A MULTI-PARAMETRIC AND REDUNDANCY-FILTERING APPROACH TO PATTERN IDENTIFICATION

Olivier Lartillot
University of Jyväskylä
Department of Music
PL 35(A)
40014 University of Jyväskylä, Finland

ABSTRACT

This paper presents the principles of a new approach aimed at automatically discovering motivic patterns in monodies. It is shown that, for the results to agree with the listener's understanding, computer modelling needs to follow as closely as possible the strategies undertaken during the listening process. Motivic patterns, which may progressively follow different musical dimensions, are discovered through an adaptive incremental identification in a multi-dimensional parametric space. The combinatorial redundancy that would logically result from the model is carefully limited with the help of particular heuristics. In particular, a notion of specificity relation between pattern descriptions is defined, unifying suffix relation – between patterns – and inclusion relation – between the multi-parametric descriptions of patterns. This enables to discard redundant patterns, whose descriptions are less specific than other patterns and whose occurrences are included in the occurrences of the more specific patterns. Resulting analyzes come close to the structures actually perceived by the listener.

Keywords: motivic analysis, pattern discovery, melodic identification, redundancy filtering, music cognition.

1. GENERAL SPECIFICATIONS

1.1. The Key Role of Musical Patterns in MIR.

Musical structures may be decomposed along two general dimensions. On the one hand, temporal gaps and musical discontinuities (such as pitch leaps, or changes in intensity, timbre, etc.) induce the determination of boundaries [2] [11] [14]. On the other hand, similar contexts in one or several musical sequences may be associated one with the others, and be related to one single conceptual description called *pattern*. Once a pattern is inferred, the identification becomes *global*, since other occurrences of the pattern can be discovered throughout the whole musical sequence or inside an entire musical corpus. Contrary to local structures, global patterns offer hence a synthetic description of the musical sequences that can be used for MIR purposes.

1.2. An Adaptive Pattern Identification

In opposition to similarity-based paradigm [4] [13], cognitive studies [7] have suggested that music identification relies on *exact* identification along multiple parametric dimensions, such as pitch, contour and rhythm. The cognitive and computational approaches to melodic identification along 'multiple viewpoints' [3] always consider each possible musical dimension separately. Resulting patterns are either rhythmic, melodic, or melodico-rhythmic, and melodic patterns result either from pitch, scale, or contour identifications. However, it seems that heterogeneous patterns may be constructed through a progressive identification along *different* musical dimensions. For instance, the pattern represented in the first line of Figure 4 consists of three notes of same pitch and rhythmic value and a fourth note of lower pitch. For such patterns to be discovered, all possible musical dimensions have to be considered during each phase of the progressive construction, and relevant viewpoints have to be selected in an adaptive way. A computational solution to this core problem is described in this paper.

1.3. An Incremental Pattern Construction

Patterns are usually discovered following two different possible strategies. In a first approach, pair-wise comparisons are made between templates, that are selectively extracted from the musical sequence [4] or that consists of all possible sub-strings within a defined range of length [13]. Once templates are identical or sufficiently similar, they are considered as occurrences of a pattern. In this way, only patterns that are included in this pre-defined set of templates – particularly, patterns of a limited size – will be discovered.

Alternatively, pattern occurrences are discovered through a progressive construction directly from the musical sequence [2] [5] [6]. First, patterns of two notes are discovered. Then the next notes following their occurrences are compared. Identifications among these continuations lead to extensions into patterns of three notes, and so on. Patterns of unlimited size may then be discovered in an optimal way, since only the necessary comparisons are made.

1.4. A Non-Selective Approach

In most current approaches, the automated pattern discovery mechanism produces a large amount of patterns that does not present any interest as such. The result need then to be reduced through additional filtering mechanisms, which select patterns featuring a good score along particular criteria. Such a global post-filtering process prevents a thorough analysis of the musical pieces. In our approach, we will try to avoid this filtering by insuring the pertinence of the pattern discovery process itself. For this purpose, we will show in particular the necessity of an automated filtering of redundant patterns, such as suffixes.

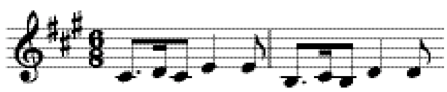
1.5. A Monodic Restriction

Some approaches [6] [13] take into account musical transformations such as note insertion, deletion, etc. Others [12] attempt to analyze polyphonic sequences. In our system, however, due to the complexity of the proposed paradigm, only monodic sequences will be considered in a first approach.

2. AN INCREMENTAL MULTIDIMENSIONAL MOTIVIC IDENTIFICATION

2.1. The Musical Dimensions

With each note may be associated different kinds of pitch values (see Figure 1). *Theoretical pitch values*, such as C#, stem from the existence of pitch *scales*, or tonality in particular. Each theoretical pitch value may then be also expressed as a *degree* on this scale. This scale degree can be represented by an integer between 0 and 7, where 0 is the tonic of the scale. In the scale degree may be included the octave position: With one particular tonic is associated value 0, with the tonic one octave higher value +7, etc. Diatonically transposed patterns – i.e. patterns that are translated along the scale degree dimension – can be identified along the *scale degree interval* – noted 's' in the remainder of the paper – that represents the scale degree difference between successive notes.



theoretical pitch	C# D C# E E B C# B D D
scale degree	2 3 2 4 4 1 2 1 3 3
scale degree interval (s)	+1 -1 +2 0 -3 +1 -1 +2 0
chromatic pitch	61 62 61 64 64 59 61 59 62 62
chromatic pitch interval (p)	+1 -1 +3 0 -5 +2 -2 +3 0
contour (c)	+ - + 0 - + - + 0
rhythmic value (r)	1.5 .5 1 2 1 1.5 .5 1 2 1

Figure 1. Description of a musical sequence following different musical dimensions. Repeated sequences of values, which form patterns, are squared. Are highlighted the dimensions integrated in our approach.

Alternatively, the pitch of each note may be expressed independently of any scale. Particularly convenient for that purpose is the *chromatic pitch* representation, which associates with each enharmonic pitch – say, each key of a piano keyboard – a position number. Following the MIDI standard, with middle C is associated the value 60, and the pitch value of each other note is computed in relation to its distance in semi-tones to middle C. Then chromatically transposed patterns – i.e. patterns that are translated along the chromatic pitch dimension – can be identified along the *inter-pitch* dimension, noted 'p', which is the chromatic pitch difference between successive notes. Finally, *contour* – noted 'c' – simply represents the sense of variation between successive notes: increasing (+), decreasing (-), or constant (0).

Finally, *rhythmic values* – noted 'r' – may be expressed by a rational number, indicating the quotient between the duration of each note and a given pulsation. For instance, as the rhythm of Figure 1 is ternary, value 1 is associated with quavers.

2.2. Incremental Pattern Construction

Patterns and their occurrences are discovered in an incremental and recursive way, but in the same time through a chronological scanning of the successive notes of the musical sequence. We will first explain the incremental construction of patterns – which generalizes Crochemore's approach [5] to a multi-dimensional space –, and will then describe its chronological adaptation.

2.2.1. Associative Memory

First, all the different parameters[1] related to each interval between successive notes are stored in *associative memories*. These memories are content-oriented, in such a way that a new interval induces a recall of all memorized intervals that are identical along one or several parameters. This can be modeled simply through hash-tables linked to each possible musical parameter. For instance, each new interval (say, the interval n6→n7 in Figure 2) is stored in a scale degree interval hash-table (named "scale interval"), at the index associated with its scale degree value (here: s = +1). All the memorized intervals featuring a same scale degree interval value are directly retrieved at the same index of the hash-table (here: the interval n1→n2).

2.2.2. Pattern Discovery

Once several intervals share a same identity along one or several parameters, a new pattern is created (here: node c pointed by the considered index of the scale interval hash-table). The description of the pattern is the list of identities (here, only s = +1), and the pattern class is the list of intervals that are considered as occurrences of this pattern (here: n1→n2 and n6→n7).

Following extensions of patterns follow the same principle. The interval that follows each occurrence of the pattern is stored in an associative memory related to the

[1] In our approach, each interval also contains a rhythmic dimension, which consists in the rhythmic value of the first note of the interval.

pattern (here: the ‘scale interval’ and ‘pitch interval’ hash-tables associated with the node c). In this way, whenever two following intervals (here: n2→n3 and n7→n8) share an identity along one or several parameters, a new extension of the pattern is created (here: d), and so on.

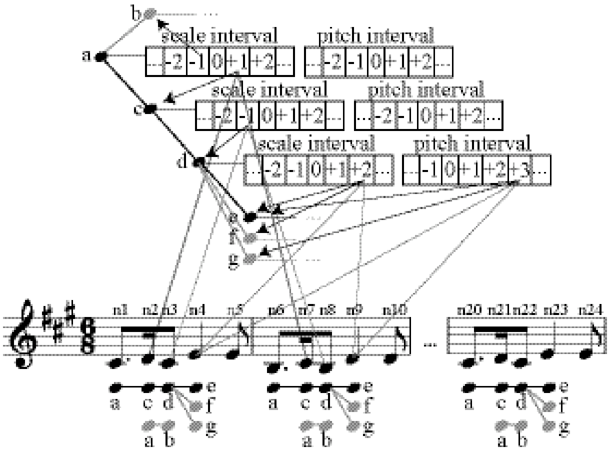


Figure 2. A musical sequence, some of its pattern occurrence trees (below), and the associated pattern tree (above), with some of the related associative memories. See the text for a detailed explanation of this figure.

2.3. Graph-Based Data Representation

2.3.1. Pattern Chains

When a pattern is progressively extended, its successive prefixes need to be stored. Each successive extension of a new occurrence of the pattern can be associated with each successive prefix of the pattern in an increasing order of length. For certain occurrences, this progressive pattern recognition is not complete and stops at one particular prefix. For these reasons, pattern may be represented as a chain of states – called *pattern chain* (PC) – featuring the successive prefixes. In Figure 2, the branch a→c→d→e, over the score, is a PC. Similarly, each pattern occurrence is also represented as a chain of states – called *pattern occurrence chain* (POC) – featuring the successive prefixes too. Each state of a POC is related to its corresponding PC. In Figure 2, each branch a→c→d→e under the score is a PO of the previously shown PC.

2.3.2. Pattern Trees

Now each state of a PC (for instance, d) can accept several different possible extensions (here: e, f and g). In this way, the set of all pattern classes forms a tree, called *pattern tree* (PT), and each PC is as a branch of the PT. This is what is represented over the score of Figure 2. Similarly, each state of a pattern occurrence can accept several different possible extensions. Hence the set of all pattern occurrences that are initiated by a same note (for instance: n1) forms a tree, called *pattern occurrence tree* (POT), and each POC initiated by the note n1 is a branch of the POT. In Figure 2, the POT initiated by n1 is represented underneath. The initial note n1 may be related to the root node (a) of the PT. Since all notes of the sequence can potentially initiate a pattern, they are all occurrences of this particular pattern a, called *note pattern*. For instance, under the POT initiated by n1 is a little POT initiated by n2 (a→b).

2.4. Chronological Pattern Construction

Now the incremental pattern construction has to be adapted to the chronological perception of notes founding the listening process. This necessity will be understood once we will consider, in the next section, the mechanisms of redundancy filtering. In a word, these mechanisms prevent the creation of particular pattern occurrences by taking into account the local context of each occurrence. If pattern occurrences are not filtered progressively, redundancy needs to be filtered by additional algorithms [6]. In the approach developed here, however, the analysis is so detailed that the simple pattern discovery process, because of the combinatorial redundancy, could not be completed without an integrated redundancy filter. That is why pattern occurrences need to be discovered chronologically.

Each new note that is heard (for instance, n9) is considered as an occurrence of the note pattern (a). In this way, new pattern occurrences may potentially be constructed from this note. Then, the interval n8→n9 between the previous note and current note is considered. Each occurrence that concludes the note n8 is successively considered (here: occurrences of d, b and a).

2.4.1. Chronological Pattern Discovery

The interval n8→n9 is memorized in the associative memory of the pattern d, b and a. As the interval n8→n9 is identified with the interval n3→n4 through the scale-interval and pitch-interval hash-tables associated with the pattern d, a new pattern e is inferred as an extension of d. The occurrence of d concluded by previous note n8 is extended into an occurrence of e concluded by current note n9. The occurrences associated with the memorized intervals (here, only n3→n4) are also extended. Patterns f and g are discovered in a similar way.

When the following note n10 will be considered, the new interval n9→n10 will be memorized in the associative memory of the patterns associated with the previous note n9 (e, f and g). However, the intervals n4→n5, on the contrary, could not be memorized in the same way. The memorization of these old intervals should therefore be done when the new patterns (e, f and g) are discovered.

2.4.2. Chronological Pattern Recognition

Consider now note n22, which concludes occurrences of d, b and a. The pattern d already accepts several extensions e, f and g. As the interval n22→n23 meets the description of extension e, the occurrence of d is simply extended into an occurrence of e concluded by note n23. The occurrences of patterns f and g are discovered in a similar way.

The incremental approach proposed here enables a multi-dimensional adaptive discovery of patterns. The use of hash-tables insures the computational efficiency of the pattern discovery process: thanks to the associative memory, remembering of old similar contexts does not need a search through the score.

3. REDUNDANCY FILTERING

3.1. Combinatorial Explosion

The pattern discovery system, as described in the previous section, shows important limitations. In particular, the number of discovered patterns is huge and the process easily enters into combinatorial explosion. This is due in particular to the redundancy of the pattern classes, which can be described along two relations.

3.1.1. Suffix Relation

When a pattern is discovered, all the possible suffixes of the patterns are also considered as patterns of their own. For instance, in Figure 2, b – which represents the identity

i2: s = -1

– is a suffix of d, which represents the sequence of identities i1→ i2, where

i1: s = +1.

Such redundant inferences should actually not be considered, unless the suffix appear alone in the musical sequence, without being a suffix of the longer pattern. This principle may be formalized with an equality relation between pattern classes. We defined the pattern classes as the set of occurrences of a pattern. The classes of pattern d and its suffix b will be considered as *equal* since each occurrence of b is a suffix of an occurrence of d. If, on the contrary, there exists occurrences of b that are not suffix of occurrences of d, then the pattern class of d would be considered as *included* in the pattern class of b.

3.1.2. Implication Relation

The second dimension of pattern redundancy stems from the notion of implication relations between pattern descriptions. Pattern e, for instance, is a succession of three identities i1→ i2→ i3, where

i3: s = +2 and p = +3.

Each identity may be compared to any other identity within any other pattern. A notion of *implication* between identities can now be defined, as a conjunction of two mechanisms.

Firstly, as the description of i4, where:

i4: s = +2,

for instance, consists in an element of the description of i3, then i4 may be considered as implied by i3.

Secondly, some parameters are direct consequences of other parameters. In particular, a contour value c = - is implied by an enharmonic pitch interval, for instance p = -2, or a scale degree interval value s = -1.

Both aspects can be unified into a single concept of identity implication. This leads us to the second dimension of pattern redundancy. Pattern f, for instance, is described by the succession of identities i1→ i2→ i4. As each successive identity of f is implied by the corresponding identity of same rank in e, then the whole description of f is implied by the whole description of e. If all occurrence of f are occurrence of e, f should not be considered as a pattern of its own. Else, this would produce a combinatorial set of redundant patterns.

3.1.3. Specificity Relation

Now suffix and implication relations can be unified, leading to a single *specificity relation*. The description h: i2→ i4, for instance, is less specific than the description e: i1→ i2→ i3, because h is an *implied suffix* of e. That is: the description of h is a suffix of the description f: i1→ i2→ i4, which is implied by the description of e.

3.2 Avoiding Redundant Description of Pattern Classes

Now the general principle ruling the pattern redundancy control may be stated as follows: If a pattern h is less specific than another pattern e, and if, in the same time, the pattern class of h is equal to the pattern class of e, then the pattern h, considered as *redundant*, should not be inferred at all.

However a pattern that is considered as redundant at one moment of the musical sequence may become non-redundant once it appears alone at a later stage of the sequence. The pattern would then be inferred, as well as all previous pattern occurrences whose existences were initially inhibited.

Put in another way, a pattern class could be described by different successions of identities, but only the most specific description should be explicitly considered. All the less specific descriptions are implicitly represented by the most specific description.

3.3. Incremental Redundancy Filtering

Now such redundancy filtering mechanism needs to be adapted to our incremental and chronological pattern discovery framework. As explained in section 2, patterns classes and occurrences are constructed through a progressive discovery of the successive intervals that constitute them.

We may remark that, when a pattern x is considered as a non-redundant suffix of another pattern y, its extension x’ may, on the contrary, become redundant. This happens when the pattern class of x’ is smaller than that of x and becomes equal to that of a more specific pattern y’ [10]. For this reason, the non-redundancy of a pattern should be checked at every phase of its extension.

Now the mechanism of incremental redundancy filtering will be explained through an example. Consider note n8 in Figure 2. The occurrence of pattern a, concluded by the previous note n7, is candidate for extension as an occurrence of a new pattern b. Consider the occurrence of pattern d concluded by current note n8.

Since the pattern class of this more specific pattern d is equal to the pattern class of b – one occurrence concluded by n3, and the other by n8 –, then pattern b will actually not be inferred.

The trouble is, the more specific pattern d can be considered only if its occurrence concluded by n8 has already been discovered. First, for the new perceived note n8, all the pattern occurrences that are concluded by the previous note n7 should be considered in a decreasing order of specificity. Then, for each of these pattern occurrences, the possible extensions have to be considered in a decreasing order of specificity of their identities.

Thanks to the mechanism presented in this section, the general pattern discovery system offer a more compact and synthetic, but in the same time lossless, representation of the motivic dimension of musical pieces. Such reduction was necessary not only for the quality of the results, but also in order to limit the computational complexity of the process.

Moreover, when a pattern is repeated several times successively, lots of redundant implicit patterns could be logically discovered, leading to another combinatorial explosion [2] [10]. Although the listener may sometimes follow some of these redundant patterns, his or her perception more generally catches the successive repetitions of the simple period. This heuristics has been included in our model [10].

4. CURRENT RESULTS

4.1. Implementation

This model is developed as a library of *OpenMusic* [1] called *OMkanthus*, and will be integrated into *MIDItoolbox* [8]. The analysis is currently undertaken on rhythmically quantified MIDI files. Rhythmic values are directly computed with respect to a pre-defined tempo, and scale degree parameters through a straightforward correspondence between pitches values and scale degrees, knowing the tonality. Contour, although theoretically included in our framework, is not taken into account in current analyses: its integration apparently needs a thorough modeling of short-term memory.

The results of the analysis can be displayed, as in Figures 3 and 4, in a score composed of a superposition of synchronous staves, each different stave representing the occurrences of a different pattern. Alternatively, the patterns progressively inferred by the model during the incremental analysis of the score can be listed. This enables to trace the analysis process, to understand the strategies undertaken, and to find the reasons of the possible unexpected behaviors.

4.2. Some Results

4.2.1. Beginning of Mozart Sonata in A, K 331.

Figure 3 presents the resulting analysis of the beginning of the upper voice of the first movement of Mozart *Sonata* in *A*, K 331. The first pattern is the main phrase of the main theme that appears twice as antecedent and consequent. The last notes of each phrase are not identified because of the little rhythmic variation that our system cannot abstract for the moment. The second pattern is the little motive repeated twice, with diatonic transposition, in the first pattern. The third pattern is purely rhythmic, and is repeated successively in the main phrase. The fourth pattern is a very short melodic phrase with two distant occurrences, whose actual perception by the listener may be questioned. Two non-pertinent patterns have also been discovered, that result from bad behaviors of the modeling.

4.2.2. Beginning of Beethoven's Fifth Symphony.

Figure 4 presents the analysis of the monodic reduction of the beginning of Beethoven's *Fifth Symphony*. The first line represents the different occurrences of the famous 4-note pattern. The second pattern is a melodico-rhythmic phrase that aggregates three successive occurrences of the 4-note pattern. The third pattern is a specification of the 4-note pattern featuring a third interval between the two last notes and a long last rhythmic value. The fourth pattern is another specification featuring a major third interval. The fifth pattern is an extension of the 4-note pattern, concluded by an ascending fourth interval.

Previous pattern discovery systems cannot offer this kind of analysis, although evident for the listeners. They would indeed include a numerous set of redundant patterns such as suffixes or redundant extensions. This shows the necessity of mechanisms of redundancy filtering such as those proposed in this paper.

4.3. Discussion and Future Works

This study has shown that the musical patterns actually discovered by the listeners cannot be reduced to simple mathematical definitions. The actual complex strategies undertaken during the listening process need to be modeled as carefully as possible.

The computational complexity of the model is not easy to assess. Indeed, due to the complex interdependencies between the different mechanisms, the behavior of the model varies extremely with regard to the musical material.

Thanks to this perceptive mimicry, the model offers promising results. Yet bad behaviors need to be controlled, and a large scope of musical expression – such as polyphony – has not been taken into account yet. Some assessments of the formal definition of patterns, though, have been attempted [9].

Acknowledgements

This study has been initiated during my PhD supervised by Emmanuel Saint-James (Paris 6 University) and Gérard Assayag (Ircam), and is now carried out in the Music Cognition Group supervised by Petri Toiviainen at the University of Jyväskylä.

5. REFERENCES

[1] Assayag, G. et al. "Computer Assisted Composition at Ircam: From Patchwork to Openmusic", *Computer Music Journal*, 23(3), 1999.

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

[3] Conklin, D., and C. Anagnostopoulou. "Representation and Discovery of Multiple Viewpoint Patterns", *Proceedings of the International Computer Music Conference*, San Francisco, USA, 2001.

[4] Cope, D. *Computer and Musical Style*. Oxford University Press., 1991.

[5] Crochemore, M. "An optimal algorithm for computing the repetitions in a word", *Information Processing Letters*, 12(3), 1981.

[6] Dannenberg, R. and N. Hu. "Pattern Discovery Techniques for Music Audio." *Proceedings of the International Conference on Music Information Retrieval*, Paris, France, 2002.

[7] Dowling, W.J., and D.L. Harwood. *Music Cognition*. Academic Press, London, 1986.

[8] Eerola, T., and P. Toiviainen. "MIR In Matlab: The MIDI Toolbox." *Proceedings of the International Conference on Music Information Retrieval*, Barcelona, Spain, 2004.

[9] Lartillot, O., and E. Saint-James. "Automating Motivic Analysis through the Application of Perceptual Rules", *Music Query: Methods, Strategies, and User Studies* (*Computing in Musicology* 13). MIT Press, 2004.

[10] Lartillot, O. "A Musical Pattern Discovery System Founded on a Modeling of Listening Strategies", *Computer Music Journal*, 28(3), 2004.

[11] Lerdahl, F., and R. Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, 1983.

[12] Meredith, D., K. Lemström and G. Wiggins. "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music" *Journal of New Music Research*, 31(4), 2002.

[13] Rolland, P.-Y. "Discovering Patterns in Musical Sequences", *Journal of New Music Research*, 28(4), 1999.

[14] Temperley, D. *The Cognition of Basic Musical Structures*. MIT Press, 1988.

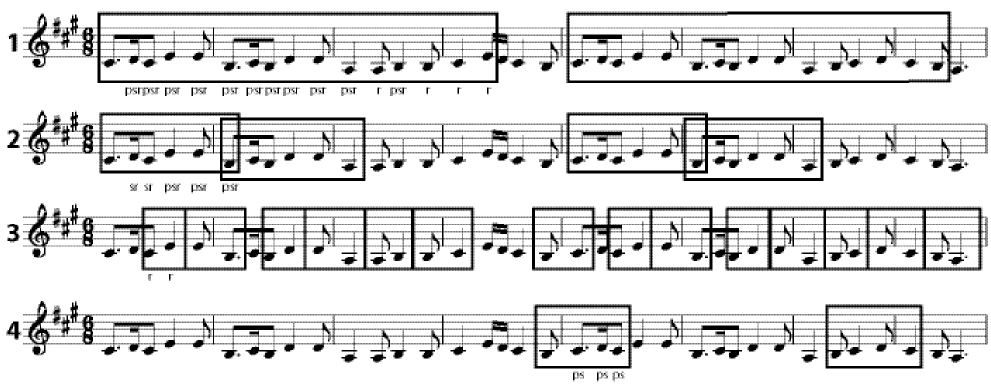



Figure 3. Analysis of the upper voice of the beginning of the first movement of Mozart *Sonata* in *A*, K 331. Each different line shows the occurrences, within the same melody, of a different pattern. The successive interval parameters taking part in the description of each pattern are indicated below each first occurrence, under the note ending each considered interval, and where ‘p’ means pitch, ‘s’ scale degree and ‘r’ rhythm.

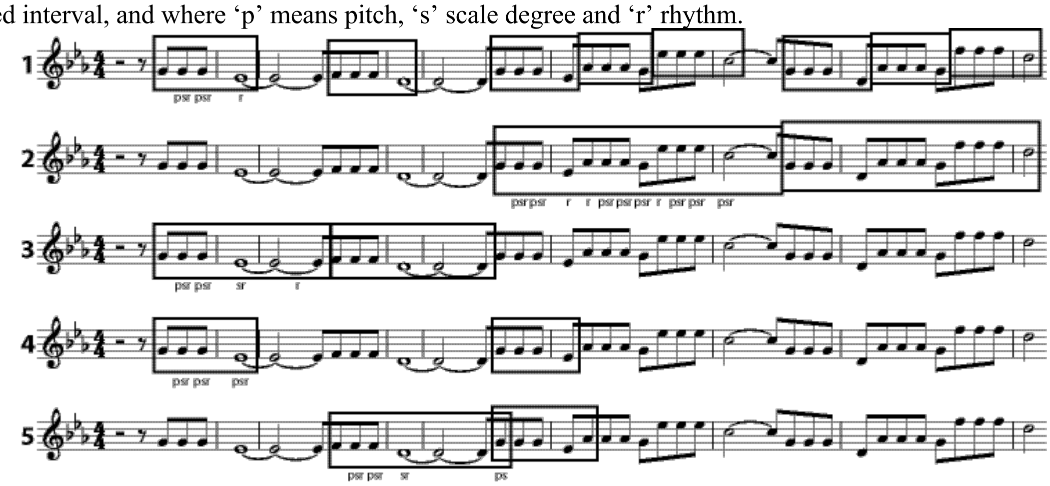


Figure 4. Analysis of the beginning of Beethoven’s *Fifth Symphony*, in the same representation that in Figure 3.