Spatial Programming for Music Representation and Analysis

Louis Bigo
IRCAM – CNRS
LACL – Université Paris Est Créteil
Paris, Fance
louis.bigo@ircam.fr

Antoine Spicher
LACL
Université Paris Est Créteil
Créteil, France
antoine.spicher@u-pec.fr

Olivier Michel
LACL
Université Paris Est Créteil
Créteil, France
olivier.michel@u-pec.fr

***Abstract*—In this paper, we show how some musical objects and some algorithmic problems arising in musical theory can be rephrased in spatial terms. This leads to the development of meaningful spatial representations and of efficient spatial programs. The corresponding developments have been implemented in MGS, a rule-based spatial programming language.**

***Keywords*-spatial programming; musical theory; MGS programming language; simplicial complex; GBF;**

I. Introduction

Spatial computing has proven to be a fruitful paradigm for the (re-)design of algorithms tackling problems embedded in space or having a spatial extension. While music can be seen according to various viewpoints, we propose in this paper a few studies of paradigmatic theoretical music problems from a spatial computing perspective.

Musical theory has received much interest by mathematicians that have found a field to develop algebraic methods to solve, in a very elegant way, problems of enumeration and classification of musical structures. Indeed, the algebraic nature of many musical formalizations has been very early assessed: from the equal temperament to combinatorial properties of the integral serialism and modal techniques. However, the topological properties of those objects have been rarely considered (nevertheless, see [1]).

Spatial computing distinguishes spaces in computations either as a *resource* or a *result*. In the following, we propose to illustrate each of these two points of view in the field of theoretical music. This paper is organized as follows: Section II provides a breve introduction to the MGS programming language that is used to illustrate examples given in this paper; Section III describes a spatial representation of harmonic relations based on a hexagonal lattice and investigates how it can be modified to fit better the harmonic structure of a musical piece; Section IV proposes a self-assembly algorithm for the construction of a topological representation of a chord series; the last section summarizes our approach based on a spatial point of view on musical theory and proposes future work.

II. A Very Short Introduction to MGS

MGS is an experimental programming language (see [2], [3]) investigating the use of rules to compute with spatial data structures. MGS concepts rely on well established notions in algebraic topology [4] and have been validated through applications in autonomic computing and in the modeling of complex dynamical systems.

A. Topological Collections

In MGS, all data structures are unified under the notion of *topological collection* [5]. A topological collection is a *cellular complex* labeled with arbitrary values. The cellular complex acts as a container and the values as the elements of the data structure.

A *cellular complex* is a space built by gluing together more elementary spaces called *cells*. A cell is the abstraction of a space with some *dimension*: a cell of dimension 0, also called 0*-cell,* corresponds to a point, a 1-cell corresponds to a line or an edge, a 2-cell is a surface (e.g. a polygon), etc. For example, a graph is a cellular complex built only with 0-cells and 1-cells: hence it is a 1-dimensional complex. An other example is pictured on the left of Figure 1.

In a cellular complex, cells are organized following the *incidence relationship*. This relation (defining a partial order on the cells) is close to the intuitive notion of *boundary*: the boundary of a cell is the set of cells of smaller dimension that surround it in the complex. For instance, the boundary of an edge (i.e. a 1-cell) is composed of two vertices (i.e. two 0-cells). The boundary of a triangular surface are three 1-cells and three 0-cells. Two cells are said *incident* if one lies in the boundary of the other. Particularly, a p-cell c is a *face* of a q-

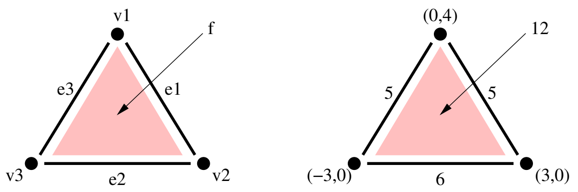


Figure 1. On the left, an example of cellular complex: it is composed of three 0-cells (v_1, v_2, v_3), of three 1-cells (e_1, e_2, e_3) and of a single 2-cells (f). The boundary of f is constituted of its incident cells v_1, v_2, v_3, e_1, e_2 and e_3. The three edges are the faces of f, and therefore f is a common coface of e_1, e_2 and e_3. On the right, data are associated with topological cells: positions with vertices, lengths with edges and area with f.

cell c', denoted $c < c'$, if they are incident and $p = q-1$. The cell c' is called a *coface* of c. See Figure 1 for an example.

More elaborated neighborhoods can be defined from the incidence relationship. In this article, we use the $<n,p>$-*neighborhood:* two cells c and c' are *$<p>$-neighbor* if there exists a p-cell that is commonly incident to c and c'. If the two cells are of dimension n, we say that they are $<n,p>$-*neighbor*. A *$<n,p>$-path* is a sequence of n-cells such that two consecutive cells are $<n,p>$-neighbor. For example, the usual notion of path in a graph corresponds to the notion of $<0,1>$-path: a sequence of vertices linked by edges.

Finally, a *topological collection* is a labeled cellular complex. An example of a topological collection is given on the right of Figure 1.

Types of collections differ from the specification of their underlying cellular complex. In the current implementation of the MGS language, usual data structures (sets, sequences, trees, arrays, etc.) are represented by vertex-labeled graphs: elements of the data structure are attached to the vertices and the edges represent the relative accessibility from one element to another in the data structure. MGS also handles more sophisticated spatial structures corresponding to arbitrary combination of cells of any dimensions. Sections III and IV uses two different types of MGS collections: *GBF* and *simplicial complexes*.

Group Based Fields (GBF). A GBF [6] is a topological collection that generalizes the usual notion of array, by labeling a cellular complex generated by a mathematical group. Such a group G is specified by a *finite presentation* (i.e. in terms of *generators* and *relations*) and can be represented by a 1-dimensional cellular complex called a *Cayley graph* [7]: each vertex in the graph corresponds to a group element; there exists an edge between two vertices h and h' if $h' = h + g$ where g is a generator of the presentation of G.

For example, the following presentation

```
< Fifth, MThird  ;  4*Fifth=MThird, 12*Fifth=0 >
```

corresponds to the cyclic Cayley graph of Figure 3. Two generators are defined: `Fifth` and `MThird` corresponding respectively to the diagonal and the downward directions on the figure. The structure of group provides the opposite directions of the generators. Therefore each vertex has four $<0,1>$-neighbors. In this paper, we only consider abelian groups; hence each pair of generators commutes. Here we have `Fifth+MThird=MThird+Fifth` meaning that starting from a vertex, the same vertex is reached by following either directions `Fifth` and then `MThird`, or directions `MThird` and then `Fifth`. Without anymore relations between generators, this presentation would generate an infinite square grid. The two additional relations allow us to make the graph periodic. Following 12 times direction `Fifth` is equivalent to not moving; Following 4 times direction `Fifth` is similar to following direction `MThird` one time.

Simplicial Complexes. A *simplicial complex* is a cellular complex where the topological cells are *simplexes*. A *p-simplex* is a p-cell that has exactly $p+1$ faces. For instance, a bounded

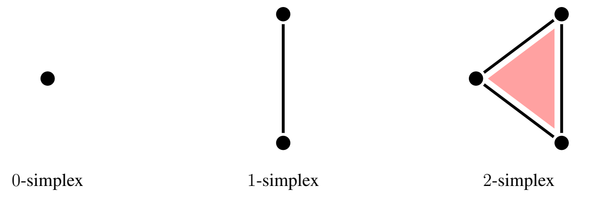


Figure 2. Some simplexes

line is a simplex but a hexahedron is not (a hexaedron is a 3-cell but has 6 faces). These objects are often represented geometrically as the convex hull of their vertices as shown in Figure 2 for p-simplexes with $p \in \{0,1,2\}$. A complex made only of simplexes is a *simplicial complex*. A graph or the Möbius strip of Figure 6 are examples of simplicial complexes.

B. Transformations

Topological collections are transformed using sets of rules called *transformations*. A rule is a pair *`pattern => expression`*. When a rule is applied on a topological collection, the sub-collections matching with the *`pattern`* are replaced by the topological collection computed by the evaluation of *`expression`*. There exists several ways to control the application of a set of rules on a collection but these details are not necessary for the comprehension of the work presented here. A formal presentation of the rewriting mechanism is given in [8]. We focus on the specification of patterns.

A *basic pattern* specifies a cell to be matched in the topological collection together with some (optional) guard. For example the expression `c / c = 3` matches a cell labeled with the value `3`. The guard is the predicate after the symbol `/`. The variable `c` can be used in the guard (and elsewhere in the rule) to denote the value of the matched cell or the cell itself, following the context (in case of ambiguity, the variable always denotes the associated value).

A pattern is a *composition* of basic patterns. There are three composition operators:

1) The composition denoted by a simple juxtaposition (e.g. "`x y`") does not constraint the arguments of the composition.
2) When two basic patterns are composed using a comma (e.g. "`x, y`"), it means that the cells matched by `x` and `y` must be p-neighbors. The dimension p depends on the topological collection or can be explicitly specified during the application of the transformation if needed. The default value for p is 1.
3) The last composition operator corresponds to the face operator: a pattern "`x < y`" (resp. "`x > y`") matches two cells x and y such that $x < y$ (resp. $x > y$).

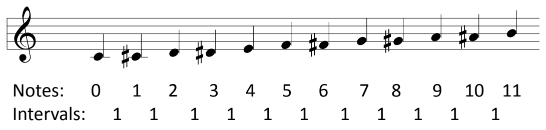
Patterns are *linear*: two distinct pattern variables always refer to two distinct cells.

III. Lattices of Notes as GBF

In this first application, we illustrate how a musical piece can be seen as a computation taking place on specific *resource* space. We focus here on the notion of *Tonnetz* and its use for the characterization of tonal music. We show that the topological collection of GBF, when carefully defined, makes it possible to describe chords progressions.

A. The Neo-Riemannian Theory

In standard Western music, the usual notation is based on the concept of *staff*. This notation makes spatially closer notes separated by smallest intervals. This emphasizing of the chromatism is well shown with the *chromatic scale*:


Notes: 0 1 2 3 4 5 6 7 8 9 10 11
Intervals: 1 1 1 1 1 1 1 1 1 1 1

In this score, each of the twelve notes (C, $C\sharp$, D, ..., A, $A\sharp$, B) is identified by an integer (starting from $C = 0$) such that the difference (modulo 12) between two notes exactly corresponds to the number of *semitones* – the smallest interval – constituting the interval between them. For example, between $E = 4$ and $G = 7$, the interval is a *minor third* corresponding to $7 - 4 = 3$ semitones.

From the tonal music point of view, this representation has two main drawbacks: (1) notes that "sound well" when played together are distant from each other (consider for example C and G), and (2) each note only has two neighbors (e.g. B and $C\sharp$ for the note C) while the dominant role of the perfect triads (three-note chords, see Section IV for more details) exhibits more than two notable harmonic neighbors.

The Neo-Riemannian representation (named after Hugo Riemann, a musicologist) has the ambition to show proximity between notes in terms of harmony instead of chromatism [9], [10]. The multi-neighborhood of a note leads to the idea of adding dimensions in the notes representation. For example, using a plane surface instead of the chromatic axis allows us to have more than two neighbors. L. Euler [11] was one of the first to propose such a representation, building his *Tonnetz* (see Figure 3), based on the two intervals of *fifth* (7 semitones) and *major third* (4 semitones).

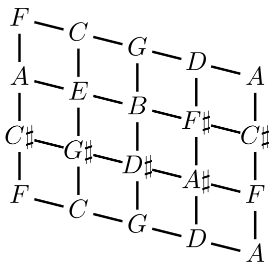


Figure 3. The *Tonnetz* after L. Euler (1739). Diagonal direction corresponds to the interval of fifth, downward direction to the major third.

B. Musical Notes as a Mathematical Group

The Neo-Riemannian approach consists in associating with each note n, a set of neighbors $S_n = \{n_1, n_2, \ldots\}$. We assume that this association

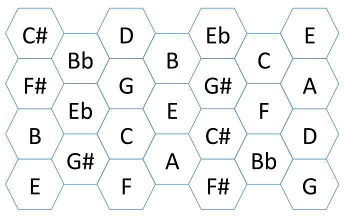


Figure 4. A Neo-Riemannian hexagonal lattice for the analysis of tonal music.

- is defined *up to a transposition*, that is moving all the notes up or down by a constant interval: for any interval i and any note n, we have $S_{n+i} = \{n_1 + i, n_2 + i, \ldots\}$;
- is *symmetric* (for instance, if C and G "sound well" together, G and C also do): $n \in S_{n'}$ implies $n' \in S_n$.

Thanks to the first assumption, the neighborhood has to be homogeneous for every note and can be characterized by a set $\mathcal{I} = \{i_1, i_2, \ldots\}$ of intervals such that for any note n, we have $S_n = \{n+i_1, n+i_2, \ldots\}$. Moreover the interval between two notes n_1 and n_2 is not in general the same than the interval between n_2 and n_1: for instance, the interval between C and G is a *fifth* (7 semitones) and the interval between G and C is a *fourth* (5 semitones). As a consequence, considering the second assumption, the neighborhood $\mathcal{I}$ has to gather intervals and their inverses: $i \in \mathcal{I}$ implies $(12-i) \in \mathcal{I}$. For example, in the *Tonnetz* of Figure 3, the neighborhood is specified by the intervals of *major third* (4 semitones) and *fifth* (7 semitones): $\mathcal{I} = \{4, 5, 7, 8\}$.

Knowing that the intervals are elements of $\mathbb{Z}_{12}$, a neighborhood $\mathcal{I} = \{i, \ldots, (12-i), \ldots\}$ can easily be equipped with a group structure whose presentation has the general form:

```
< i, ... ; all the relations in ℤ12 between i, ...>
```

The corresponding MGS GBF allow us to handle such Neo-Riemannian representations easily in a context of spatial programming.

This point of view generalizes many different existing musical representations. The *Tonnetz* is an example; its group presentation is given in Section II. The chromatic representation is also an instance corresponding to the presentation

```
< Semitone  ;  12*Semitone = 0 >
```

where only the semitone interval is taken into account. Looking back to the very motivation of the Neo-Riemannian approach for tonal music, one could choose the intervals of the fundamental triads for a representation that suits well the traditional harmony [12]:

- the *minor third* (3 semitones) and its inverse, the *major sixth* (9 semitones),
- the *major third* (4 semitones) and its inverse, the *augmented fifth* (8 semitones),
- the *fifth* (7 semitones) and its inverse, the *fourth* (5 semitones).

This leads to the following GBF specification:

```
< mThird, MThird, Fifth ;
    mThird + MThird = Fifth, 4 mThird = 0,
    3 MThird = 0, 12 Fifth = 0 >
```

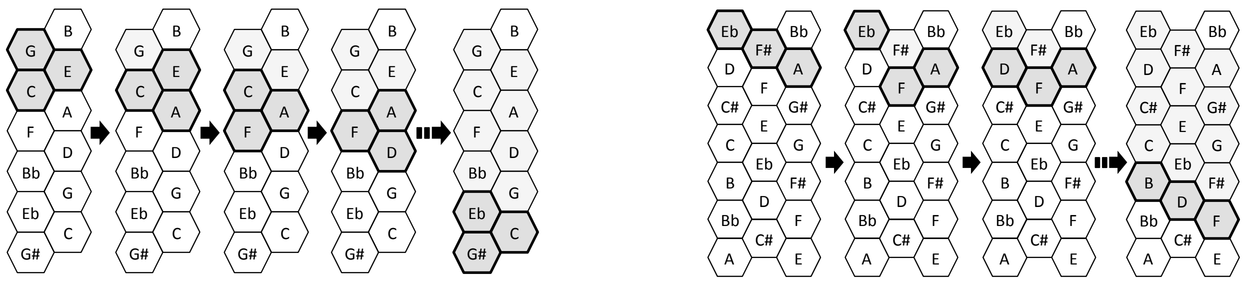


Figure 5. Representations on chord progressions: on the left the second movement of L. van Beethoven's 9th Symphony on a lattice generated by the intervals of *minor third*, *major third* and *fifth*, on the right the Prelude Op28 n°4 of F. Chopin on a lattice generated by the intervals of *minor third*, *major third* and *semitone*. Notes of the chords are in dark gray, notes previously played in the sequence are represented in light gray.

The corresponding Cayley graph, given Figure 4, is a hexagonal lattice (deeply studied by musicologists) that captures many aspects of tonal music. As an example, we observe in this lattice that the three notes C, E, G, composing a perfect major chord, form the most compact possible part of the space.

C. Generators Signature for a Musical Classification

A musical piece can be seen as a spatial behavior taking place on a spatial representation of notes. Let consider a chord progression. Each chord can be represented with a GBF collection where only the notes of the chord are labelled (for example with a boolean). Thus, the whole progression corresponds to an ordered sequence of collections. For example, Figure 5 on the left shows a chord progression included in the second movement of L. van Beethoven's 9th Symphony, using the previously introduced hexagonal representation. The hexagonal lattices have been voluntary unfolded vertically and truncated horizontally to underline the very regularity of this progression: at every step, one note only changes and the chord seems to spatially move downward.

Although the hexagonal representation spatially captures the construction rules used by L. van Beethoven to compose this progression, this is not always the case. For instance, the Prelude Op28 n°4 of F. Chopin also includes a chord progression where, like in the previous one, one note only is changing at every step. Nevertheless, the same experience does not make appear any spatial coordinated displacement in the evolution of the chords. In fact, contrary to the Beethoven's progression, the changing note follows here a chromatic progression. The chromatic interval being not part of the generators of the hexagonal lattice, it is understandable that the local evolutions do not appear clearly. It is obvious that the generators have to be carefully "*tuned*" to the studied piece. Here, a lattice generated by the *minor second* (1 semitone), the *minor third* and the *major third*, reveals the logical evolution of this chord progression (see Figure 5 on the right). Quite naturally arises the following question to the musicologist: *how can we interpret this change of generator of the fifth, which is very harmonic, to the generator of the minor second, which is dissonant?* We will not affirm that this change sets the Prelude of F. Chopin in a less tonal category, compared to the Symphony of L. van Beethoven. But it may be a sign announcing the evolution of music towards the emergence of atonality arriving at the 20th century.

This small experience reveals that the set of generators defining a network can be associated with the "*signature*" of a musical piece and maybe this concept of signature could be generalized to characterize a composer or a musical era. We are currently working on this aspect of the Neo-Riemannian approach.

IV. Chords Series Analysis

A mathematical analysis of music generally relies on the definition of a *model*, that is a mathematical object, whose specific properties describe faithfully some musical characteristics. In the previous section, we followed this approach by defining topological spaces used to exhibit some musical properties. In this section, we focus on the process used to build such a mathematical model of music. Thus, we consider here spaces as a *results* of a spatial computing. In particular, we describe a self-assembly mechanism used to define a spatial representation of a tonality based on triadic chords. Then, we propose to use this mechanism for the study of a tonality representation based on four-note chords and the analysis of a chord series to F. Chopin.

A. Tonality and Möbius Strip

A *tonality* is associated with a scale of notes, including one note corresponding to the *tonic* and one to the *mode* (*e.g.* minor or major). For example, the tonality of *C major* is characterized by the diatonic scale (C, D, E, F, G, A, B) starting from the tonic C and including the note E (the *major third* of C) defining the mode as major.

From such a scale, the tonality is defined by the covering of the notes using the seven *triadic chords*. In C major, these chords are:

$$I_C = \{C, E, G\} \quad II_C = \{D, F, A\} \quad III_C = \{E, G, B\}$$
$$IV_C = \{F, A, C\} \quad V_C = \{G, B, D\} \quad VI_C = \{A, C, E\}$$
$$VII_C = \{B, D, F\}$$

This definition of tonality may be interpreted geometrically. Each note of the scale is represented by a point. Then, if two notes belong to at least one of the chords defining the

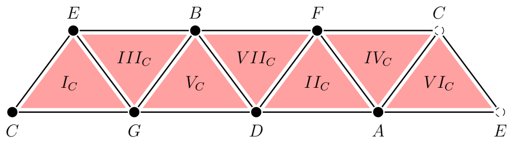


Figure 6. Representation of tonality C major.

tonality, their corresponding points are linked by an edge. Finally, the triadic chords are identified in this network by the triangular surfaces specified by the corresponding triplet of notes. Figure 6 shows this geometrical interpretation for the tonality C major.

This interpretation is fruitful in this example since the obtained mesh corresponds to a *Möbius strip* [13]. This particularly remarkable space was first exhibited by A. Schoenberg and has been deeply studied for music analysis. As an example, the boundary of the strip (a closed $<0,1>$-path) reveals the *cycle of fifths*, one of the main important structure in tonal harmony. More recently, the strip has been used as one of the ingredients for the elaboration of a mathematical model of music based on a functorial point of view [1].

B. Self-assembly of Chords

Since the Möbius strip is a good geometrical representation of the seven triadic degrees and their harmonic relations, we propose in this section a self-assembly algorithm for building systematically such a topological structure representing an arbitrary set of chords. The main idea is to let chords *react* with each other, where the reaction rule consists of the identification of common elements shared by chords. The study of the final object (obtained at the steady state when no more chord can react) can enlighten some properties of the chords.

Chord Representation. Chords are the basic elements to be assembled. Since a chord is a set of notes, we propose in the following to represent a chord by a *simplex*. In this context, a chord composed of n notes is then represented by a $(n-1)$-simplex. For example, chord I_C corresponds to the triangle $\{C,E,G\}$. It is interesting to note that the boundary of this triangle is composed of edges $\{C,E\}$, $\{E,G\}$ and $\{C,G\}$. These three 1-simplexes are directly associated with the fundamental intervals in tonal music of *thirds* and *fifth* discussed earlier. This explains why the Möbius strip captures so well the harmonic relations in tonal music.

Self-assembly Process. We propose to build the simplicial representation of a set of chords by using an accretive growing process [14]. The growth process is based on the identification of the simplexes boundaries. Nevertheless, this topological operation is not elementary since it holds in all dimensions. For example, merging chords I_C and III_C in the Möbius strip construction requires us to identify three elements: the vertices E and G, and the edge $\{E,G\}$.

A simple way of programming the identification is to apply recursively to a fixpoint the merge of topological cells that exactly have the same faces. In our example, this consists

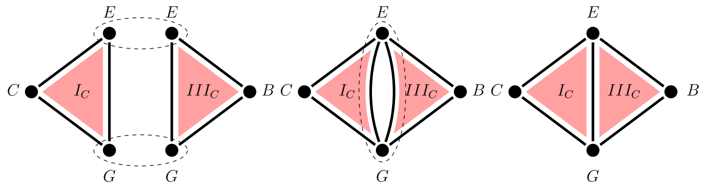


Figure 7. Identification of boundaries.

in merging nodes E and G from I_C and III_C, and then in merging the resulting edges $\{E,G\}$ as shown on Figure 7. The corresponding topological surgery can be expressed in the MGS syntax as follows:

```
transformation identification = {
  s1 s2 / (s1==s2 & faces(s1)==faces(s2))
  =>
  let c = new_cell (dim s1)
                   (faces s1)
                   (union (cofaces s1)
                          (cofaces s2))
  in  s1*c
}
```

The rule specifies that two elements `s1` and `s2`, having the same labels and the same faces in their boundaries, merge into a new element `c` (that has the union of the cofaces of `s1` and `s2` as cofaces) labeled as `s1` (which is also the label of `s2`).

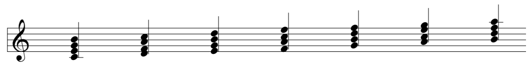
In Figure 7, the transformation `identification` is called twice. At the first application (from the left complex to the middle), vertices are identified. The two topological operations are made in parallel. At the second application (from the complex in the middle to the right), the two edges from E to G that share the same boundary, are merged. The resulting edge has the 2-simplexes I_C and III_C as cofaces, that correspond to the union of the cofaces of the merged edges. Finally (on the right), no more merge operation can take place and the fixpoint is reached.

As expected, the fixpoint application of `identification` to the triadic chords I_C, ..., VII_C builds the Möbius strip given in Figure 6.

C. Some Applications

In this last part, we apply our algorithm of geometrical chords representation for two musical purposes.

Four-note Chords Definition of Tonality. In some kinds of music like Jazz, chords are often richer than the triadic degrees. In this context, we propose to build the geometrical representation of a tonality in the same way we did before, except that we focus on the *four-note chords*. Adding a note to the seven degrees (*e.g.* I_C becomes $\{C,E,G,B\}$) increases the harmonic relations in the scale, allowing more complex arrangements. The seven four-note chords of the C major tonality are:



The geometrical representation of a four-note chord is a 3-simplex, that is a tetrahedron. The elaboration of the

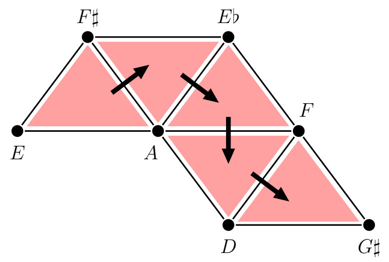


Figure 8. Simplical representation of F. Chopin's Prelude Op28 n°4.

simplicial complex associated with the seven chords is almost impossible by hand, making its topological study very difficult. Thanks to the dimension-free definition of transformation `identification`, the simplicial complex is automatically computed by MGS. After computing the Euler characteristic and the orientability coefficient of the complex, its topology appears to be a *torus*. On the contrary of the Möbius strip whose boundary exhibits the cycle of fifths (1 dimension), the topology of a torus suggests two dimensions: *what do these two dimensions correspond to?* Moreover, the torus is orientable and the Möbius strip is not: *knowing that by construction the Möbius strip of the triadic degrees is a sub-complex of the four-note chords torus, why is the non-orientability lost?* Such questions are currently investigated by musicologists at IRCAM using MGS to extract and build systematically the topological objects associated with various musical pieces.

Analysis of F. Chopin's Prelude Op28 n°4. In section III, we have proposed to embed the chords of the eight first measures of this prelude in a hexagonal grid given *a priori*. On the contrary, we propose here to build and study the specific geometrical representation of this progression of chords.

The whole simplicial complex associated with this chord series (*i.e.* obtained by the fixpoint iteration of transformation `identification` on the set of chords) cannot be easily visualized. Nevertheless, a sub-complex is given in Figure 8. A remarkable fact of this chord progression is that only one note is different between two consecutive chords. This property holds on the fourteen chords starting from the second one. Being composed of three-note chords, such a progression corresponds to a $<2,1>$-path in the associated simplicial complex: such a path is composed of 2-simplexes (the chords) connected by 1-simplexes (the two common notes). This path is partially presented by black arrows for the five first chords in Figure 8. We have enumerated all the possible $<2,1>$-paths with maximal length. It is interesting to note that there exist exactly 120 possible paths. Finally, among all these possibilities, F. Chopin chose the one with the smallest distance between chords. Indeed, the intervals between the different two notes of consecutive chords are a semitone for all transitions. This result confirms that the semitone is an important interval for this piece and is a good choice of generator for the hexagonal representation of section III. We are currently investigating more deeply the topology of this simplicial complex.

V. Conclusion and Future Work

We have shown in this paper, through the description and analysis of two problems in musical theory – from the GBF based Neo-Riemannian representation of music to the building of simplicial complexes associated with chord series in a musical piece by F. Chopin – that musical theory is an application area of first interest for spatial computing.

As seen in this very preliminary work, the building and processing of abstract spaces appears to be a key issue for musical analysis and we believe that the path taken in this paper can help improve and develop new tools to assist musicologists in their work. At the frontier of musical theory and computer science, many questions raised in this paper are currently being investigated at IRCAM with the help of the domain-specific spatial programming language MGS. A whole domain is still to be investigated: musical composition.

Acknowledgements. The authors would like to thank C. Agon, G. Assayag and M. Andreatta from the REPMUS team at IRCAM and J.-L. Giavitto from the LIS team, IBISC Lab. at University of Évry for endless fruitful discussions. This research is supported in part by the IRCAM and the University Paris Est-Créteil Val de Marne.

References

[1] G. Mazzola *et al.*, *The topos of music: geometric logic of concepts, theory, and performance*. Birkhäuser, 2002.
[2] J.-L. Giavitto and O. Michel, "Mgs: a rule-based programming language for complex objects and collections," in *Electronic Notes in Theoretical Computer Science*, M. van den Brand and R. Verma, Eds., vol. 59. Elsevier Science Publishers, 2001.
[3] J.-L. Giavitto, "Topological collections, transformations and their application to the modeling and the simulation of dynamical systems," in *Rewriting Technics and Applications (RTA'03)*, ser. LNCS, vol. LNCS 2706. Valencia: Springer, Jun. 2003, pp. 208 – 233.
[4] J. Munkres, *Elements of Algebraic Topology*. Addison-Wesley, 1984.
[5] J.-L. Giavitto and O. Michel, "Data structure as topological spaces," in *Proceedings of the 3nd International Conference on Unconventional Models of Computation UMC02*, vol. 2509, Himeji, Japan, Oct. 2002, pp. 137–150, lecture Notes in Computer Science.
[6] ——, "Declarative definition of group indexed data structures and approximation of their domains." in *Proceedings of the 3nd International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming (PPDP-01)*. ACM Press, Sep. 2001.
[7] R. Lyndon and P. Schupp, *Combinatorial group theory*. Springer Verlag, 2001, reprint of vol. 89 of Ergebnisse der Mathematik und ihrer Grenzgebiete, Springer (1977).
[8] A. Spicher, O. Michel, and J.-L. Giavitto, "Declarative declarative mesh subdivision using topological rewriting in mgs," in *Int. Conf. on Graph Transformations (ICGT) 2010*, ser. LNCS, vol. 6372, Sep. 2010, pp. 298–313.
[9] J.-M. Chouvel, "Analyser l'harmonie - aux frontières de la tonalité."
[10] ——, "Au-delà du système tonal."
[11] L. Euler, *Tentamen novae theoriae musicae ex certissismis harmoniae principiis dilucide expositae*. Saint Petersburg Academy, 1739.
[12] R. Cohn, "Neo-riemannian operations, parsimonious trichords, and their "tonnetz" representations," *Journal of Music Theory*, 1997.
[13] A. Schoenberg, *Harmonielehre*, U. Edition, Ed., 1911.
[14] J.-L. Giavitto and A. Spicher, *Systems Self-Assembly: multidisciplinary snapshots*. Elsevier, 2008, ch. Simulation of self-assembly processes using abstract reduction systems, pp. 199–223, doi:10.1016/S1571-0831(07)00009-3.