistical characteristics.

One limitation of these methods comes from the lack of selectivity of global criteria. Hence, by selecting longest patterns, one may discard short motives that listeners may nevertheless consider as highly relevant. Even more problematic is the fact that particular structural configurations (that will be studied in Sect. 11.3.3 and 11.3.4) lead to the generation of very long artefacts that do not necessarily correspond to the types of structure that were looked for in the first place.

The design of the model presented in this chapter was motivated by the aim of discovering the origins of the combinatorial explosion hampering the efficiency of computational models. The objective was to build a model as simple as possible, founded on heuristics ensuring a compact representation of the pattern configurations without significant loss of information, thanks to an adaptive and lossless selection of most specific descriptions. We found that pattern redundancy is based on two core issues: *closed patterns* (Sect. 11.3.1) and *cyclic patterns* (Sects. 11.3.3 and 11.3.4).

Another class of methods consists of cataloguing all possible pairs of sequential repetitions, and in clustering these pairs into pattern classes (Rolland, 1999) (see also Chap. 12, this volume). This paradigm allows for compact description, akin to the closed patterns selection that will be presented in Sect. 11.3.1, but no method in this category has yet been conceived of that performs exhaustive mining of heterogeneous patterns.

11.3.1 Closed Pattern Mining

When a pattern is repeated, all *more general* pattern representations it encompasses are repeated as well. A sequence of descriptions S_1 is *more general* than another sequence of descriptions S_2 if the sequence S_1 features less information than the sequence S_2, so that S_1 can be directly inferred from the description given by S_2. In such cases, we can also say that S_2 is *more specific* than S_1. More *general* patterns correspond to substrings[8] (prefixes, suffixes, prefixes of suffixes) of the pattern, but also to more general representations of such substrings where individual descriptions are replaced by more general ones as defined in Sect. 11.2.7. Figure 11.6 shows an

[8] More generally, subsequences can be considered as well. This will be studied in Sect. 11.4.

example of proliferation of general patterns. Examples of multiparametric pattern comparison in music are shown by arrows in Fig. 11.3.

Restricting the search to the most specific (or "maximal") patterns is excessively selective as it filters out potentially interesting patterns (such as CDE in Fig. 11.6), and would solely focus on large sequence repetitions.[9] Pattern redundancy can be filtered out without loss of information by taking into account both patterns' descriptions and the ways in which these patterns repeat in the sequences: the set of repetitions of a given pattern provides relevant structural information that should not be excluded from the analysis, unless that set of repetitions is *equivalent* to the set of repetitions of another more specific pattern. In our example, the set of two repetitions of pattern ABCDE is equivalent to the set of two repetitions of ABCDECDE: each occurrence of ABCDE is a construct that can be further extended to become an occurrence of ABCDECDE. On the other hand, the set of four repetitions of pattern CDE is not equivalent to the set of two repetitions of ABCDECDE, as there are more occurrences of CDE.

In order to obtain a compact lossless description, each possible set of pattern repetitions is subject to a *closure* operation that assigns its most specific pattern description, which is hence called a *closed pattern* (Pasquier et al., 1999). In our example, the most specific description of the set of two repetitions is ABCDECDE, and the most specific description of the pattern repeated four times is CDE.

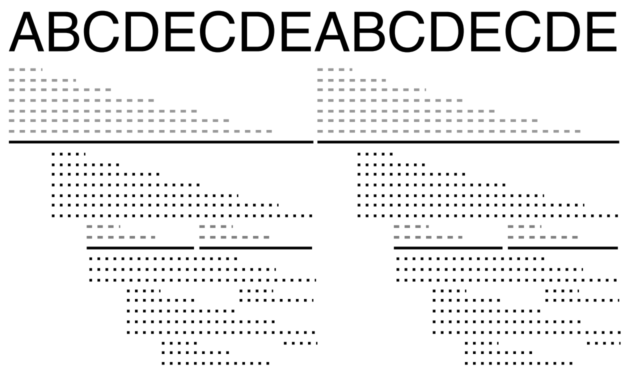

Fig. 11.6 Patterns found in a sequence of symbols. Below the sequence, each row represents a different pattern class with the occurrences aligned to the sequence. Solid lines correspond to closed patterns (the upper one is the maximal pattern), grey dashed lines to prefixes of closed patterns, and dotted lines to other non-closed patterns

[9] Note that the term "maximal" is used here in the way that it is commonly used in the data-mining literature to mean a pattern that is not included in any other patterns (see, e.g., Bayardo, 1998). This differs from the way "maximal"' is used in Meredith et al.'s (2002) concept of a "maximal translatable pattern", of which the pattern CDE is actually an example. See Chap. 13, this volume, for a definition of the latter concept.

Closed patterns are usually considered as preliminary information that is further processed for maximal or frequent pattern mining (Conklin and Bergeron, 2008; Pasquier et al., 1999). However, such global filtering results, once again, in the patterns that are finally produced as output being of limited practical use. On the other hand, by keeping the whole closed pattern lattice without further filtering, a detailed motivic analysis can be produced without combinatorial explosion. We call such an approach *exhaustive closed pattern mining*.

11.3.2 One-Pass Closed Pattern Mining

In previous studies on closed pattern mining in computer science (e.g., Pasquier et al., 1999; Wang et al., 2007), the pattern dictionary is constructed in a breadth-first fashion, while considering the whole document to be analysed (in our case, the piece of music). First, all closed patterns of length 1 are constructed, then all closed patterns of length 2, and so on. But such methods create a large number of hypothetical patterns that turn out to be non-closed once the pattern tree is further extended.

By adopting a one-pass approach, as presented in Sect. 11.2.6, there is no useless inference of non-closed patterns, and the closure test is far simpler. Indeed, in the one-pass approach, each pattern inference is made for a particular pattern occurrence candidate, for which we can infer the most specific description. This ensures that the obtained pattern is either closed, or is a prefix that will subsequently be extended into a longer closed pattern. Indeed, there cannot be more specific patterns that would contain that particular occurrence, apart from the possible extension of that occurrence into an occurrence of a longer pattern. Prefixes of closed patterns are the only type of possible non-closed pattern retained in the pattern tree. This is to ensure that the latter is a complete pattern prefix tree, which is easier to use for tracking patterns during the one-pass approach. It can also be shown that the one-pass approach uncovers the whole set of closed patterns.

When considering a given pattern candidate at a given point in the piece of music, we need to be already informed about the possible existence of more specific pattern occurrences at the same place. Hence, for a given note, patterns need to be extended *in decreasing order of specificity*, so that constructs[10] are built also in decreasing order of specificity.

For instance, in Fig. 11.7, when analysing the last note, E, there are two candidate patterns for extension, ABCD and CD. We first extend the most specific pattern to obtain ABCDE; then, when considering the more general pattern, CD, extension CDE is found as non-closed and is thus not inferred.

[10] A construct is an occurrence of a pattern (cf. Sect. 11.2.3).

Fig. 11.7 Closed patterns found in a sequence of symbols. The occurrence during which a pattern is discovered is shown in black. Dashed extensions indicate two possible pattern extensions when integrating the last note

11.3.2.1 Comparing Constructs

As has already been mentioned, the specificity of the one-pass approach for closed pattern mining is that general/specific comparisons are not made between patterns in general, but only between constructs that are aligned on a particular note in a musical sequence. We need to define more precisely this notion of comparing constructs. A construct *A* is more general than another construct *B*, if the set of notes connected to *A* is included in the set of notes connected to *B*, and if, for each of those notes, the underlying pattern's individual description in *A* is equal to or more general than the corresponding individual description in *B*.

For instance, in Fig. 11.5, a construct indicated with a grey disk is more general than another construct in black vertically aligned on the same note. For note 3-3, for instance, occurrences of "A4" and "$\mathrm{D4}, 1 \overrightarrow{+2\mathrm{M}, 1}\ \mathrm{F}\sharp 4, 2 \overrightarrow{+, 1}, 3$" are both more general than the occurrence of "$\mathrm{D4}, 1 \overrightarrow{+2\mathrm{M}, 1}\ \mathrm{F}\sharp 4, 2 \overrightarrow{+2\mathrm{m}, 1}\ \mathrm{A4}, 3$".

We can notice that it is possible to compare the extension of a construct to other constructs already inferred even if that construct extension is not actually inferred yet. More precisely, a construct *A* extended with a new individual description *d* is more general than another construct *C* already inferred if the two conditions below are verified:

- the construct *A* is more general than a prefix *B* of the construct *C*,
- the new description *d* is more general than the description of the extension from *B* to *C*.

This will be considered in the next paragraph as the comparison of a *candidate construct extension* with actual constructs.

11.3.2.2 Integration in the One-Pass Approach

The closed pattern filtering constraint can simply be integrated in step number 3 of the description of the one-pass approach given in Sect. 11.2.6.2, by replacing the condition "If there does not exist a child of pattern *p* with that description *d*" by the following condition: "If the candidate construct extension is not more general than another construct already inferred".

11.3.2.3 Motivic/Thematic Classes

Extending the exhaustive method developed in the previous section to the heterogeneous pattern paradigm allows us to describe all possible sequential repetitions along all parametric dimensions. This leads to very detailed pattern characterization, describing all common sequential descriptions between any pair of similar motives. However, a more synthetic analysis requires structuring the set of discovered patterns into motivic or thematic classes. Manual motivic taxonomy of these discovered patterns was demonstrated by Lartillot (2009). Lartillot (2014a) devised a method for the collection of all patterns belonging to the same motivic or thematic class into what is called a *paradigmatic sheaf*.

11.3.3 Cyclic Pattern Mining

When repetitions of a pattern are immediately successive, another combinatorial set of possible sequential repetitions can be logically inferred, as shown in Fig. 11.8. As each occurrence is followed by the beginning of a new occurrence, each pattern can be extended by a description that is identical to the first description of the pattern. This extension can be prolonged recursively, leading to a proliferation of patterns in an intertwined network (Cambouropoulos, 1998), which is detrimental to the effectiveness of the computation and the clarity of the results.

Avoiding overlapping between pattern occurrences (Cambouropoulos, 1998) allows for some of these patterns to be filtered out. But the effect of this heuristic is limited: as soon as the cyclicity is prolonged further, by repeating even further the periodic motive, those patterns that were filtered out will reappear.

This redundancy can be avoided by explicitly modelling the cyclic loop in the pattern representation. Once a successive repetition of a pattern has been detected, such as the 3-note pattern starting the musical example in Fig. 11.9, the two occurrences are *fused into one single construct*, and all the subsequent notes in the cyclic sequence are progressively added to that construct. This *cyclic construct* is first used to track the development of the new cycle (i.e., the third cycle, since there were already two cycles). The tracking of each new cycle is guided by a model describing

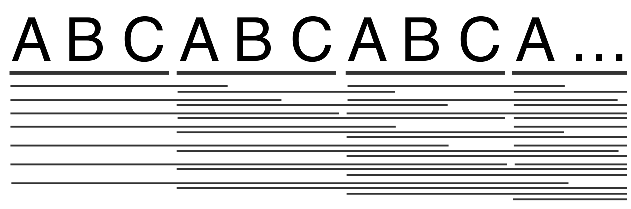

Fig. 11.8 Closed patterns found in a cyclic sequence of symbols. The occurrences of the pattern shown in thick lines do not overlap, whereas those shown in thin lines do

the expected sequence of musical parameters, i.e., the whole pattern *P* that is being repeated. Initially, for the third cycle, this model corresponds to the pattern that was repeated twice in the two first cycles.

- Every time a new cycle is being constructed, the growing cyclic construct keeps in memory the cycle model that describes a complete cycle, as well as a *cycle state* that indicates the current position of the last cycle of the cyclic construct.
- Every time the cycle state reaches the whole cycle model, a cycle has been completed, so that the cycle state points back to the beginning of the cycle model, which remains unchanged.
- If the current candidate extension is actually more general than the expected continuation given by the cycle state (such as the modification, at the beginning of bar 2 in Fig. 11.9, of the decreasing sixth interval, replaced by a more general decreasing contour), the cycle state is generalized accordingly. The cycle model, although now more specific than the actual cycle, is kept for the moment unchanged as it can still be used as a guiding point. When the cycle is completed, the cycle model can now be replaced by the new generalized cycle model given by the complete generalized cycle state.
- If at some point, the new note does not match at all the corresponding description in the model, the cyclic sequence is terminated.

This method allows us to track the cyclic development of repeated patterns, while avoiding the proliferation of patterns that was discussed above. Indeed, each of these problematic structures is detected as a candidate construct extension that is more

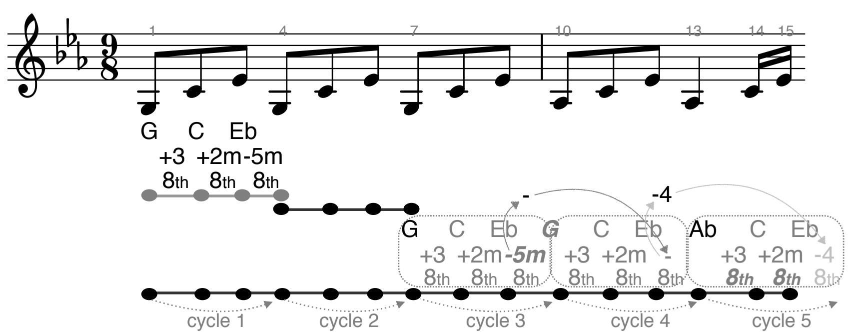

Fig. 11.9 Two successive repetitions of the three-note pattern that begins this sequence, characterized by a pitch sequence (G, C, E♭, and back to G), a pitch interval sequence (ascending perfect fourth (+3), ascending minor third (+2m) and descending minor sixth (−5m)), and a rhythmical sequence made of a succession of eighth notes. This successive repetition leads to the inference of a cyclic chain, indicated at the bottom of the figure. When this cycle is initially inferred, at note 7, the model of the cycle, labelled “cycle 3”, corresponds to the initial pattern description. At note 10, some descriptions expected by the model (indicated in bold italics) are not fulfilled, but a more general description is inferred (descending gross contour (−)). Consequently, the model for the next cycle (i.e., cycle 4) is generalized accordingly. At note 13, a new regularity is detected, due to the repetition of pitch A♭ and of descending perfect fifth (−4). Consequently, the model for cycle 5 is specialized accordingly

general than another construct already inferred. As such, following the closed pattern constraint defined in Sect. 11.3.2.2, the redundant structure is not inferred at all.

In the waltz melody in Fig. 11.2, the successive repetition of pattern *a* at the beginning of the piece leads to the creation of a cyclic construct related to that pattern. That is why the occurrences of the pattern are shown in a single horizontal line with tick marks showing the start of each new cycle.[11] From bar 5 on, this cycle is progressively altered: the initial specific pattern is not repeated exactly anymore, but a more general pattern continues to be repeated. The generalization of the initial cycle is indicated by a dotted line in Fig. 11.2.

11.3.4 Pattern Figure—Cyclic Background

Cyclic patterns engender further structural issues to be considered. A cyclic pattern can be extended into a non-cyclic pattern: this would happen when a cyclic pattern is repeated several times, and for several of those occurrences, the cycle at a specific phase is followed by the same description d. Consider, for example, 'ABCABCABD... ABCABCABCABD'.

Without the addition of particular mechanisms to handle this phenomenon, the acyclic pattern corresponding to that particular cycle phase is extended with that description d. In the example, D extends the acyclic patterns AB, or CAB, etc. But this information is insufficient, as it fails to represent the fact that the description d extends not only that particular acyclic pattern, but also a cyclic pattern. In our example, D can be considered as an extension of the cyclic pattern ABCA... at phase AB.

We can account for this type of situation by defining a new type of pattern extension related to cyclic contexts. In the example, as shown in Fig. 11.10, AB is extended by D in the particular cyclic context where AB is a phase of the cycle ABCA.... In this way, if a new cyclic repetition of ABCA... occurs with the same ending, it will be immediately recognized as a new occurrence of that extension of the phase AB.

Another difficulty arises when the description d that extends one particular phase of the cycle is compatible with its next phase. In other words, d does not lead to an interruption of the cycle, as in the previous case, but rather to a continuation of the cycle with the addition of new information. For instance, in the waltz melody in Fig. 11.2, patterns *b*, *c* and *d* could, in fact, be considered extensions of the cyclic pattern initiated by the successive repetitions of pattern *a*. The problem in this case is that this extended pattern could theoretically be further extended by the cyclical pattern, leading to a chaotic proliferation of patterns. For instance, pattern *d*, which simply represents the repetition of pitch B4, could in principle be extended by the next descriptions in the cycle: decreasing interval, unison, ascending interval, and

[11] We should note that the tick marks do not necessarily correspond to segmentation points, but are used here only as a way to represent the locations of an arbitrarily chosen phase in each cycle. Hypothetical segmentation points can be inferred based on other strategies, such as local grouping, as discussed in Sect. 11.4.2.

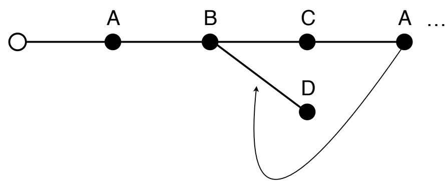

Fig. 11.10 Pattern tree with cyclic extension: AB is extended by D in the context where AB is in fact a phase of the cycle ABCA...

so on, without any clear limit in this pattern extension. If the cycle is repeated many times, this would engender very long and uninformative patterns.

This difficulty can be solved by using a heuristic, inspired by the 'figure-ground' principle: the additional information given by the description d can be considered as an extra layer on top of the cyclic pattern description. Further extensions of the pattern by descriptions richer than the cycle description lead to an extension of the extra layer, while keeping track of the background cycle. Whenever the next description corresponds merely to the cyclical description without additional information, this can be considered as a return of the background information without the extra-layer structure. Hence this would be perceived as a termination of the specific pattern on top of the cyclical background: in other words, the figure is not extended any more. The integration of this heuristic in the model allows us to suppress the chaotic proliferation of redundant structures.

11.4 Ornamented Pattern Mining

11.4.1 Ornamentation and Reduction

An ornamentation of a motive generally consists of the addition of one or several notes—the ornament—between some of the notes of the initial motive. Each repetition of a given motive can be ornamented in its own way, or not ornamented at all. In order to be able to detect the pattern repetition corresponding to the reduced motive, it is necessary to go beyond the "*surface*" of the actual music representation, i.e., beyond the mere succession of notes forming a single "*syntagmatic chain*" (de Saussure, 1916).

The detection of pattern repetition with tolerance for the addition, suppression or modification of individual descriptions has been previously addressed with the use of edit distance and dynamic programming (Rolland, 1999). This technique allows for a similarity distance to be computed between specific pattern occurrences, based on their local transformations. This leads to a network of numerical distances between occurrences, which cannot easily be turned into a clear categorization of pattern occurrences into pattern classes with precise descriptions. Moreover, the use of numerical distances requires a choice of similarity threshold, leading to a non-exhaustive search that is dependent on parametric choices. Another way to detect pattern repetition despite ornamentation is the "geometric approach", with patterns made of notes scattered anywhere in the score without constraint of sequentiality between the successive notes (Meredith et al., 2002). As mentioned in Sect. 11.3, similarity-based and geometric approaches have not yet been used for exhaustive mining of heterogeneous patterns.

We propose a solution to the problem of ornamentation reduction that is compatible with our vision of pattern discovery as an exhaustive search for exact identification along multiple parametric dimensions. The approach stems from the idea that, in a motive formed of notes that are not immediately successive in the syntagmatic surface, there is already implicitly a syntagmatic chaining between these successive notes in the motive that could be perceived as such. We see that beyond the initial syntagmatic chain at the "surface" emerges a more complex syntagmatic network. In the most complex case, we could theoretically imagine that all pairs of notes could be syntagmatically connected, but, perceptually speaking, we can understand that there should be constraints on the establishment of such syntagmatic connections.

It is then possible to find the whole set of closed patterns along all possible branches using the algorithm presented in the previous sections, appropriately generalized to the new framework. This will be shown in Sect. 11.4.3. But before that, we present one way to construct the syntagmatic network: first, a hierarchical local grouping structure is derived, then the resulting structures are used to determine the structurally more important notes and construct syntagmatic connections between them.

11.4.2 Syntagmatic Network Based on Local Grouping

There has been significant research around the concept of local segmentation, studying the emergence of structure related to the mere variability in the succession of musical parameters. These studies (Cambouropoulos, 2006; Lerdahl and Jackendoff, 1983; Tenney and Polansky, 1980) focus on the analysis of monodies, and model this structural phenomenon as a segmentation of the monody, which cuts the temporal span at particular instants, resulting in a linear succession of segments. In these approaches, the heuristics for segmentation are based on a mixture of several constraints related to what happens both before and after each candidate segmentation point, which leads to approximate and incomplete segmentation results. Here we present a simple approach that reveals a clear structural description and that can be explained with simple principles. The approach focuses on grouping instead of segmentation. In other words, what needs to be characterized is not the segmentation boundaries between notes, but rather the groups of notes that are progressively constructed. In this study, so far, this clustering mechanism has been applied only to the time domain, for grouping based on temporal proximity, following the second Grouping Preference Rule (GPR2) from Lerdahl and Jackendoff's (1983) *Generative Theory of Tonal*

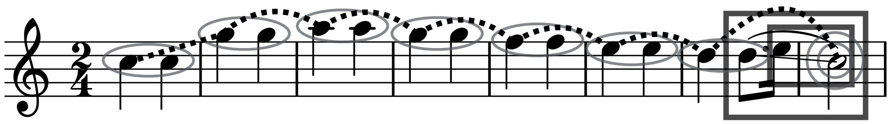

Fig. 11.11 Beginning of the right-hand part of the theme of Mozart's Variations on "*Ah, vous dirai-je maman*", K. 265/300e. Repeated notes are shown with ellipses. Local groups are represented with rectangles. The last note, playing the role of group head, is circled. Syntagmatic connections that are drawn between non-successive notes are shown with straight and curved lines. By traversing the syntagmatic chain represented by the dotted lines, we can recognize the successive notes of the underlying theme (in English: "*Twinkle, Twinkle, Little Star*")

Music (GTTM). Another type of grouping, based on change (GPR3) is currently under study.

In the time domain, local grouping can aggregate in a purely hierarchical fashion. For any succession of two notes n_1 and n_2, with inter-onset-interval (IOI) I, a local group is formed by taking all successive notes before n_1 and after n_2 as long as the IOI between successive notes does not exceed I. Smaller groups are related to short IOIs while larger groups that contain other groups are related to longer IOIs. For instance, the excerpt in Fig. 11.11 consists of a succession of quarter notes, except for the last three notes. The shortest note (a sixteenth note) induces a local group, joining that note with the final note. On a slightly larger scale, the last three notes are grouped together, because the longer internal IOI, a dotted eighth note, is shorter than the IOIs between the preceding quarter notes.

In Fig. 11.12, notes 2 and 3 form a local group because note 2 is a sixteenth note, which is shorter than the rhythmic values that precede and follow it. The same applies for notes 4 and 5. On a larger scale, because the quarter note (note 6) is longer than all the notes before, these first 6 notes also form a local group. Notes 7 to 9 are grouped together for the same reason: their successive IOIs do not exceed an eighth note while the IOIs before and after that group are larger (one quarter note).

By definition, a time-based local group terminates with a note that is followed by an IOI (before the next note) that is significantly longer than the IOIs between notes within the group. As such, the local group can be perceived as a phrase that terminates with a concluding note that has a higher structural importance. This hypothesis seems to offer some general interest, even though it might not be always valid—in particular, if notes within the group are accentuated. Following this observation, we formalize this hierarchy of notes in local groups by associating with each local group a main note, or "head", following Lerdahl and Jackendoff's (1983) Time-Span Reduction terminology, which would in the simple case be the last note of the group. Heads are circled in Fig. 11.12.

Based on this local grouping, two rules for inferring syntagmatic connections between distant notes are proposed as follows:

- Any local group head is connected to the note preceding the group.
  Thus, in Fig. 11.12, note 3, head of the local group starting with note 2, is connected to the note immediately before that group (note 1).

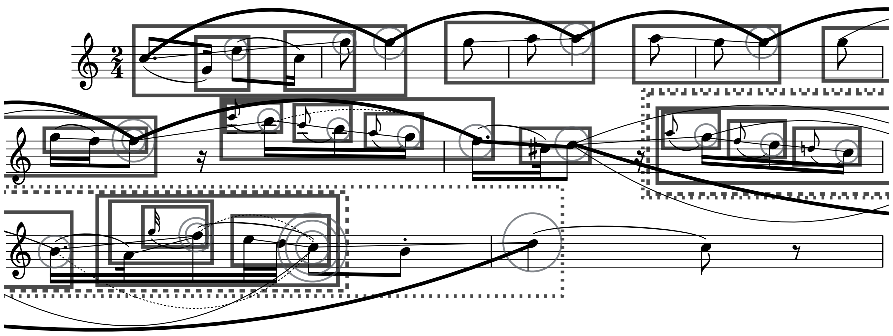

Fig. 11.12 Beginning of the right-hand part of Mozart's Variation XI on "*Ah, vous dirai-je maman*", K. 265/300e. Local groups are represented by rectangles, notes that play the role of group heads are circled. Syntagmatic connections that are drawn between non-successive notes are shown with straight and curved lines. By traversing the syntagmatic chain represented by bold lines, we can recognize the successive notes of the underlying theme (in English: "*Twinkle, Twinkle, Little Star*")

- Any local group head is connected to the first element in the group. If the first note of the group is not included in another internal local group, that first element is the first note of the group. Otherwise, it is the head of the largest first local group included in the local group.
  Thus, in Fig. 11.12, note 6 is connected to note 1, because note 1 is not part of any other local group. At the beginning of the second bar in stave 2, the note F is the head of a local group. The first note of that group, note 4 in stave 2, is included in another smaller local group whose head is the next note (note 5 in stave 2). Hence note F is connected to that note 5, as shown by the dotted curved line above the two notes.

A syntagmatic chain of notes with the same pitch forms a single *meta-note* whose time onset is defined to be that of the first note. This meta-note can be syntagmatically connected to the note that follows the meta-note. For instance, in Fig. 11.11, repetitions of the same pitch form a single meta-note, and successive meta-notes are syntagmatically connected, forming one syntagmatic chain with successive pitches C, G, A, G, F, E, D and C.

11.4.3 Closed Pattern Mining in a Syntagmatic Network

We can observe that a motive is a construct that is more general than an ornamentation of that motive. If a motive is always ornamented in the same way, the actual closed pattern is the ornamented motive, while the reduced motive is a non-closed pattern. If and only if the motive appears at least once without ornament, or with an ornamentation that does not share any commonality with the other ornamentations, does the motive become a closed pattern.

As before, closed pattern mining requires that constructs be inferred in decreasing order of specificity: a hypothetical construct extension should be compared to more specific constructs already inferred. In this more general context, this means that ornamented constructs should be discovered before their corresponding reduced constructs.

This implies the following ordering of operations. As before, the score is analysed chronologically, note by note. But for each note, n_i, there exist one or more syntagmatic connections from one or more previous notes $p_{i,j}$ to that note n_i. We consider these previous notes in reverse order, starting with the most recent note and working backwards in time. For each previous note, $p_{i,j}$, we try to extend the constructs it terminates with the syntagmatic connection between $p_{i,j}$ and n_i.

For each candidate construct extension, we need, as before, to consider all more specific constructs already inferred. The comparison between constructs is the same as in Sect. 11.3.2.1, because the proposed formalism can immediately be adapted to a syntagmatic network: the prefix B of the more specific construct C is not necessarily the immediate prefix of C, so that the description of the extension of B into C corresponds to a syntagmatic connection that is above the surface. This syntagmatic connection is therefore a chaining of several syntagmatic connections from the surface. The parametric description of that connection is hence a summation of the individual descriptions.

These rules allow for the detection of closed patterns along the different paths of the syntagmatic network. For instance, it allows for the identification of the repetition of the pitch sequence C, G, A, G, F, E, D, C that appears in the theme and variation XI shown in Figs. 11.11 and 11.12 (and similarly in other variations in the piece). To be more precise, a more detailed pattern C, G, G, A, A, G, G, F, E, D, C can be detected.

11.5 Evaluation

The system described above has been implemented in Matlab and is publicly available as part of the *MiningSuite* (Lartillot, 2015).

11.5.1 MIREX Task on Discovery of Repeated Themes & Sections

One version of the algorithm (*PatMinr* in *MiningSuite* 0.7.1) has been tested in the MIREX task on Discovery of Repeated Themes & Sections (Collins, 2014). The ground truth, called the Johannes Kepler University Patterns Test Database (JKUPTD), is based on motives and themes in analyses by Barlow and Morgenstern, Schoenberg and Bruhn, repeated sections marked explicitly in the score and supplementary annotations by Tom Collins. The ground truth is not made available in order to prevent developers from overfitting their algorithms to the test data.

Because *PatMinr* did not produce an analysis of one of the pieces in the test database for technical reasons,[12] the algorithms participating in this MIREX competition in 2014 were compared on the 4 remaining pieces of the test database. *PatMinr*'s results using the establishment metric are not particularly high: its establishment precision is .62 while Velarde and Meredith's (2014) method obtains better values for both versions they submitted: .67 for VM1 and .63 for VM2. *PatMinr*'s establishment recall is .56 and F1 is .5, which are both lower than for most of the other algorithms. *PatMinr* gives better results with the occurrence metric: occurrence precision ($c = .75$) is .88 whereas other algorithms' measures are between .48 to .75; occurrence recall is .76 whereas Velarde and Meredith's (2014) VM1 method gives a better result of .82; occurrence F1 is .81 whereas other algorithms' measures are between .34 and .6. This superiority in the occurrence metric means that the algorithm can find a large number of the occurrences of a given pattern, despite their transformations. This capability is related to the heterogeneous pattern representation that was presented in Sect. 11.2. Three-layer precision is .51, which is the highest value, also met by VM1; three-layer recall is also the best one at .494, which is very slightly better than 2 other algorithms: VM2 and Meredith's (2013) SIATECCOMPRESSSegment; three-layer F1 is .43, which is lower than Velarde and Meredith's (2014) method, with value .49 for VM1 and .45 for VM2.

11.5.2 Example on the MIREX Development Database

The JKU Patterns Development Database (JKUPDD) allows developers to try out and train algorithms to be used on the JKUPTD. Details of the analysis of one particular piece of music included in the JKUPDD, the Fugue in A minor from Book II of J. S. Bach's *Well-Tempered Clavier* (BWV 889) are presented by Lartillot (2014a). The ground truth consists of the first two bars of the third entry in the exposition, along with the other two voices that constitute this fugue (Bruhn, 1993). The third entry is chosen because it is the first entry where the subject and the two countersubjects are stated together. The fugue's subject is detected by *PatMinr* as one single motivic/thematic class, i.e., one complete paradigmatic sheaf, resulting from the bundling method presented in Sect. 11.3.2.3. All occurrences indicated in the ground truth are retrieved. The patterns forming this thematic class are longer than the two-bar motive indicated in the ground truth. The limitation of all subjects and counter-subjects in the musicological analysis to two bars stems from a theoretical understanding of fugue structure that cannot be automatically inferred from a direct analysis of the score.

The model used in this analysis did not integrate mechanisms for the reduction of ornamentation, as discussed in Sect. 11.4. The only melodic ornamentation appearing amid the occurrences of the fugue's subject is the addition of a passing note after

[12] The piece was not analysed entirely, because the version of the algorithm was producing redundant patterns in such a way that the computation took an excessive amount of time and had to be interrupted.

the first notes of some occurrences. This leads to a small error in the model's results, where the first actual note is not detected.

The thematic class related to the first countersubject is extracted in the same way, forming a paradigmatic sheaf. The pattern class given by the model corresponds mostly to the ground truth. Here again, some occurrences present similar extensions that are inventoried by the model, although they are ignored in the ground truth. One occurrence is not properly detected, once again due to the addition of passing notes.

The second countersubject is more problematic, because it is only 7 notes long. Several other longer patterns are found by the model, and the specificity of this countersubject is not grounded on characteristics purely related to pattern repetition. As mentioned above, the ground-truth selection of these three patterns is based on principles related to fugue rules, namely the synchronized iteration of the three patterns along the separate voices. It seems questionable to expect a general pattern mining algorithm non-specialized to a particular type of music to be able to infer this type of configuration.

The analysis offered by the computational model offers richer information than simply listing the occurrences of the subjects and countersubjects. It shows what musical descriptions characterize them, and details particular commonalities shared by occurrences of these subjects and countersubjects.

Various versions of the presented computational model have been used to analyse pieces of music of different genres: among others, an 18th-century French folk song "*Au clair de la lune*", or the beginning of the upper voice of the theme of the *Andante grazioso* of Wolfgang Amadeus Mozart's Piano Sonata No. 11 in A major, K. 331 (Lartillot, 2014b).

11.6 Discussion

11.6.1 About Complexity

The determination of the computational costs of the method in time and space is a complex question that is currently under investigation. The objective of the approach was to achieve an exhaustive analysis, while keeping the representation compact and the computation realistic. When evaluating the behaviour of the successive versions of the model, there has been an implicit concern that the number of structural inferences for each successive note analysed should not grow asymptotically. In other words, a linear complexity in input to output size relationship is desired. Closed pattern mining and elementary principles of cyclic pattern mining allow the output size to be reduced significantly without loss of structural information. Still, further reduction in time complexity would require enhancing the cyclic pattern mining mechanisms as well as adding long-term memory limitations, so that a certain type of structural complexity is discarded. Core hypotheses here are that the human cognitive system also carries out some kind of selection process in order to reduce the number of inferred structures, and that a similar type of selection might be modelled computationally. Assessing the validity of these claims will require significant further work.

11.6.2 Connection With Serial Pattern Learning

According to Restle's (1970) theory of serial pattern learning, learning a patterned sequence involves dividing it into subparts, where some subparts are generated by applying simple rules on other subparts. A pattern representation such as the one developed in this chapter allows a more general understanding of structure, in which sequential repetitions are not necessarily repeated successively: they can start anywhere in the sequence. Moreover, the proposed pattern representation models not only the final structure, but also the incremental perspective taking place while observing (i.e., listening to) a sequence.

When pattern occurrences do occur successively, this corresponds to cyclic patterns as studied in Sect. 11.3.3, and this corresponds to the "repeat" operation R in Restle's theory (or equivalently to the "same" operation S in Simon's (1972) model). In Restle's and Simon's theories this is represented by a mere succession of a given pattern. For the example shown in Fig. 11.8, this would be represented as $R^n(A+B+C)+A$ using Restle's formalism.[13] The cyclic pattern representation shows that each occurrence of the pattern is immediately followed by another occurrence, by showing that each occurrence is followed by the first note of a new occurrence. But this first note is not only the first note of an occurrence of that pattern, it is also the note that follows a previous complete occurrence. This is what is represented by the cyclic pattern: at each step in each occurrence of the pattern, we know that the occurrence being constructed actually follows a previous occurrence of the periodic pattern. Moreover, whereas Restle's and Simon's representations impose one single understanding of the structure as the successive repetition of one given period, the cyclic pattern representation, due to its symmetry, implicitly encompasses all possible rotations of that period—for example, ABCABCA can be understood not only as (ABC)(ABC)A, but also as A(BCA)(BCA).

11.6.3 Future Work

By controlling the factors of combinatorial redundancy, the approach proposed in this chapter allows for the generation of a detailed description of pattern repetitions. The approach is incremental, progressively analysing the musical sequence through one single pass. This allows the structural complexity to be controlled using simple heuristics. Besides the chronological approach, other techniques introduced in the model include the concept of motivic cyclicity as a way to filter out the proliferation of

[13] We note that prefixes, such as the final A which is a prefix of ABC, are not represented as such in Restle's representation.

redundant patterns, and the integration of heuristics based on the figure-ground Gestalt principle. We might hypothesize that the proposed model offers some explanation concerning the ways listeners actually perceive and understand music; however, further experiments need to be carried out in order to test this claim. The cognitive validation of the principles underlying the model presented in this chapter could form a topic for future work.

Gross contour needs to be constrained by factors related to local saliency and short-term memory (Lartillot, 2009). The integration of beat positions and articulation information, as shown in the examples developed in Sect. 11.2, is currently under development. The study of ornamentation based on a syntagmatic network needs further investigation: the solutions proposed in Sect. 11.4 only partially solve this problem—further mechanisms need to be taken into account.

The model presented in this chapter is restricted to the analysis of monodies, which is evidently a major limitation on the scope of application of the analytic method. Polyphonic scores could be analysed if they are decomposed into a set of monodies, which is possible for pieces like fugues that are in a contrapuntal style. Of course, the automation of voice extraction from polyphony remains a challenging problem. We are currently working on extending the model towards the detection of repeated monodic patterns within a general polyphony without prior specification of monodic lines. We also plan to generalize the model to the detection of patterns in chord sequences.

We are also currently investigating the modelling of patterns of patterns, where the meta-pattern consists of, for instance, various occurrences of the same pattern. Motivic analysis can also provide useful input for the analysis of metrical structure, leading to an interesting interdependency between these two types of analysis.

Acknowledgements This chapter benefited from valuable comments and suggestions from David Meredith and two anonymous reviewers. The work reported in this chapter was carried out within the European project, "Learning to Create" (Lrn2Cre8). The project Lrn2Cre8 acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859.

References

Bayardo, R. J. (1998). Efficiently mining long patterns from databases. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of data (SIGMOD '98)*, pages 85–93.

Bruhn, S. (1993). *J. S. Bach's Well-Tempered Clavier: In-Depth Analysis and Interpretation*. Mainer International.

Cambouropoulos, E. (1998). *Towards a general computational theory of musical structure*. PhD thesis, University of Edinburgh.

Cambouropoulos, E. (2006). Musical parallelism and melodic segmentation: A computational approach. *Music Perception*, 23(3):249–268.

Collins, T. (2014). MIREX 2014: Discovery of repeated themes and sections. http://www.music-ir.org/mirex.

Conklin, D. and Anagnostopoulou, C. (2001). Representation and discovery of multiple viewpoint patterns. In *Proceedings of the International Computer Music Conference (ICMC 2001)*, pages 479–485, Havana, Cuba.

Conklin, D. and Bergeron, M. (2008). Feature set patterns in music. *Computer Music Journal*, 32(1):60–70.

de Saussure, F. (1916). *Cours de linguistique générale*. Payot.

Lartillot, O. (2005). Multi-dimensional motivic pattern extraction founded on adaptive redundancy filtering. *Journal of New Music Research*, 34(4):375–393.

Lartillot, O. (2009). Taxonomic categorisation of motivic patterns. *Musicae Scientiae*, Discussion Forum 4(B):25–46.

Lartillot, O. (2014a). In-depth motivic analysis based on multiparametric closed pattern and cyclic sequence mining. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 361–366, Taipei, Taiwan.

Lartillot, O. (2014b). An integrative computational modelling of music structure apprehension. In *Proceedings of the International Conference on Music Perception and Cognition (ICMPC 2014)*, pages 80–86, Seoul, South Korea.

Lartillot, O. (2015). Miningsuite. https://code.google.com/p/miningsuite.

Lerdahl, F. and Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. MIT Press.

Meredith, D. (2013). COSIATEC and SIATECCompress: Pattern discovery by geometric compression. In *MIREX 2013 (Competition on Discovery of Repeated Themes & Sections)*. Available online at http://www.titanmusic.com/papers/public/MeredithMIREX2013.pdf.

Meredith, D., Lemström, K., and Wiggins, G. A. (2002). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321–345.

Nattiez, J.-J. (1990). *Music and discourse: Toward a semiology of music*. Princeton University Press.

Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999). Efficient mining of associative rules using closed itemset lattices. *Information Systems*, 24(1):25–46.

Restle, F. (1970). Theory of serial pattern learning: structural trees. *Psychological Review*, 77(6):481–495.

Rolland, P.-Y. (1999). Discovering patterns in musical sequences. *Journal of New Music Research*, 28(4):334–350.

Ruwet, N. (1987). Methods of analysis in musicology. *Music Analysis*, 6(1-2):11–36.

Simon, H. A. (1972). Complexity and the representation of patterned sequences of symbols. *Psychological Review*, 79(5):369–382.

Tenney, J. and Polansky, L. (1980). Temporal gestalt perception in music. *Journal of Music Theory*, 24(2):205–241.

Velarde, G. and Meredith, D. (2014). A wavelet-based approach to the discovery of themes and sections in monophonic melodies. In *Music Information Retrieval*

Evaluation Exchange (MIREX 2014), Competition on Discovery of Repeated Themes and Sections, Taipei, Taiwan.

Wang, J., Han, J., and Li., C. (2007). Frequent closed sequence mining without candidate maintenance. *IEEE Transactions on Knowledge and Data Engineering*, 19(8):1042–1056.