Spatial Programming for Musical Representation and Analysis

Louis Bigo & Antoine Spicher & Olivier Michel

mgs.spatial-computing.org

LACL, University Paris Est Créteil

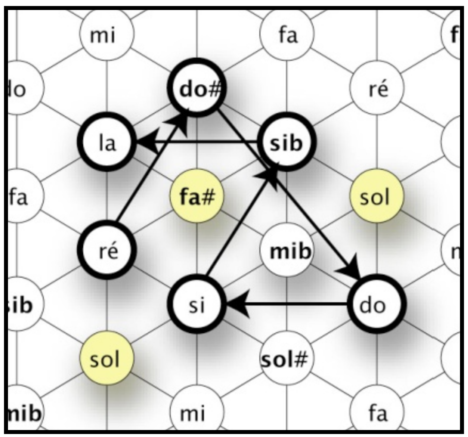
LIFO, University of Orléans – Mai 23-24, 2011

Context

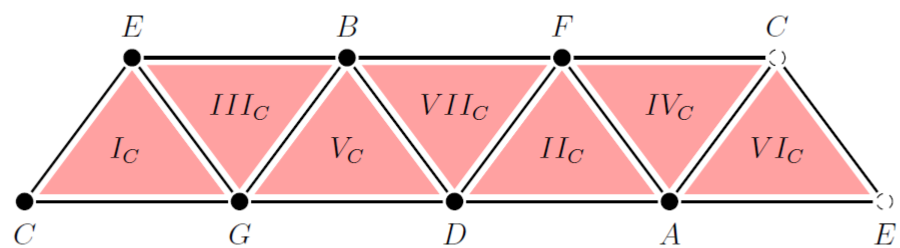
- Spatial computing
  - Compute in space
  - Compute space
- MGS: a programming language for spatial computing
  - Introduction of topological concepts in a programming language
  - Two main principles
    - Data structure: topological collection
    - Control structure: transformation
- Musical analysis

Outline

- Background on MGS spatial-computing
- Music and spatial computing



Space for musical representation

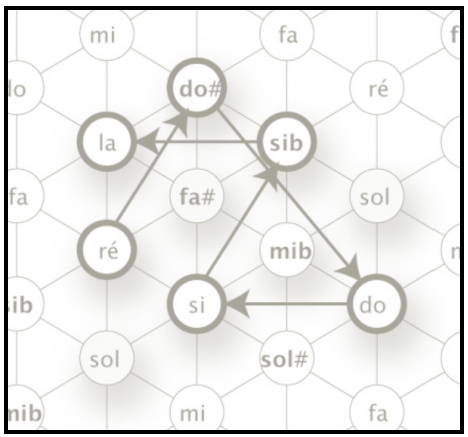


Space as musical representation

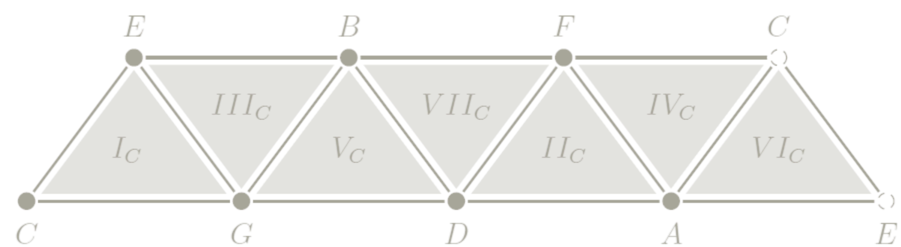
- Conclusion

Outline

- Background on MGS spatial-computing
- Music and spatial computing



Space for musical representation

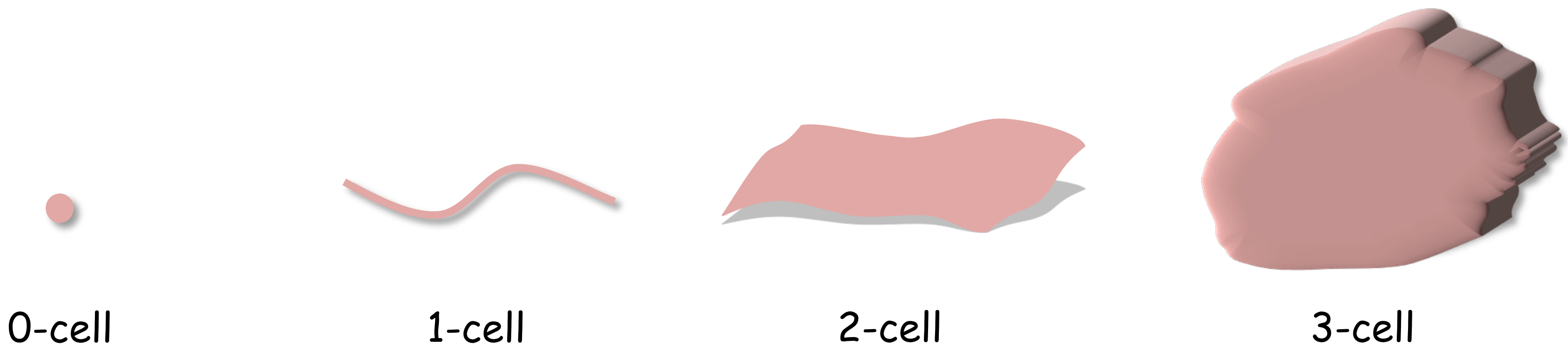


Space as musical representation

- Conclusion

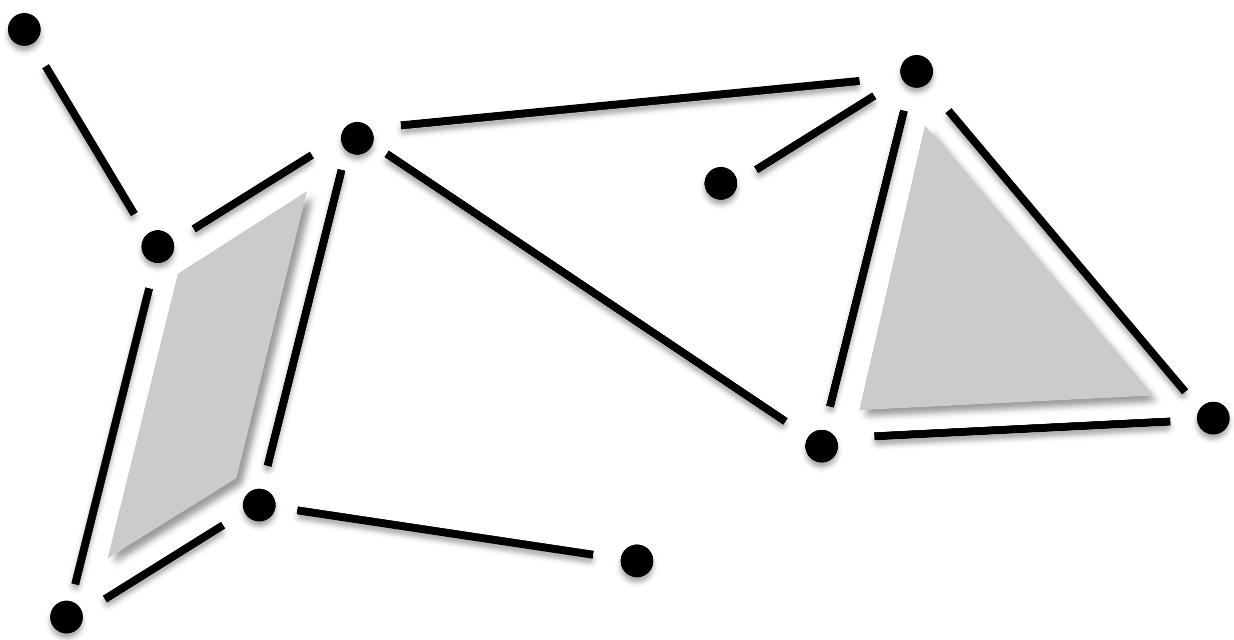
MGS Main Concepts

- Topological collections
  - Structure
    - A collection of ***topological cells***



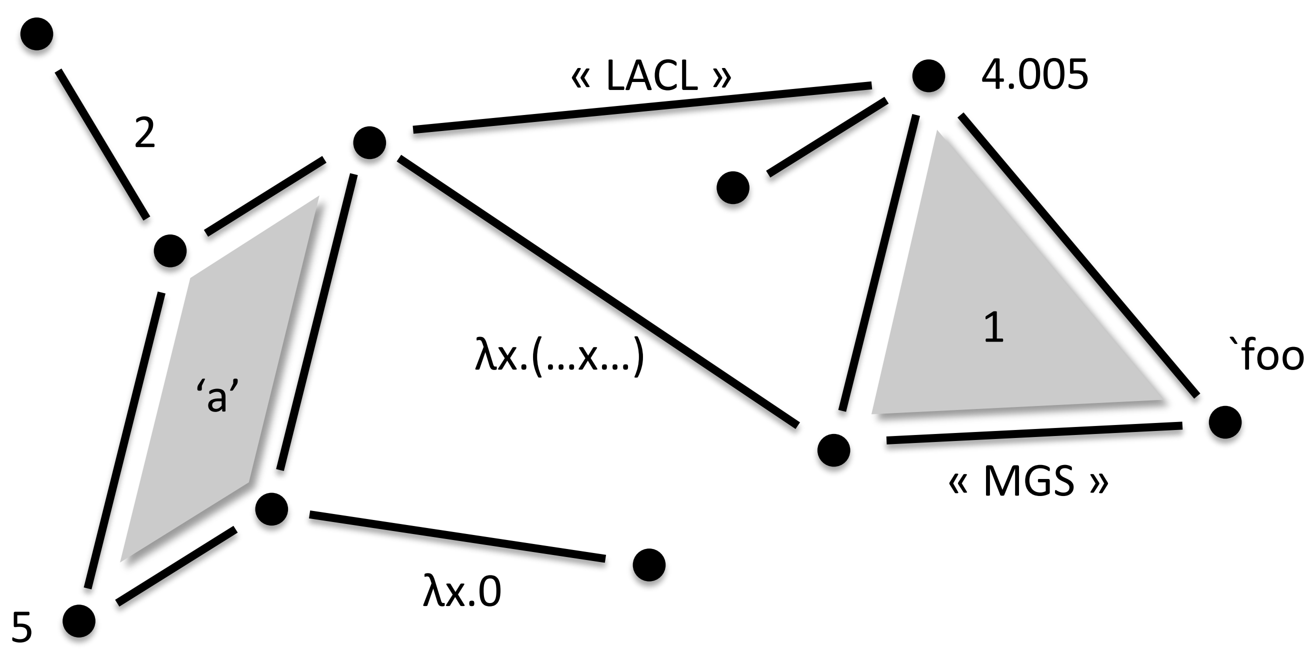
MGS Main Concepts

- Topological collections
  - Structure
    - A collection of topological cells
    - An ***incidence relationship***



MGS Main Concepts

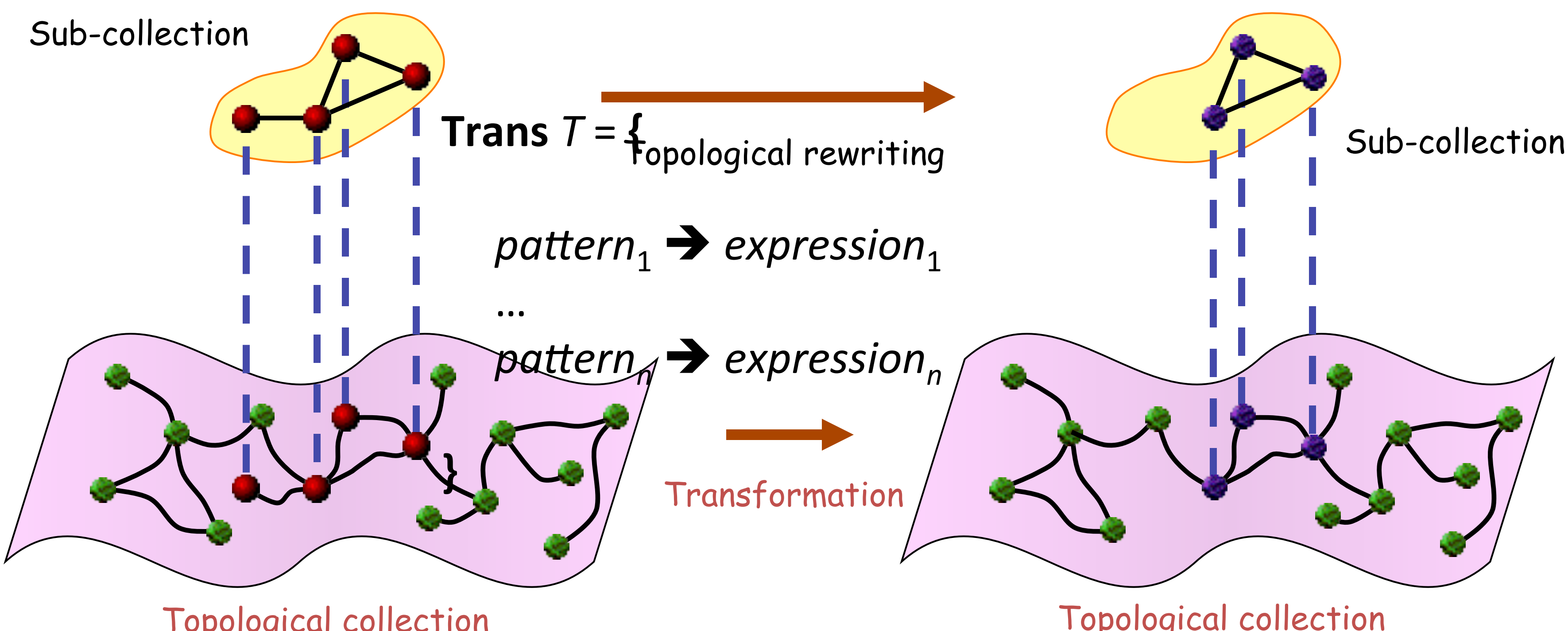
- Topological collections
  - Structure
    - A collection of topological cells
    - An incidence relationship
  - Data: **association of a value with each cell**



MGS Main Concepts

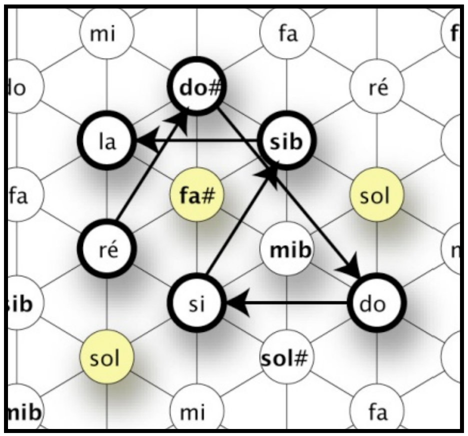
- Transformations
  - Functions defined by case on collections

    Each case (pattern) matches a sub-collection
  - Defining a rewriting relationship: ***topological rewriting***

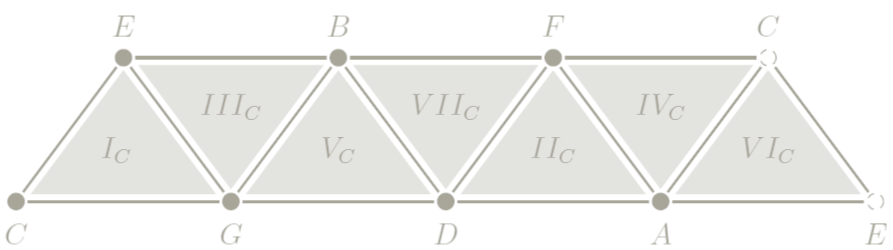


Outline

- Background on MGS spatial-computing
- Music and spatial computing



Space for musical representation



Space as musical representation

- Conclusion

Neo-Riemannian Problematic

- Traditional western music representation
  - Based on the use of a *staff*
  - Main drawbacks for visualization of harmony
    - *Contrapuntal* proximity

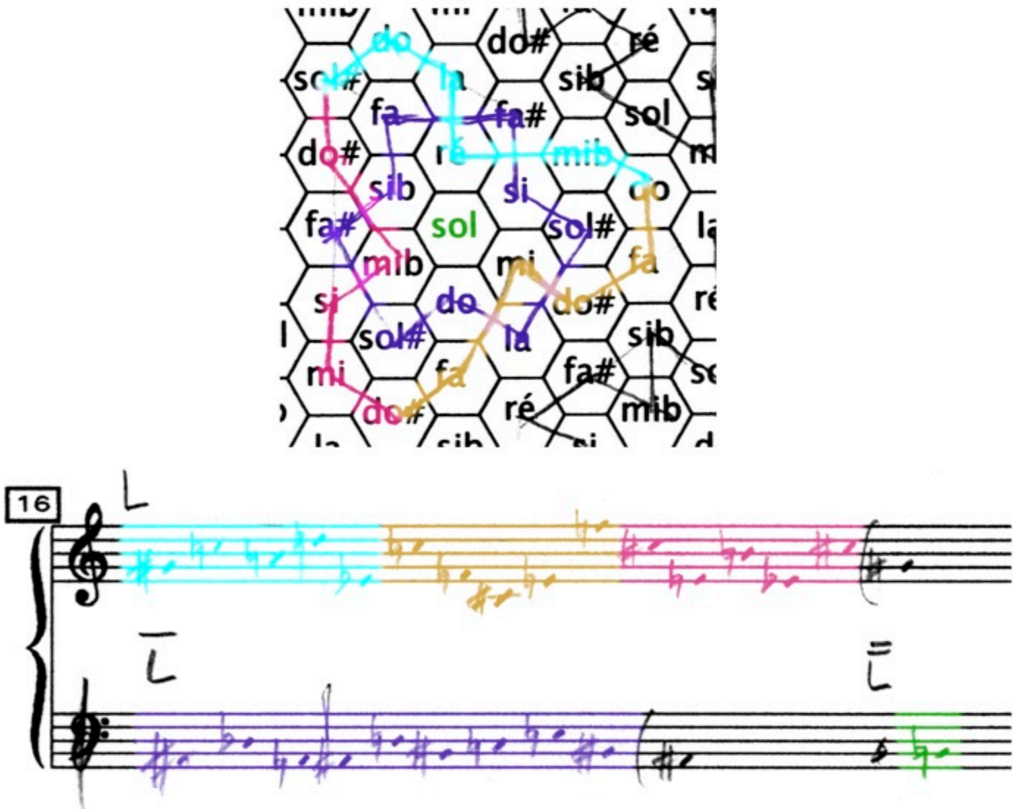
      The spatial distance between notes is not relevant for harmonic purpose
    - Spatially close patterns can sound very different

- Neo-Riemannian representation of music
  - Graphical representation of harmony rules
  - Consonance used as a graphical criterion

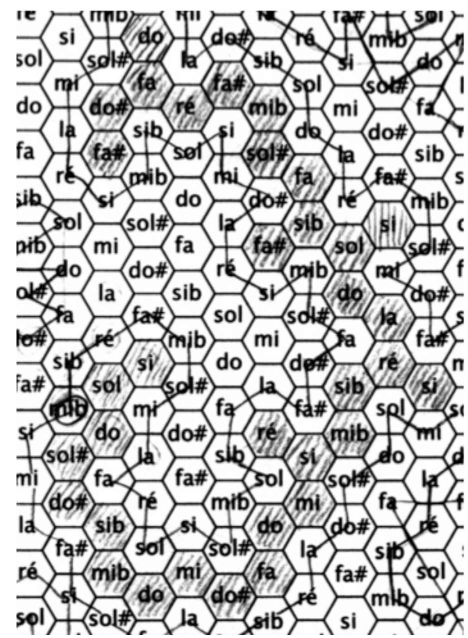
    Two notes that sound well must be spatially close

Neo-Riemannian Problematic

- Musical composition (J.-M. Chouvel)



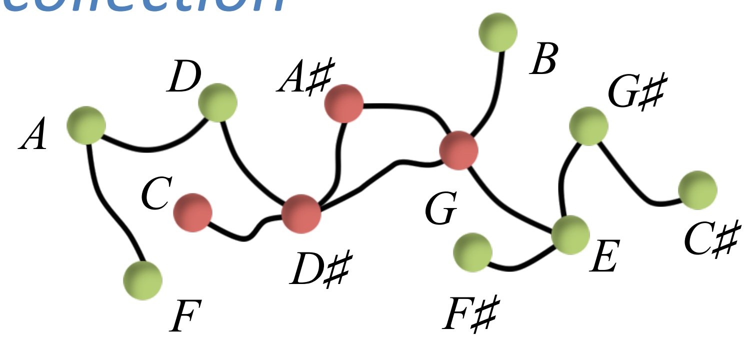
Musical segments as successive transformations of a shape in the lattice



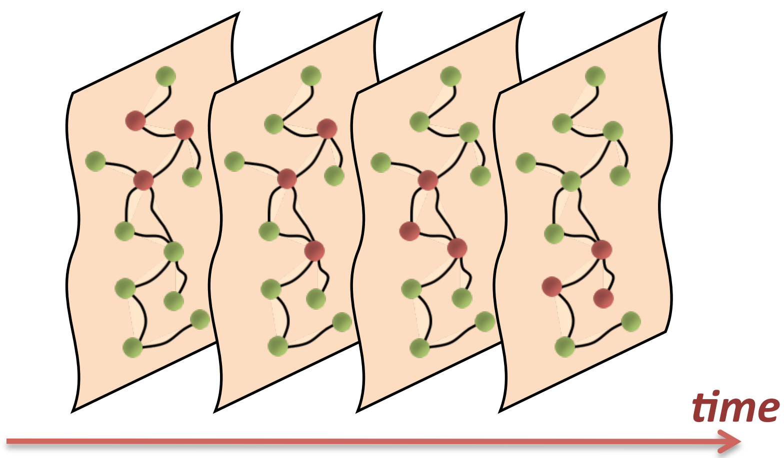
Musical melody as a path in the lattice

Spatial Representation of Musical Sequences

- Temporal succession of musical events
- Musical event as a *topological collection*
  - Positions are notes
  - Labels represent played notes



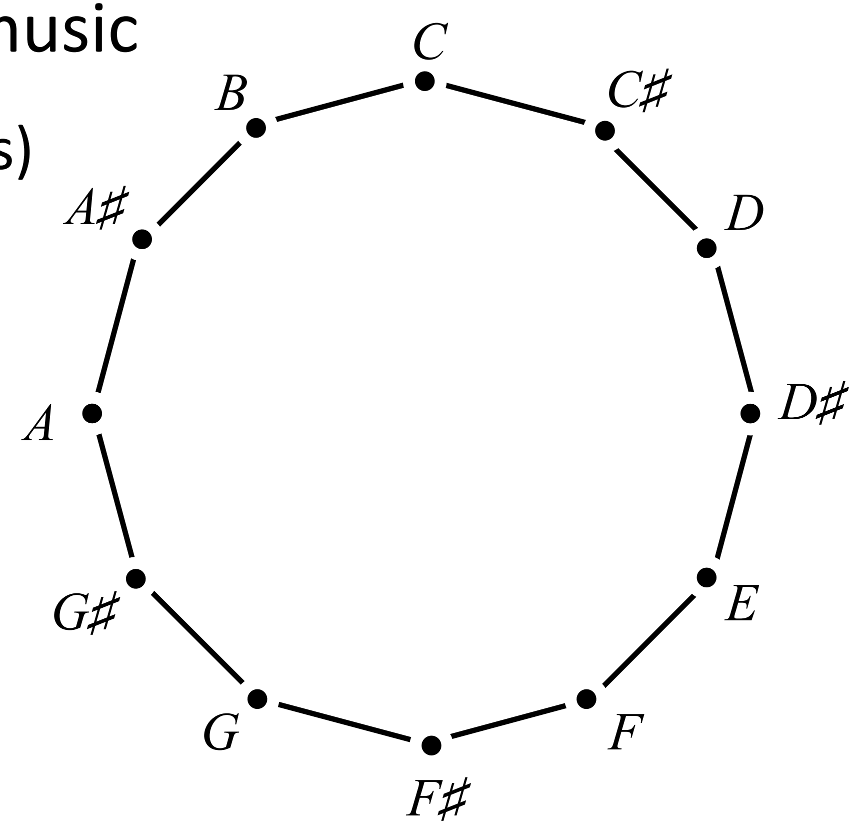
- Succession of events as a *stream of collections*



Extract of the 2nd movement of the Symphony No. 9 (L. van Beethoven)

Formalization of Notes Neighborhoods

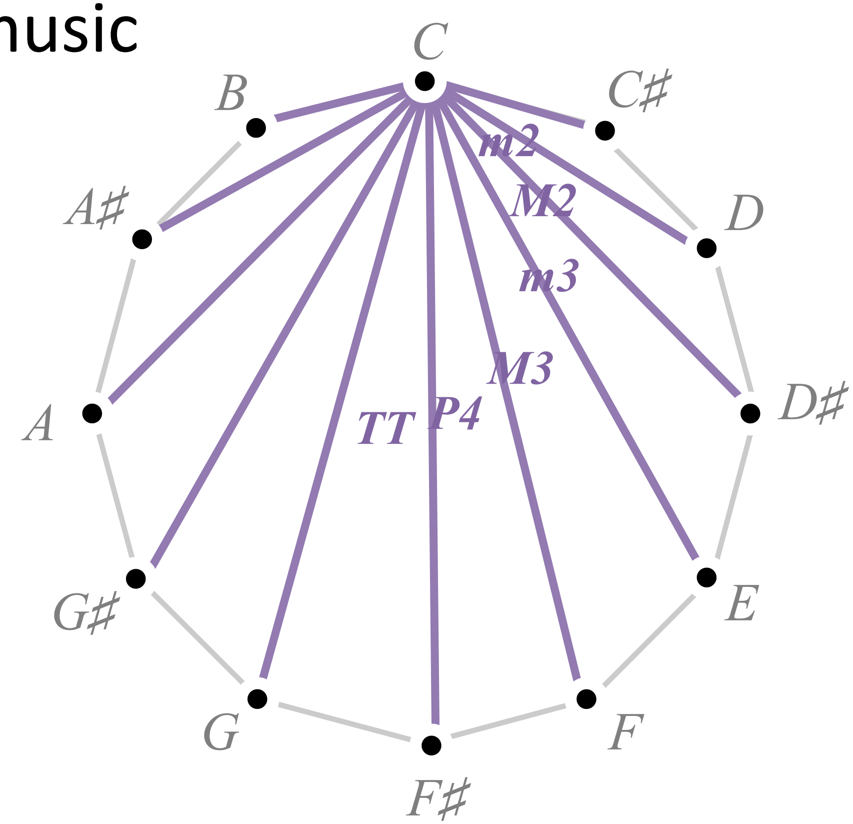
- Which neighborhoods for significant visualization?
- Strong algebraic structure of music
  - Set $\mathcal{N}$ of notes (i.e. pitch classes)
    We do not consider octaves



$$\mathcal{N} = \{\ C,\ C\#,\ D,\ D\#,\ E,\ F,\ F\#,\ G,\ G\#,\ A,\ A\#,\ B\ \}$$

Formalization of Notes Neighborhoods

- Which neighborhoods for significant visualization?
- Strong algebraic structure of music
  - Set $\mathcal{N} = \{ C, C\sharp, \ldots, A\sharp, B \}$
  - Group $(\mathcal{I},+)$ of intervals
    - Relative difference between notes

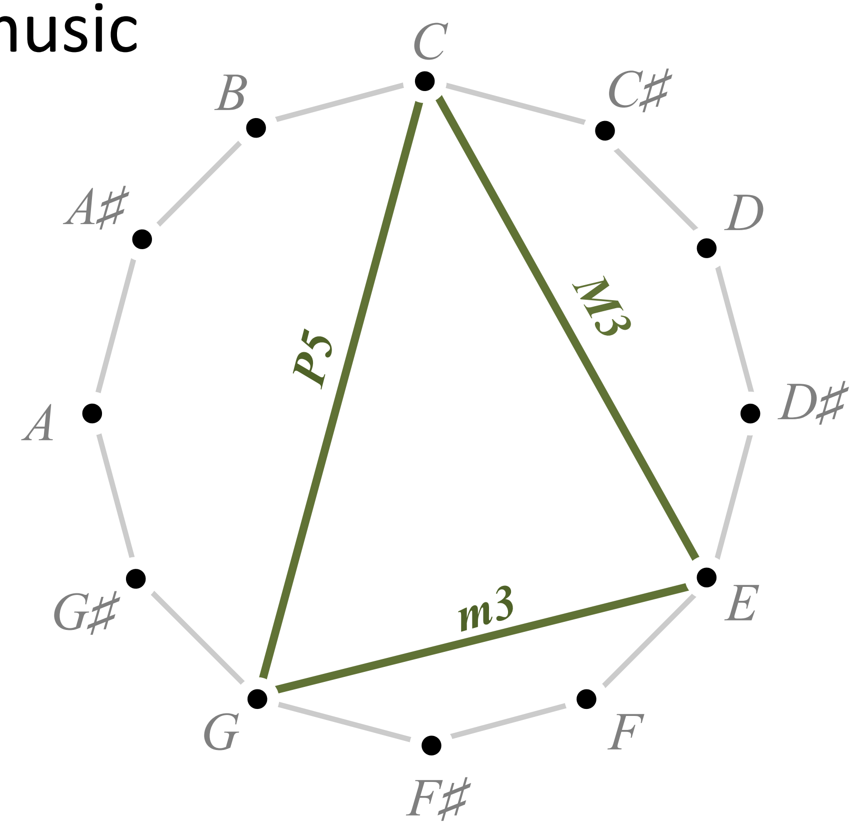


$$\mathcal{I} = \{ P1, m2, M2, m3, M3, P4, TT, P5, m6, M6, m7, M7 \}$$

Formalization of Notes Neighborhoods

- Which neighborhoods for significant visualization?
- Strong algebraic structure of music
  - Set $\mathcal{N} = \{ C, C\sharp, \ldots, A\sharp, B \}$
  - Group $(\mathcal{I},+)$ of intervals
    - Relative difference between notes
    - $(\mathcal{I},+) \cong (\mathcal{Z}_{12},+)$ (isomorphism)
    - Example

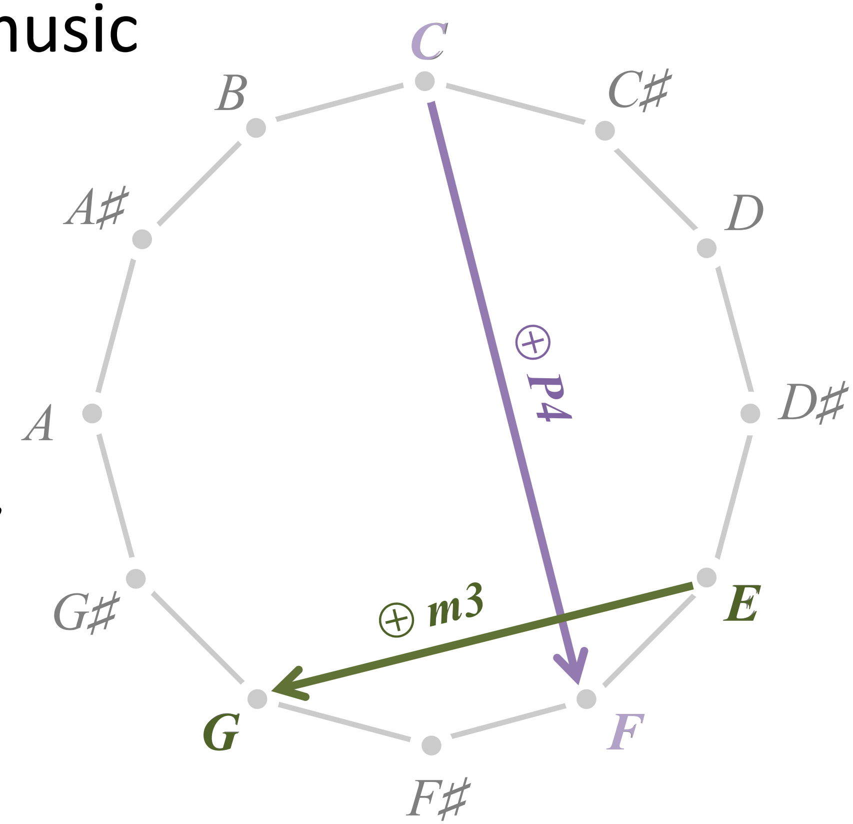
      $M3 + m3 = P5$



$$\mathcal{I} = \{ P1, m2, M2, m3, M3, P4, TT, P5, m6, M6, m7, M7 \}$$

Formalization of Notes Neighborhoods

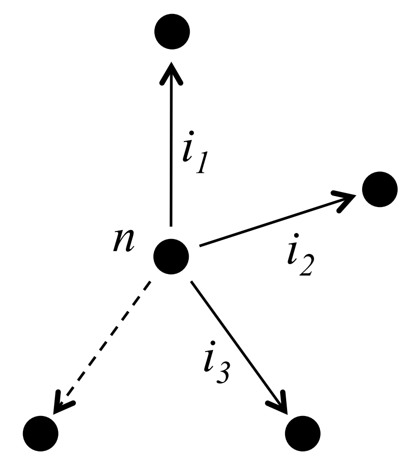
- Which neighborhoods for significant visualization?
- Strong algebraic structure of music
  - Set $\mathcal{N} = \{ C, C\sharp, \ldots, A\sharp, B \}$
  - Group $(\mathcal{I},+) = \{ P1, \ldots, M7 \}$
  - Transposition $\oplus$ of notes
    - $n \oplus i = n'$
      if i is the interval between n and n'
    - Example
      - $C \oplus P4 = F$
      - $E \oplus m3 = G$



$\oplus$ action of $\mathcal{I}$ over $\mathcal{N}$

Formalization of Notes Neighborhoods

- Consonance as a neighborhood relationship
  - $S \subset \mathcal{N} \times \mathcal{N}$
  - $(n_1, n_2) \in S$ if n_1 “sounds well” with n_2
- Assumptions on S
  - S is symmetric
    - $(n_1, n_2) \in S \Rightarrow (n_2, n_1) \in S$
  - S is defined up to a transposition
    - $\forall i \in \mathcal{I}, \quad (n_1, n_2) \in S \Rightarrow (n_1 \oplus i, n_2 \oplus i) \in \mathrm{S}$
    - (C, G) sounds well $\Rightarrow$ $(E = C \oplus M3, B = G \oplus M3)$ sounds well
- S characterized by a subset I of $\mathcal{I}$
    - $I = \{ i_1, i_2, \ldots, i_n \}$
    - $\forall n \in \mathrm{N}, \forall \mathrm{i} \in I, \quad (n, n \oplus i) \in S$



Formalization of Notes Neighborhoods

- Spatial representation of S
  - I as a set of group generators
    - $< I >$ subgroup of $\mathcal{I}$ generated by the elements of I
    - Example $I = \{ M2 \}$

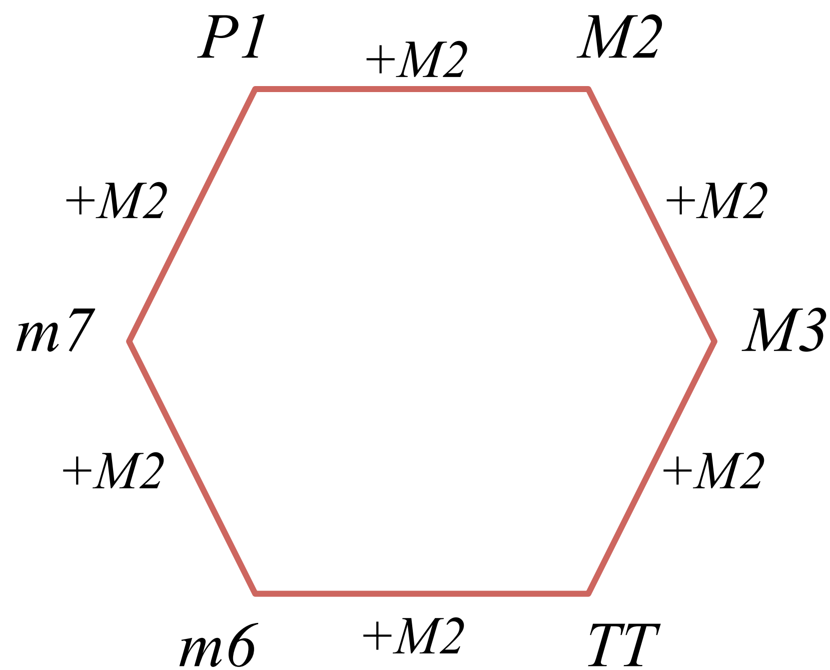
$$S = \{ (C,D), (C\#,D\#), \ldots \}$$

$$I = \{ M2 \}$$

$$\begin{aligned} < I > &= \{ P1, P1 + M2, P1 + 2.M2, \ldots \} \\ &= \{ P1, M2, M3, TT, m6, m7 \} \end{aligned}$$

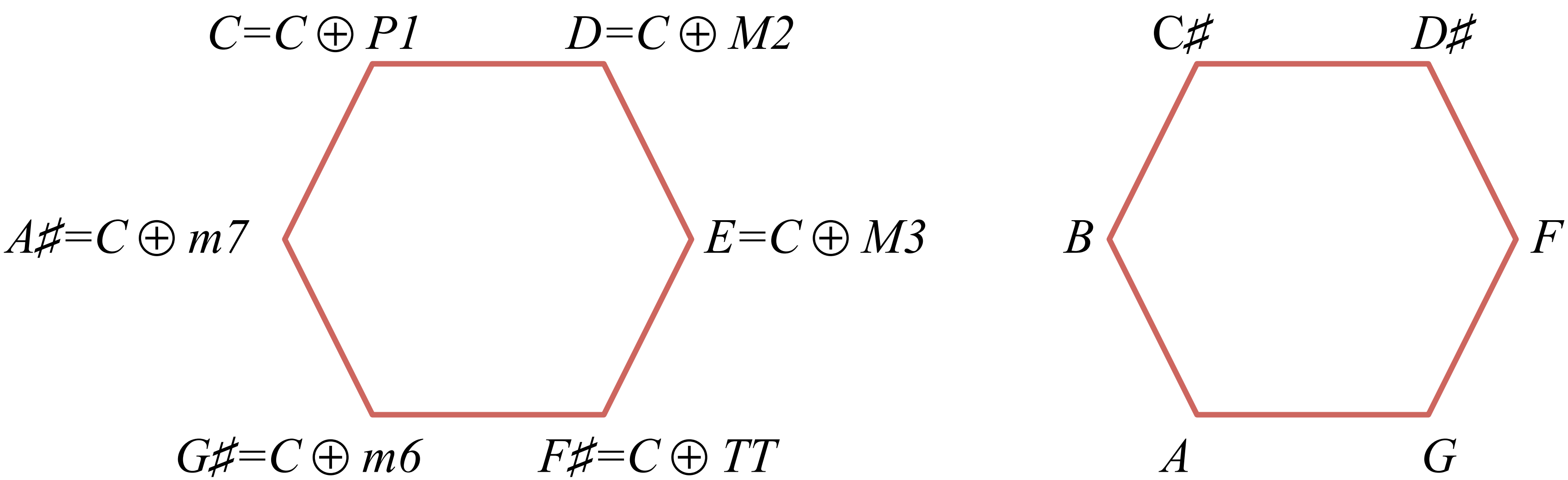
Formalization of Notes Neighborhoods

- Spatial representation of S
  - I as a set of group generators
  - Graph representation of $< I >$
    - Cayley's graph
      - Vertices: intervals of $< I >$
      - Edges: generators of I
    - Example with $I = \{ M2 \}$



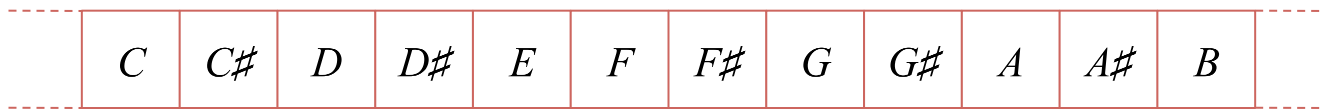
Formalization of Notes Neighborhoods

- Spatial representation of S
  - I as a set of group generators
  - Graph representation of $< I >$
  - Representation of S based on Cayley graph
    - Action of $< I >$ on N
    - Example with $I = \{\ M2\ \}$

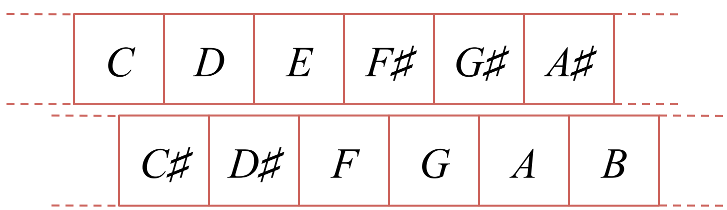


Applications

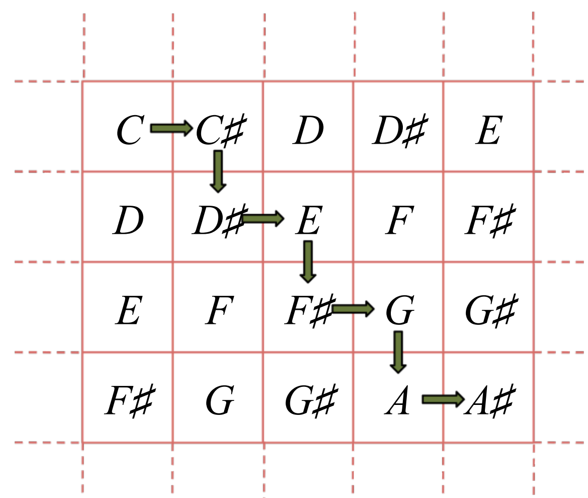
- Scale representations
  - Chromatic scale $I = \{ m2 \}$



  - Whole-tone scale $I = \{ M2 \}$

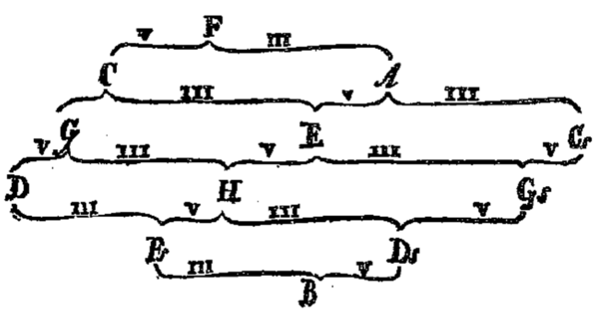


  - Diminished scale $I = \{ m2, M2 \}$

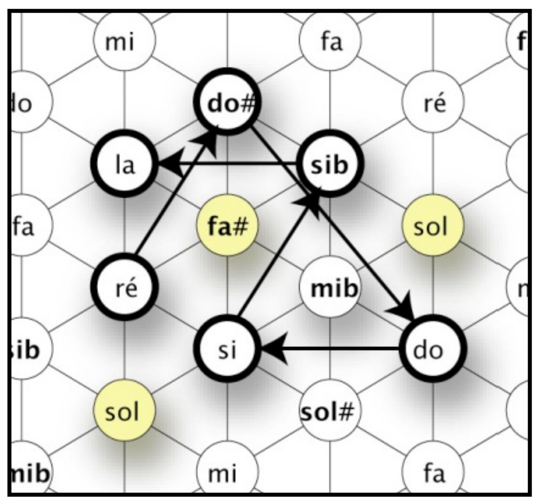


Applications

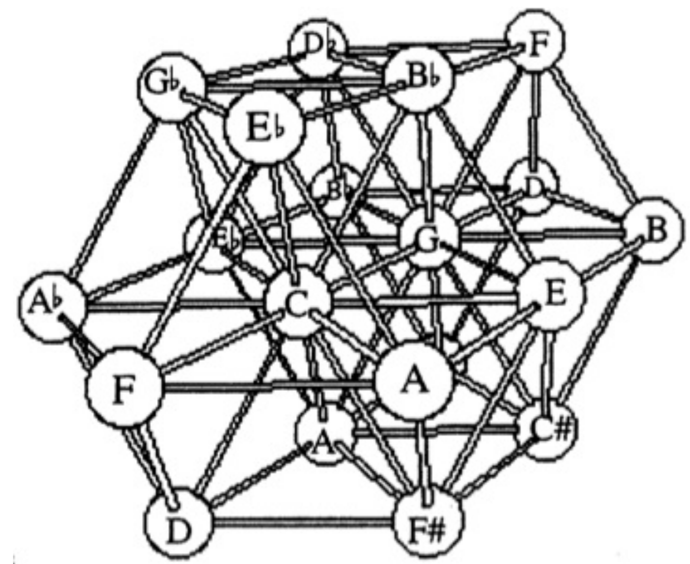
- Traditional harmony representation
  - $I = \{\ M3,\ P5\ \}$ (Euler's Tonnetz)
  - $I = \{\ m3,\ M3,\ P5\ \}$ (Harmonic table)
  - $I = \{\ m3,\ M3,\ P5,\ m7\ \}$ (3D-Tonnetz)



(Euler)



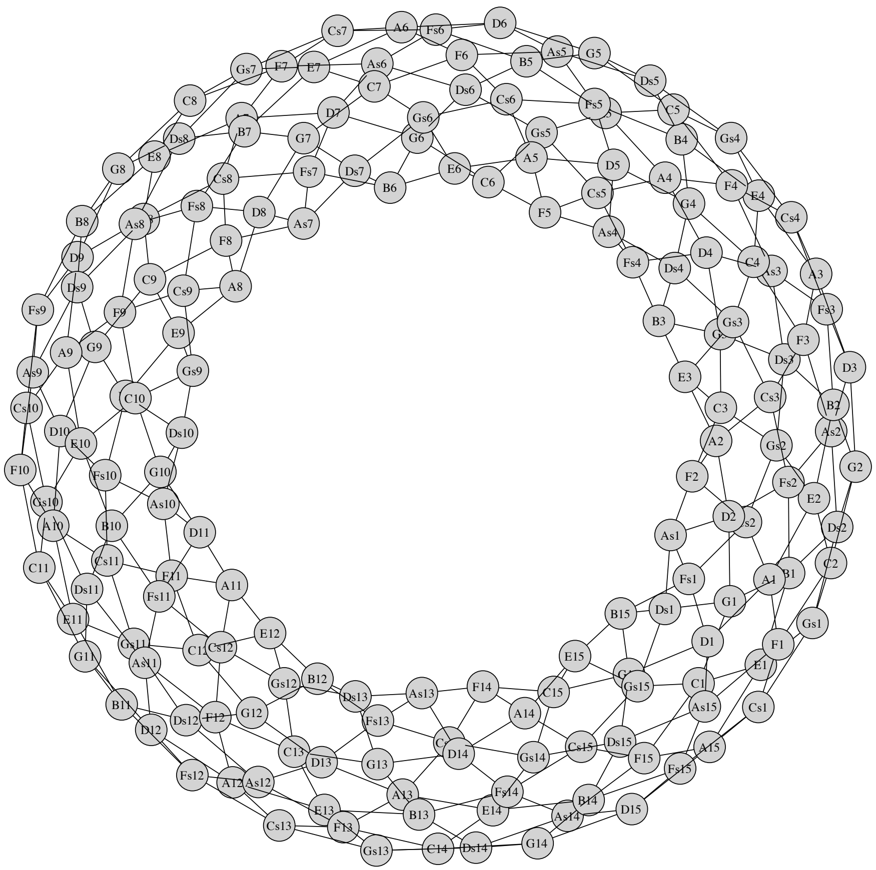
(J.-M. Chouvel)



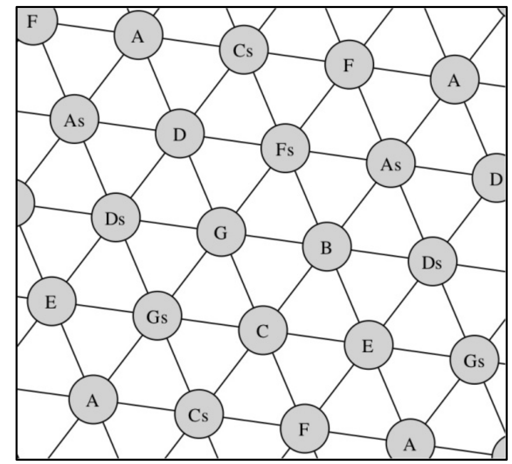
(E. Gollin)

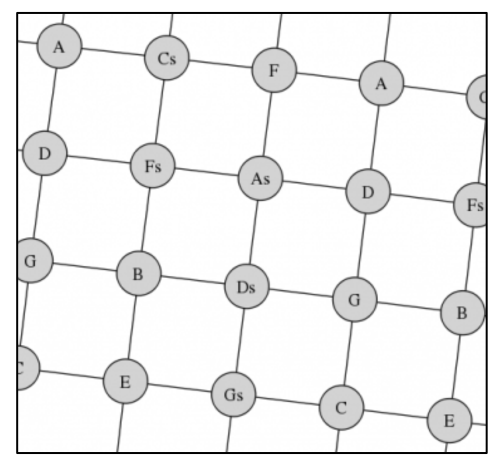
Applications

- Automatic graph generation



unfolding





Applications

- Instruments conception

$I = \{ m2, P4 \}$ (Guitar)

$I = \{ m2, P5 \}$ (Violin)

$I = \{ m2, M2, m3 \}$ (Accordion)

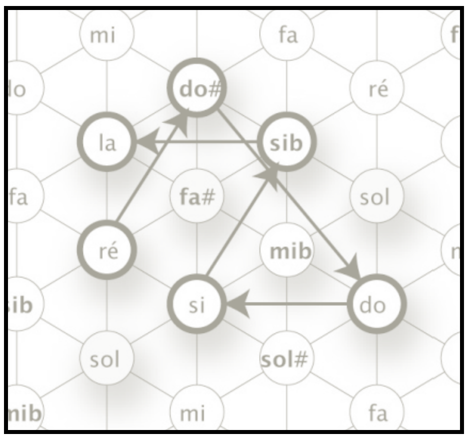
Applications

- Analysis example
  - Signature of a piece
  - Example : F. Chopin Prelude

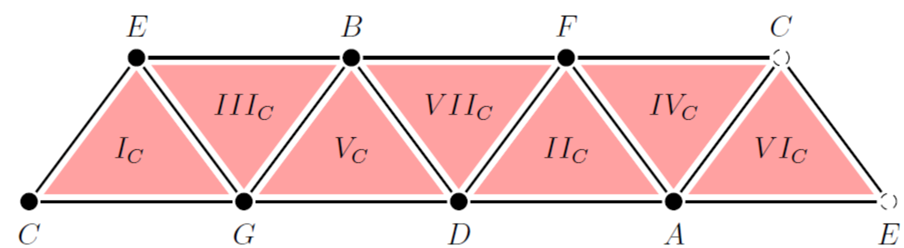
Extract of the Prelude N.4
Op28 of F. Chopin

Outline

- Background on MGS spatial-computing
- Music and spatial computing



Space for musical representation

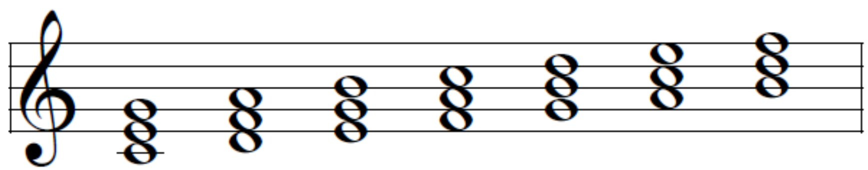


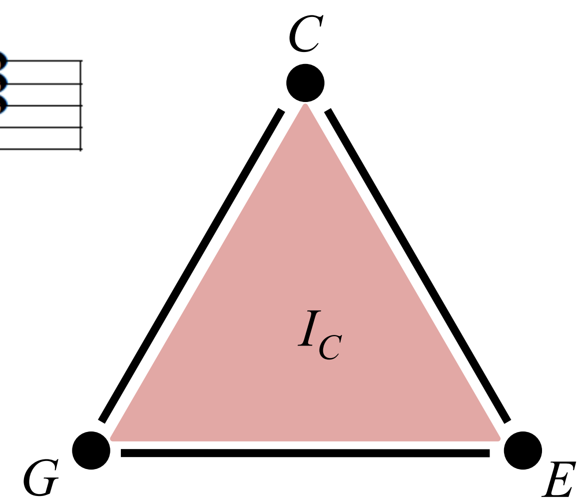
Space as musical representation

- Conclusion

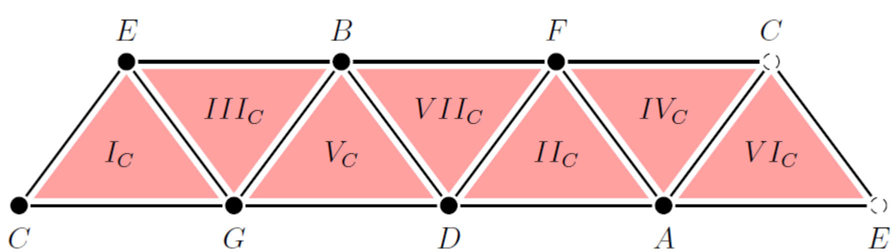
Tonality and Möbius Strip

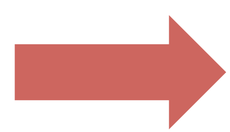
- Motivation: spatial visualization of tonality
- Association of a chord set with the tonality: the *degrees*
  - Example: C-major tonality

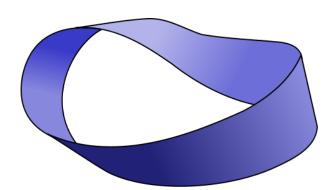




- Spatial representations
  - Note = vertex
  - Chord = surface
- Fusion of the common notes for the 7 degrees



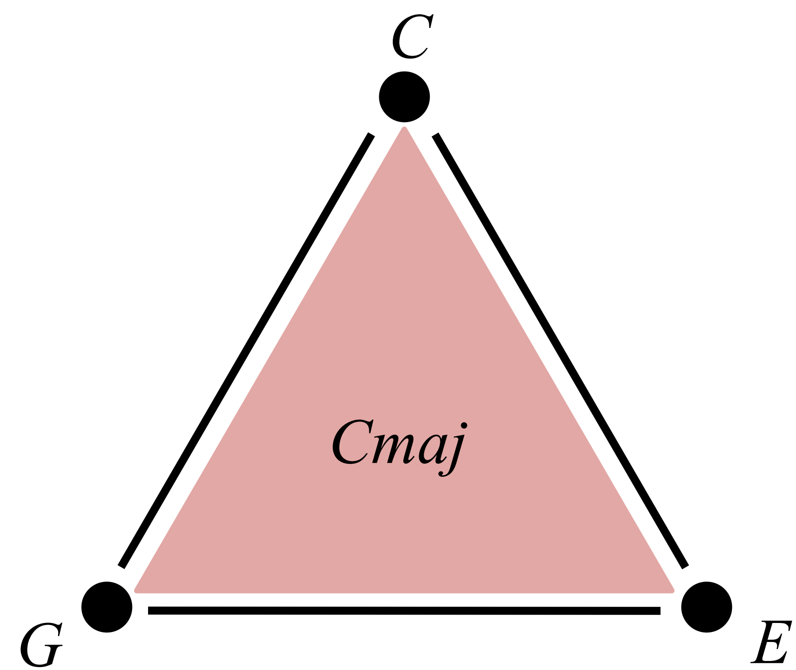




[Mazzola02]

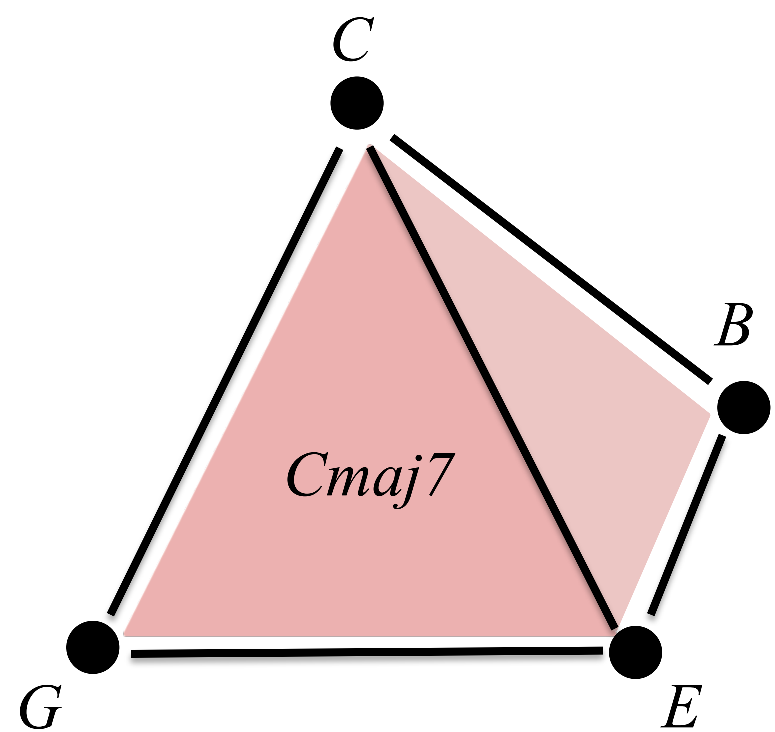
Self-Assembly of Chords

- Automation of the process for the analysis of other chords sequences
- *Reaction* of the chords between themselves
- Simplicial representation of musical objects
  - Note: 0-simplex
  - 2-note chord: 1-simplex
  - 3-note chord: 2-simplex



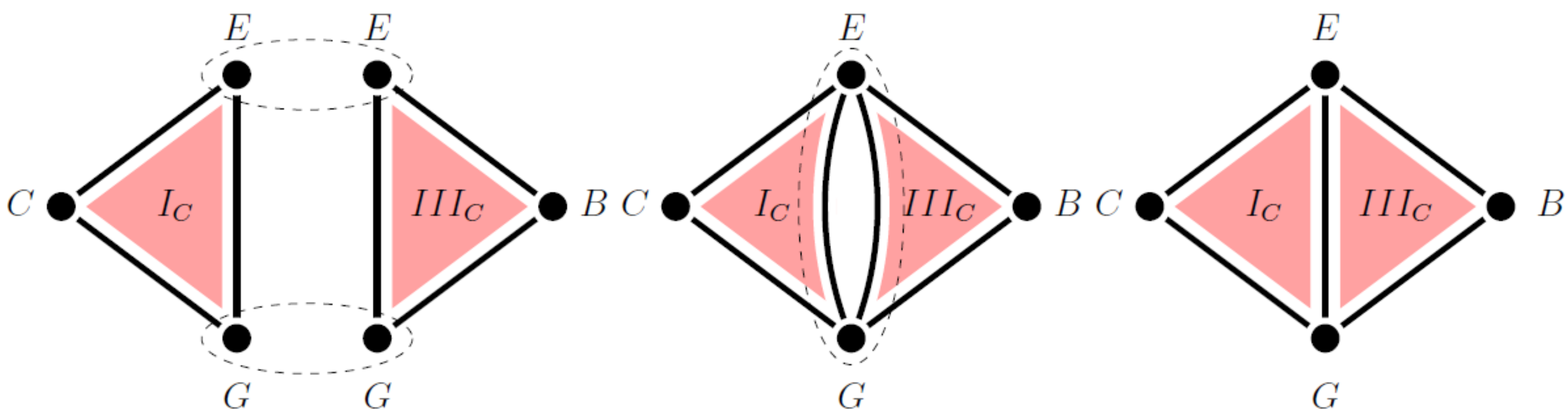
Self-Assembly of Chords

- Automation of the process for the analysis of other chords sequences
- *Reaction* of the chords between themselves
- Simplicial representation of musical objects
  - Note: 0-simplex
  - 2-note chord: 1-simplex
  - 3-note chord: 2-simplex
  - 4-note chord: 3-simplex



Self-Assembly of Chords

- MGS transformation for self-assembly process

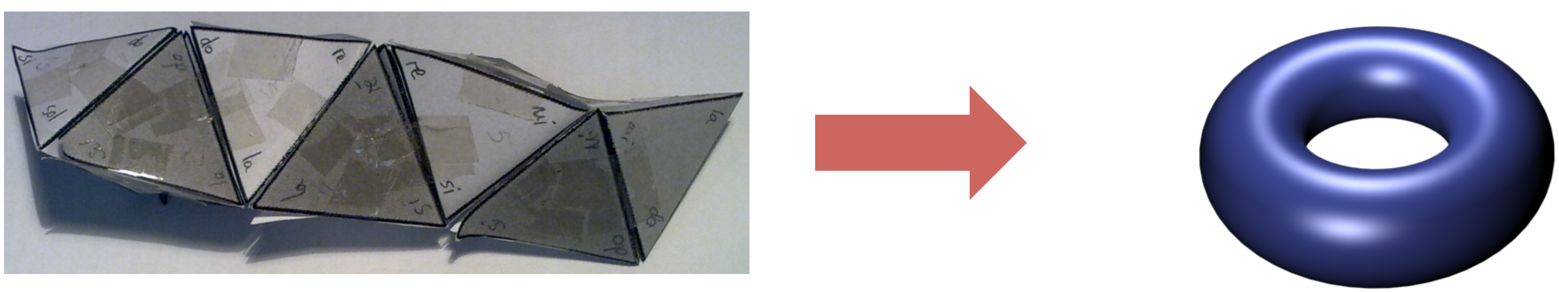



```
Trans identification = {
  s1 s2 / (s1 == s2 & (faces s1) == (faces s2))
    =>
      let c = new_cell (dim s1)
                       (faces s1)
                       (union (cofaces s1)
                              (cofaces s2))
      in s1 * c
}
```

Applications

- Four-note degrees of C-major tonality

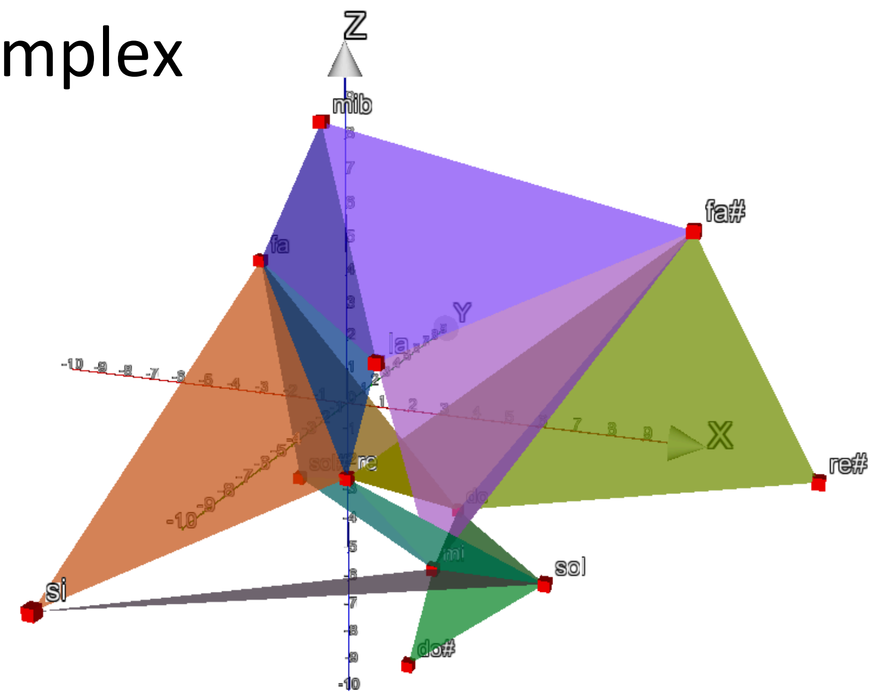
- Chord = 3-simplex (tetrahedrons)
- Self-assembly



Applications

- Extract of the Prelude No. 4 Op. 28 of F. Chopin

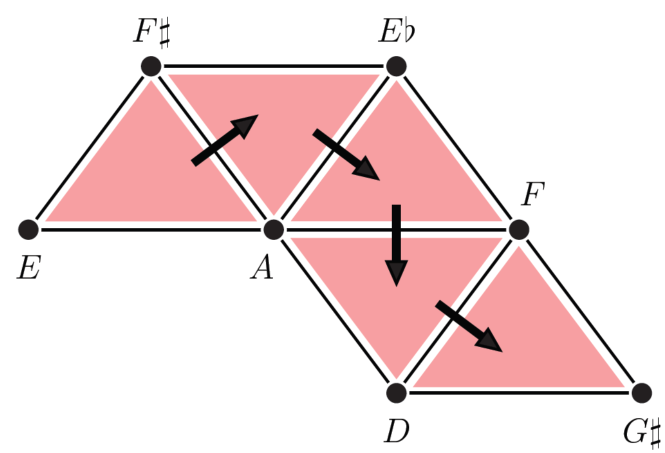
- Simplicial complex



Applications

- Extract of the Prelude No. 4 Op. 28 of F. Chopin

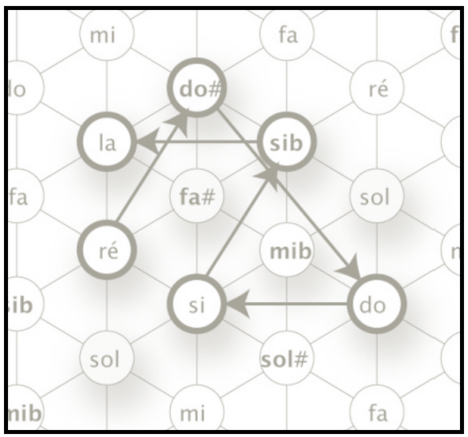
- Analysis of the path under the chords



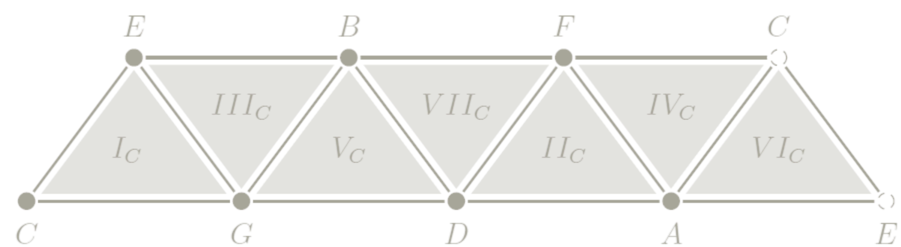
- The path chosen by F. Chopin is associated with the smallest movements on the chords

Outline

- Background on MGS spatial-computing
- Music and spatial computing



Space for musical representation



Space as musical representation

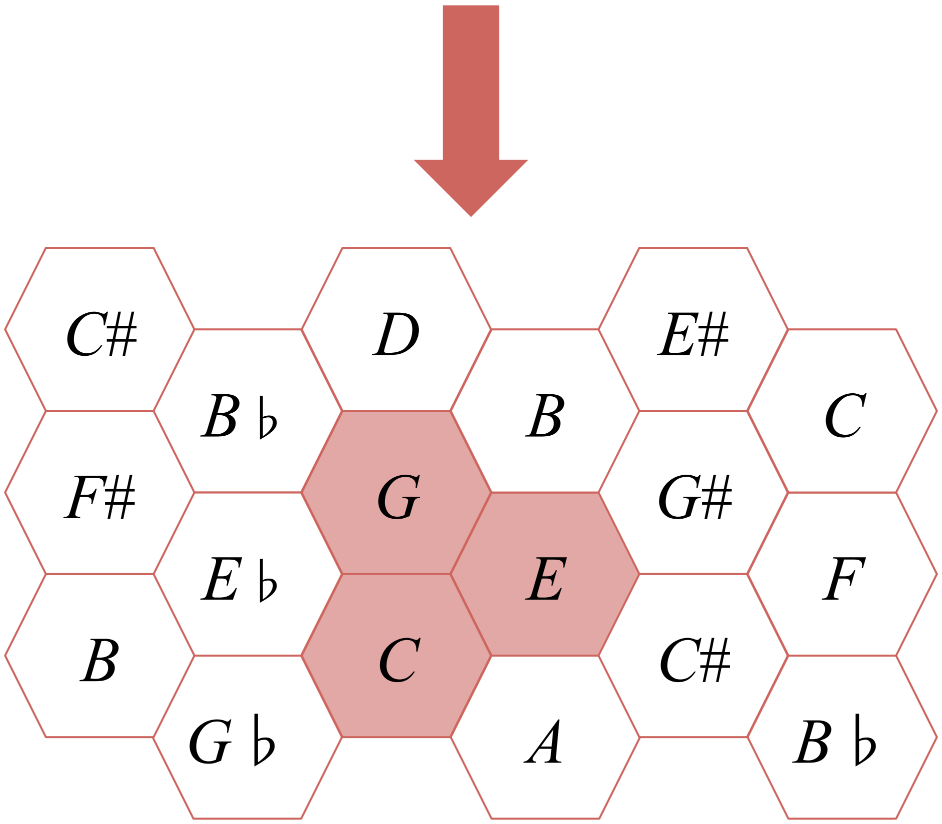
- Conclusion

Conclusion & Perspectives

- Preliminary work
- Strong collaborations with composers / musicologists
- Extend the validation on more musical problems
  - Spatial representations of other musical properties (timbre, rythm, fingering etc.)
- Extension to study musical styles
- Spatial properties ⇔ musical properties
- Acknowledgements

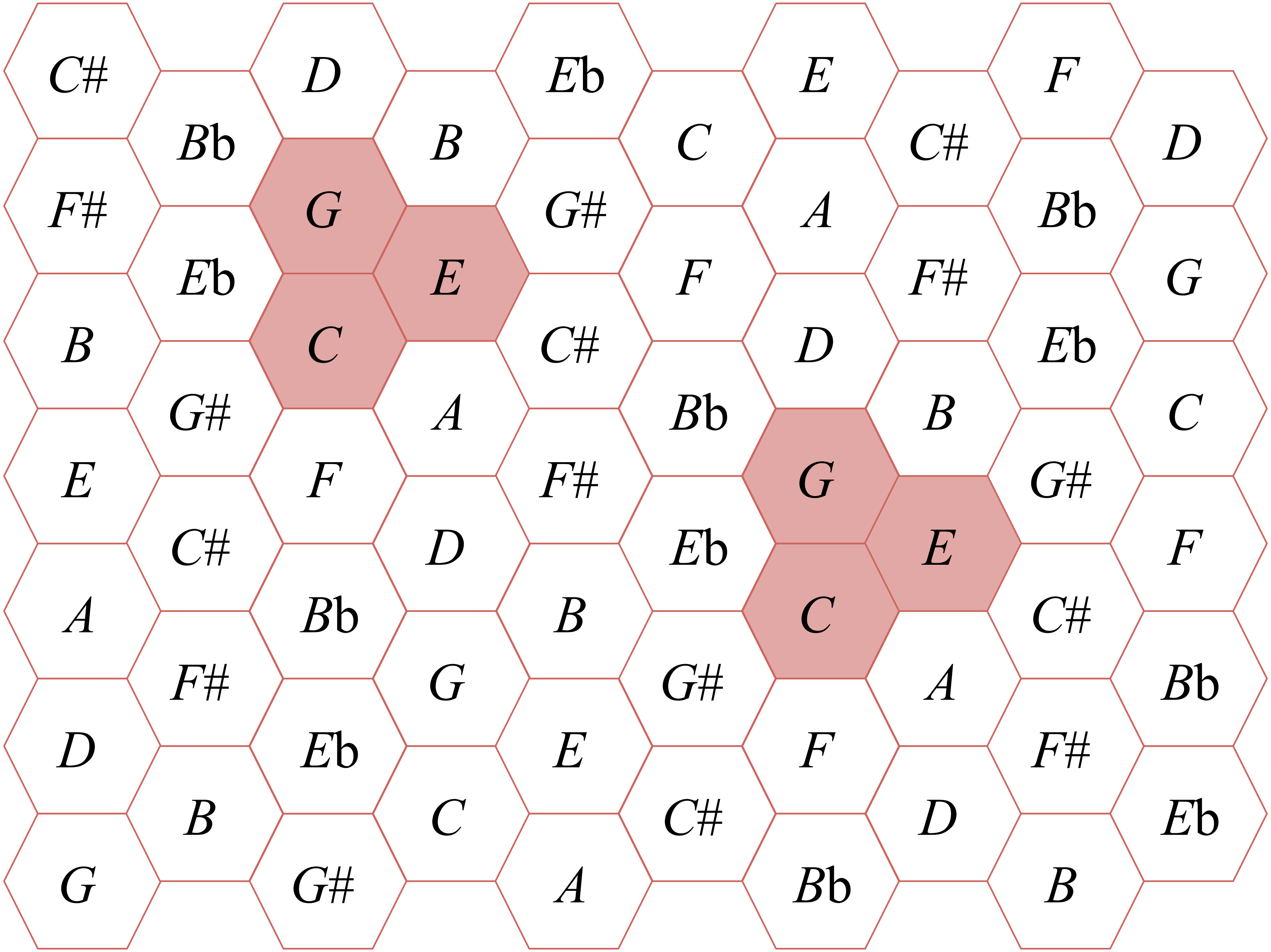
Jean-Louis Giavitto, Moreno Andreatta, Carlos Agon,
Jean-Marc Chouvel

Extract of the 2nd movement of the Symphony No. 9 (L. van Beethoven)

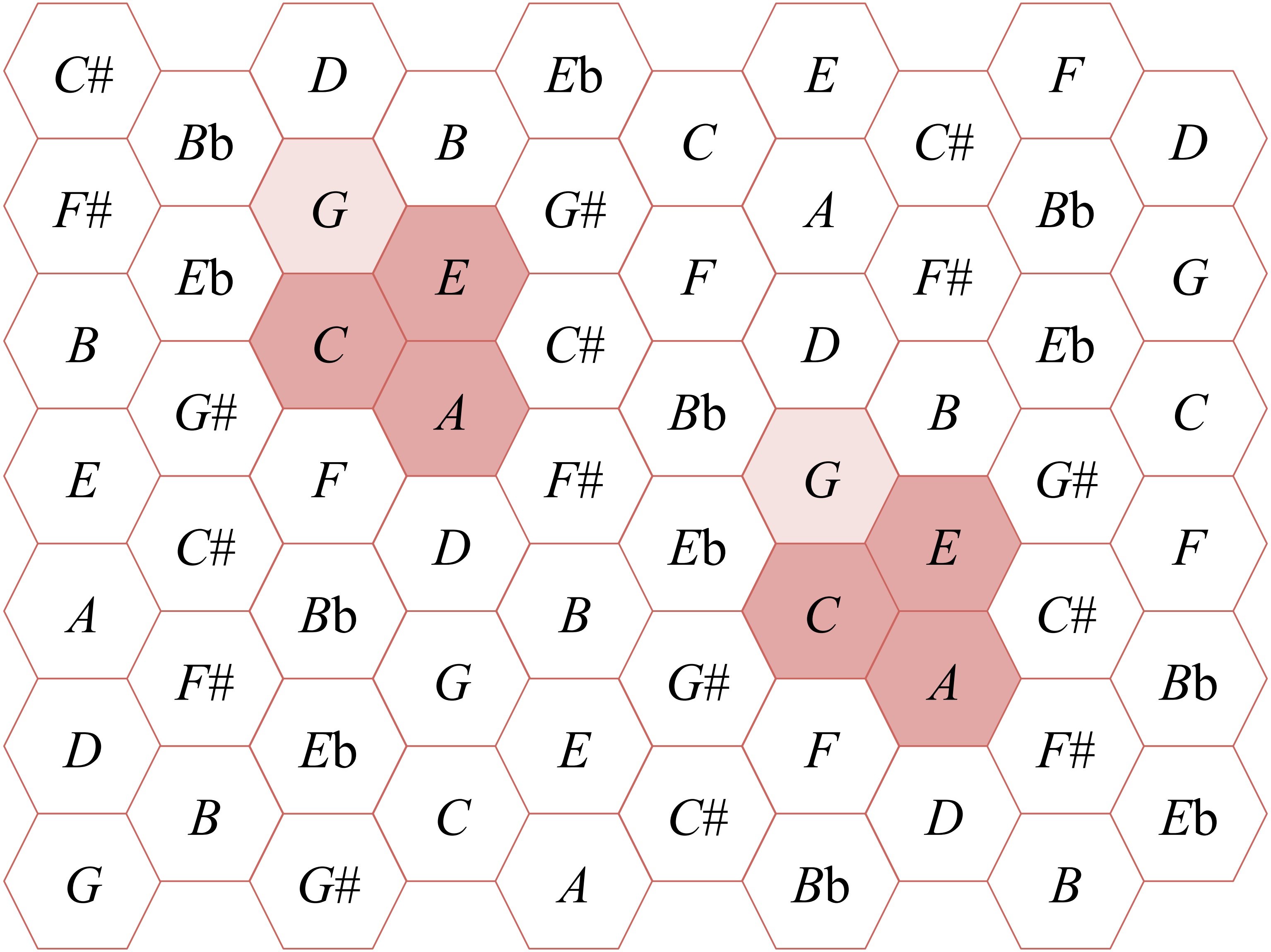


b

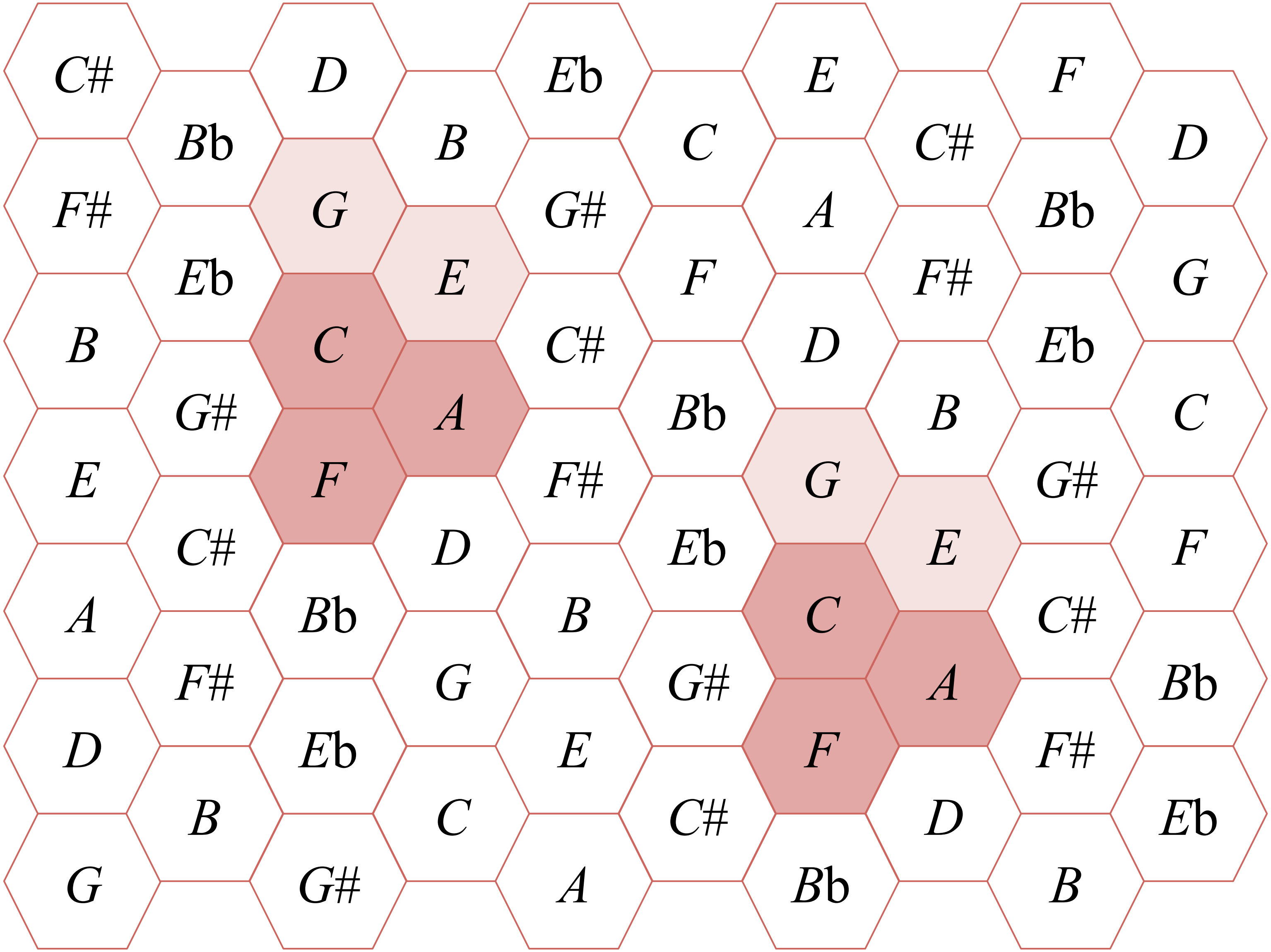
Extract of the 2nd movement of the Symphony No. 9 (L. van Beethoven)



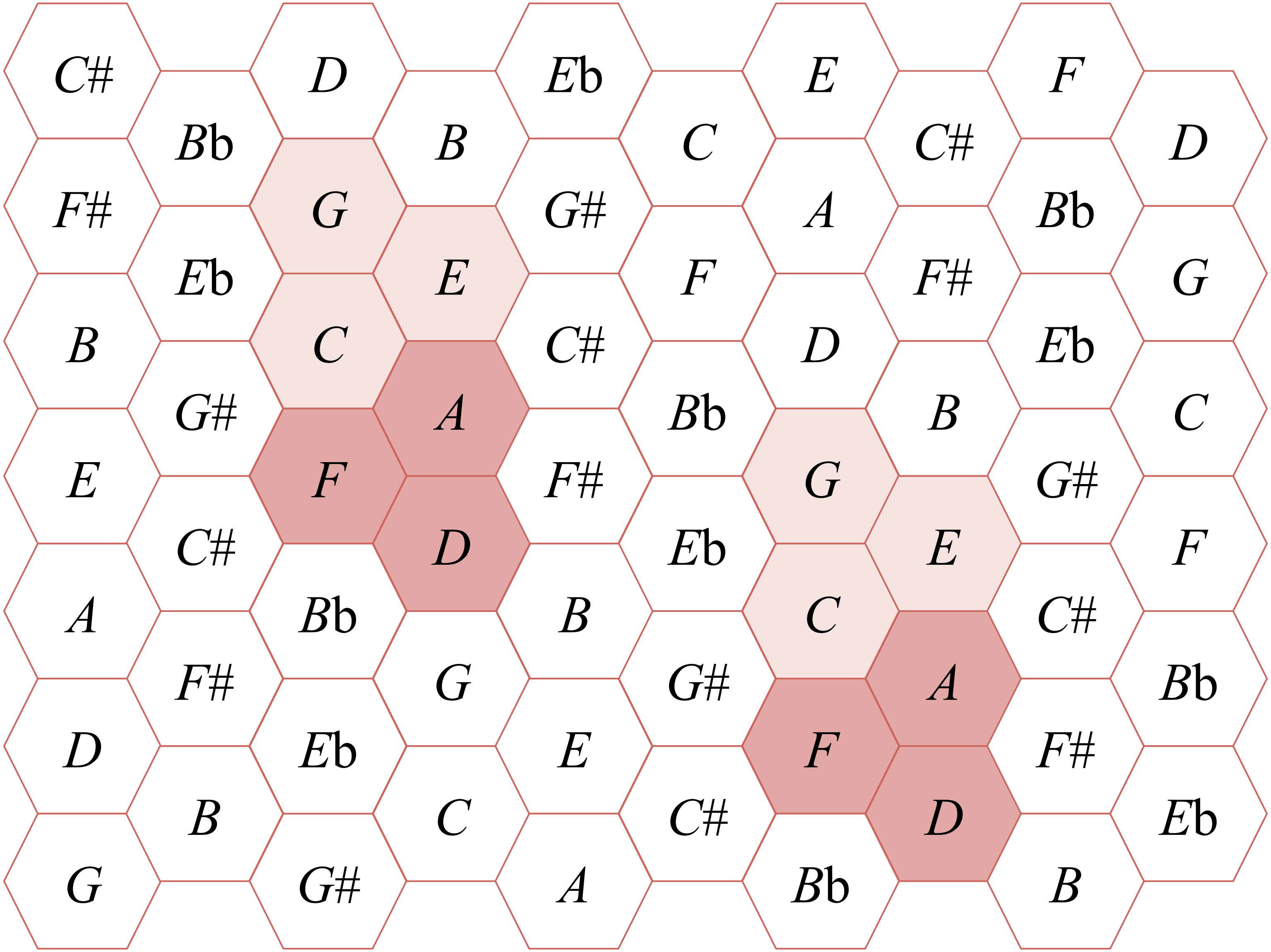
Extract of the 2nd movement of the Symphony No. 9 (L. van Beethoven)



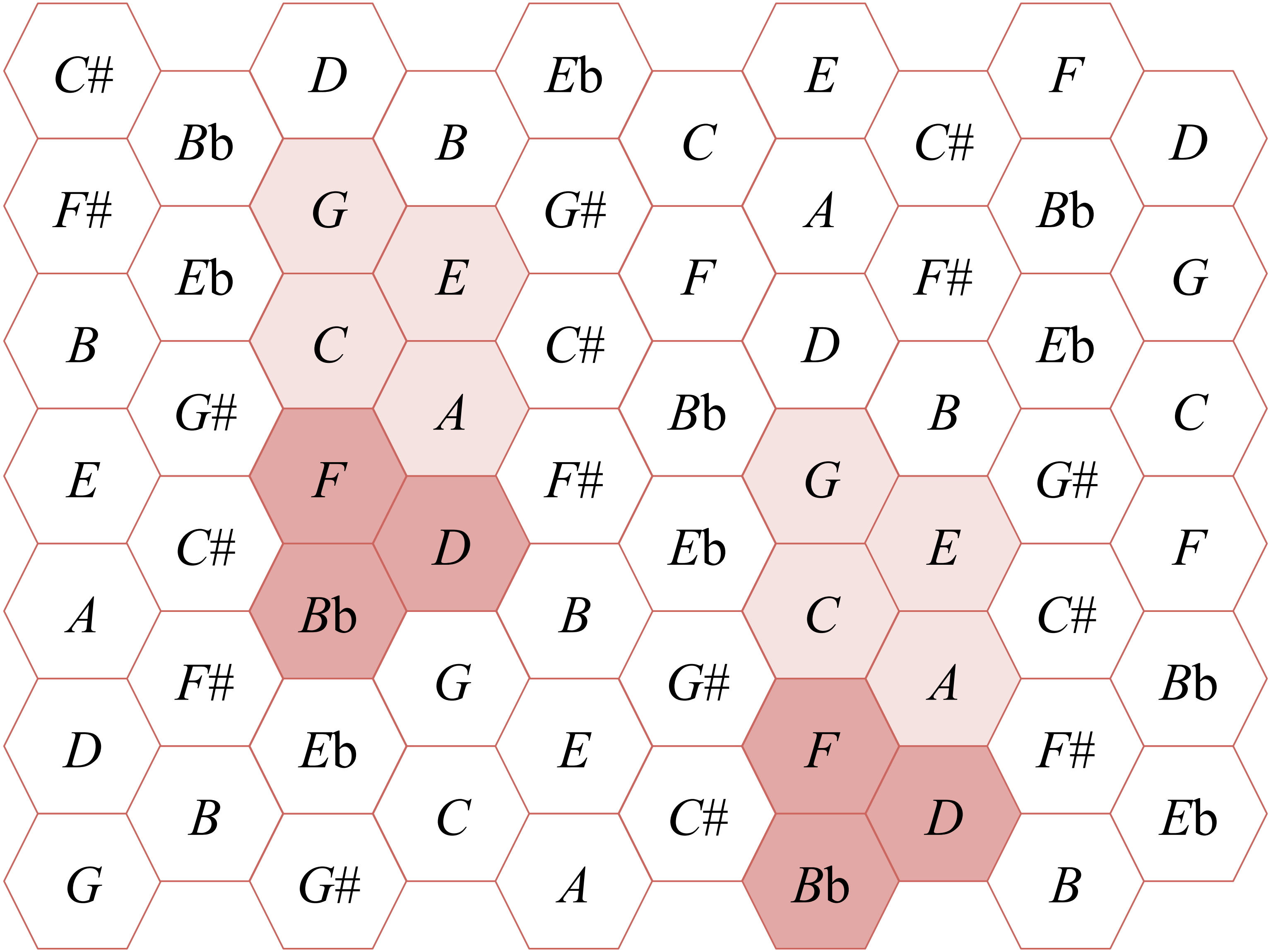
Extract of the 2nd movement of the Symphony No. 9 (L. van Beethoven)



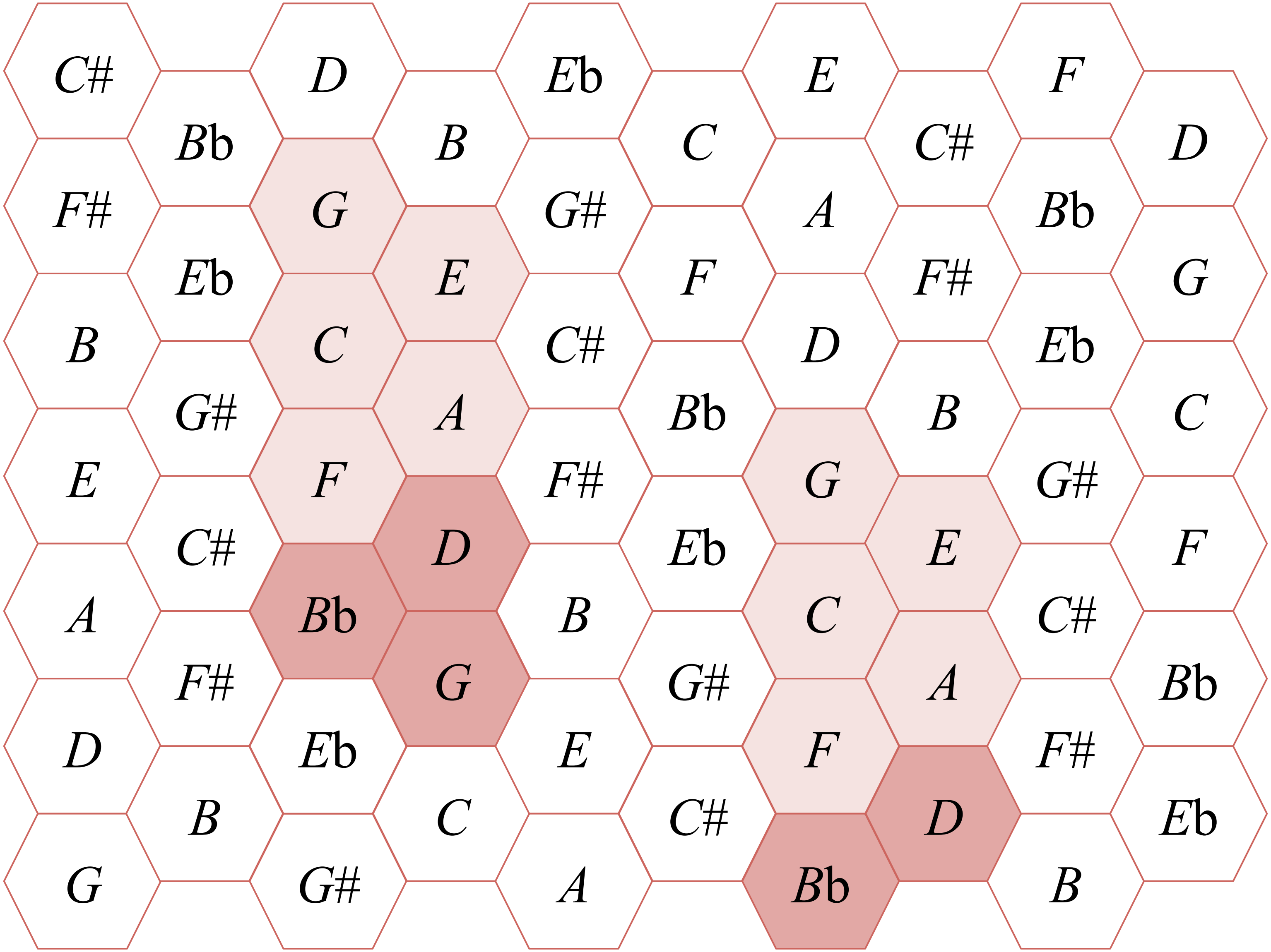
Extract of the 2nd movement of the Symphony No. 9 (L. van Beethoven)



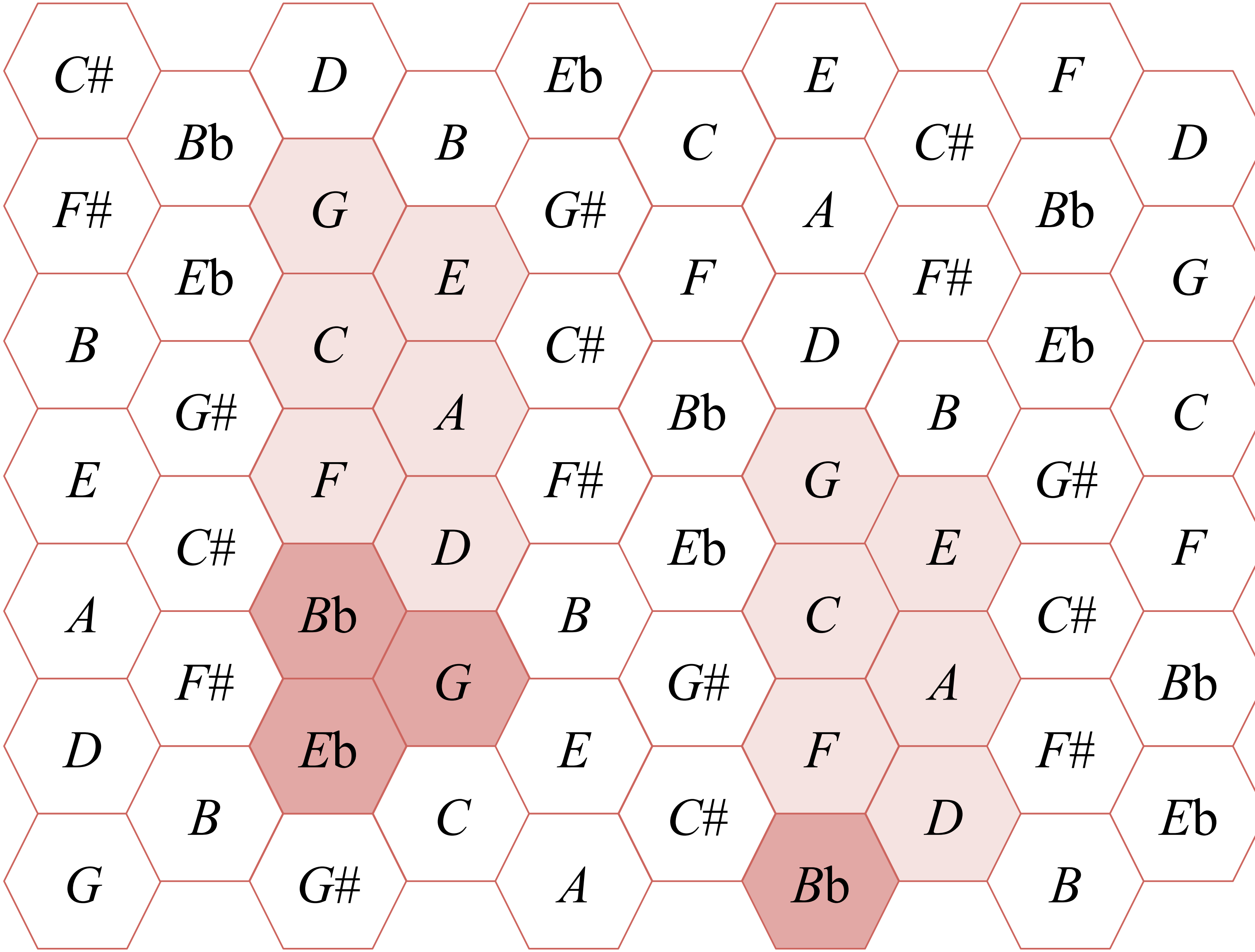
Extract of the 2nd movement of the Symphony No. 9 (L. van Beethoven)



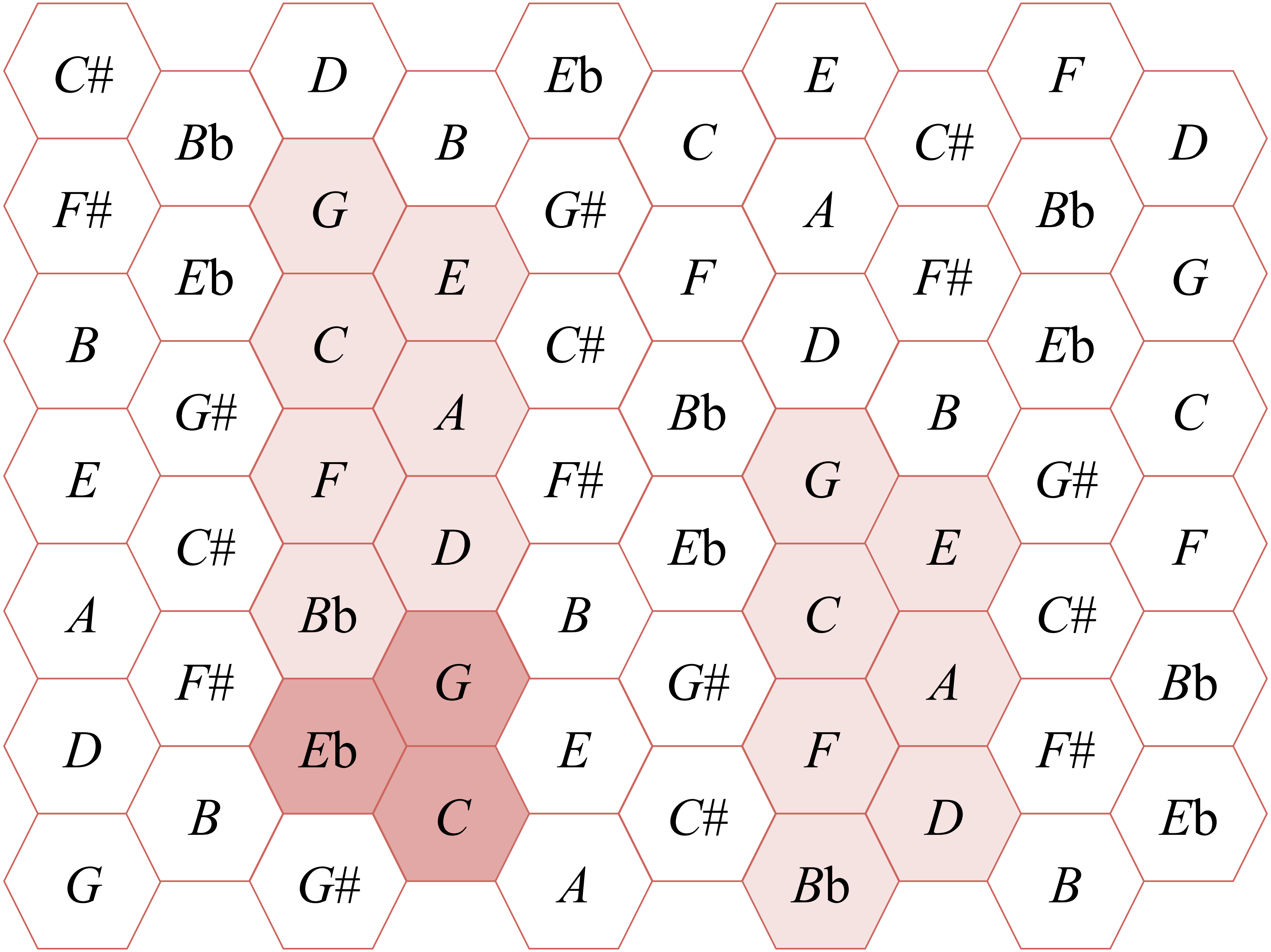
Extract of the 2nd movement of the Symphony No. 9 (L. van Beethoven)



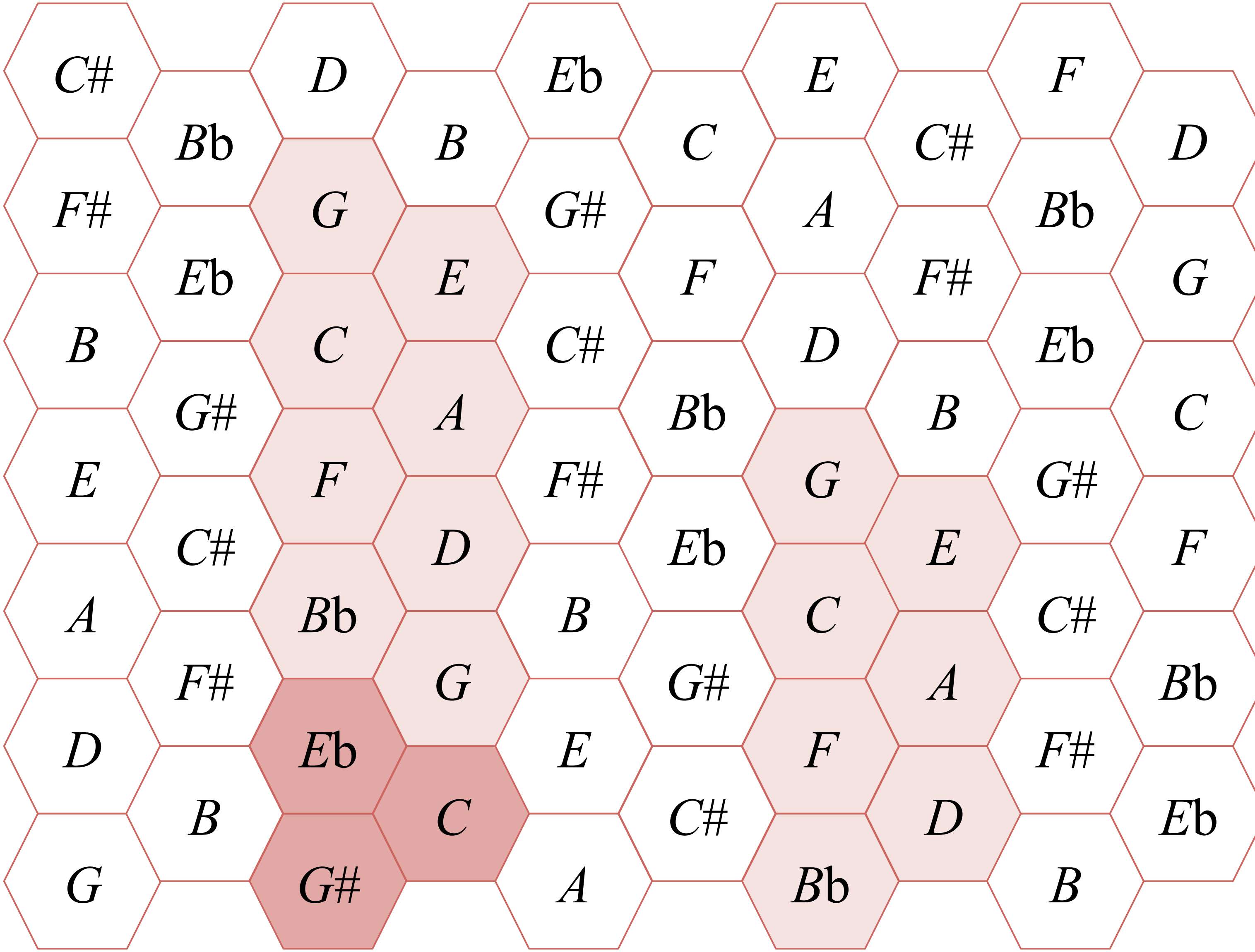
Extract of the 2nd movement of the Symphony No. 9 (L. van Beethoven)



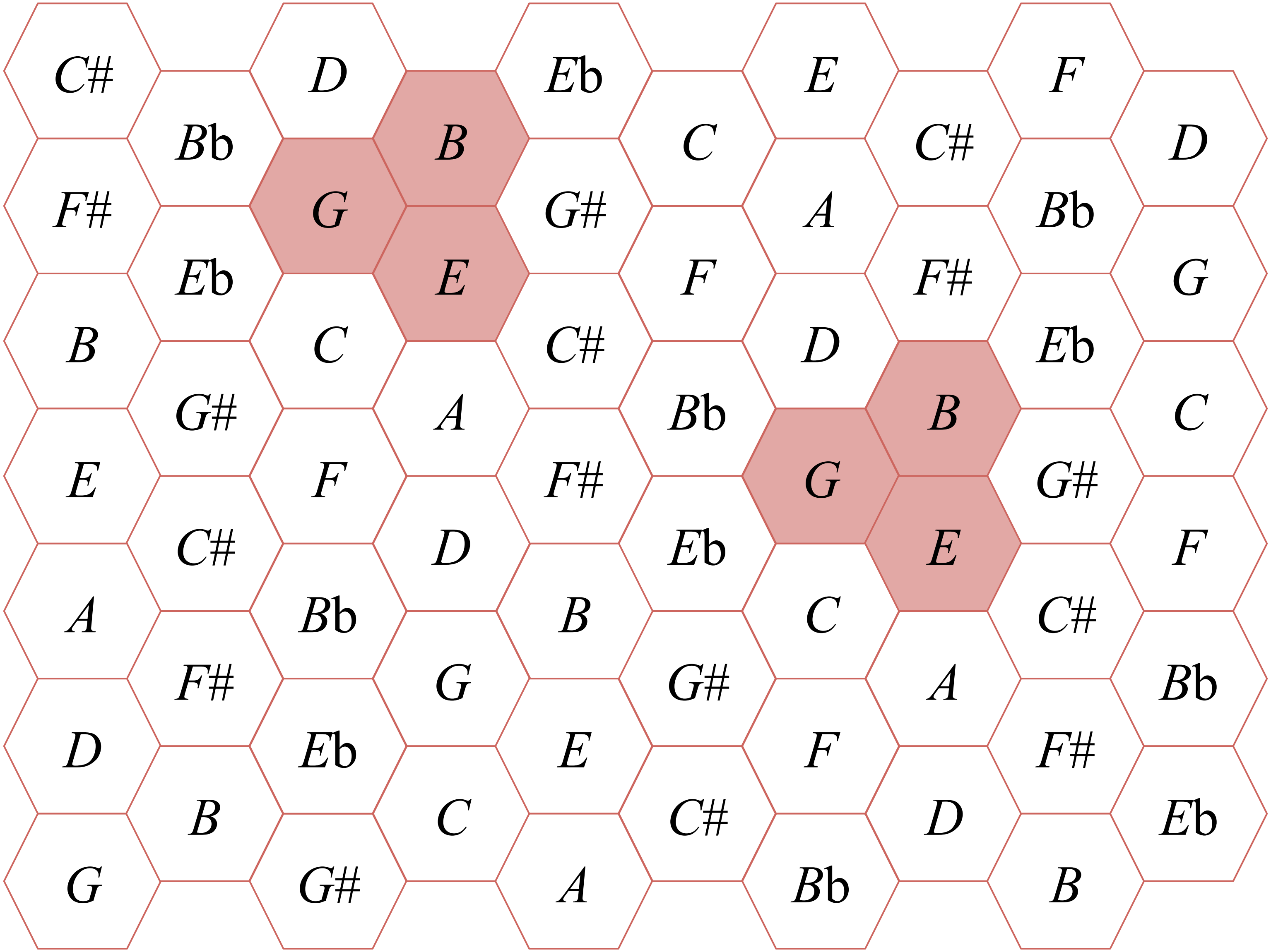
Extract of the 2nd movement of the Symphony No. 9 (L. van Beethoven)



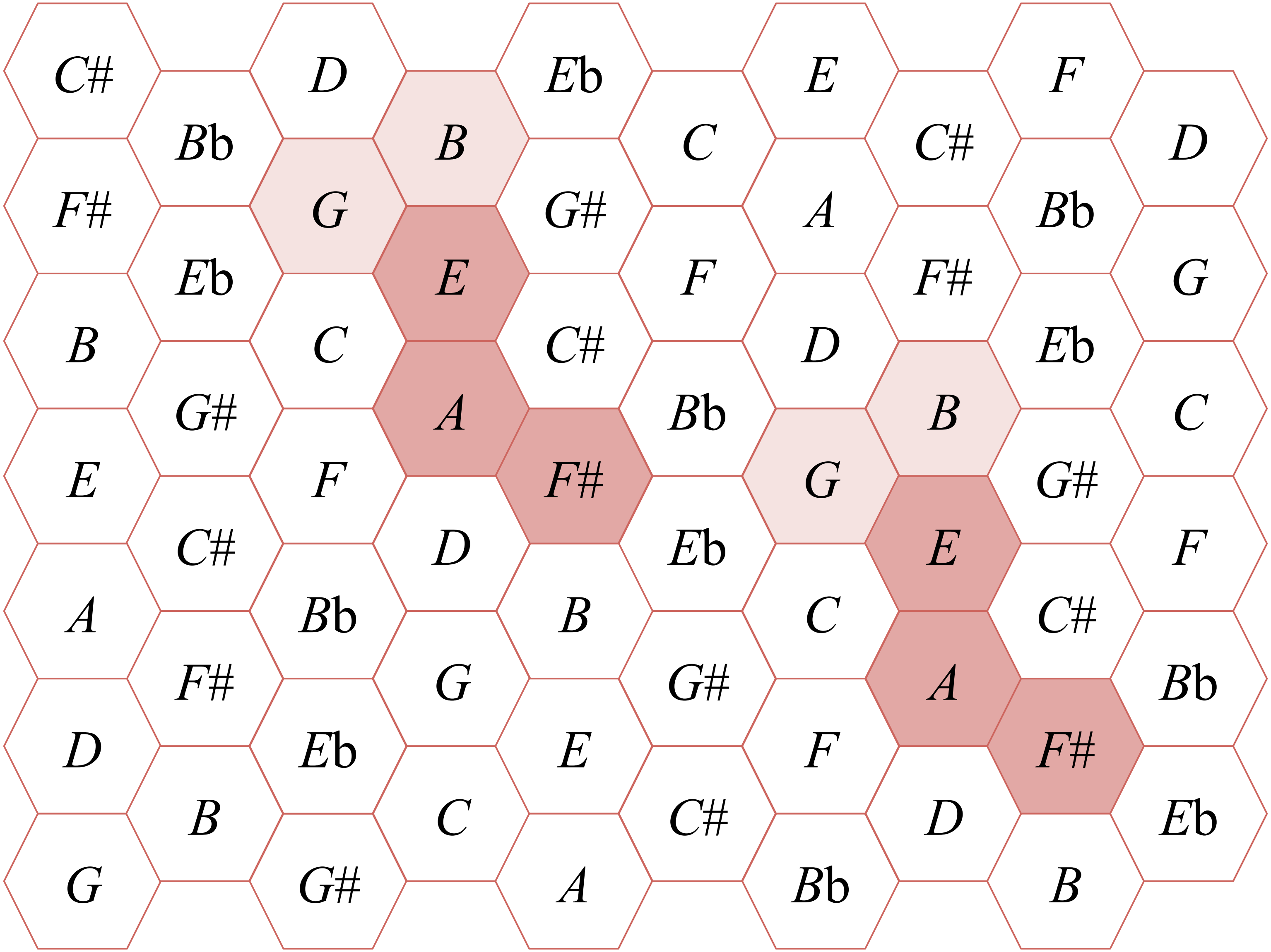
Extract of the 2nd movement of the Symphony No. 9 (L. van Beethoven)



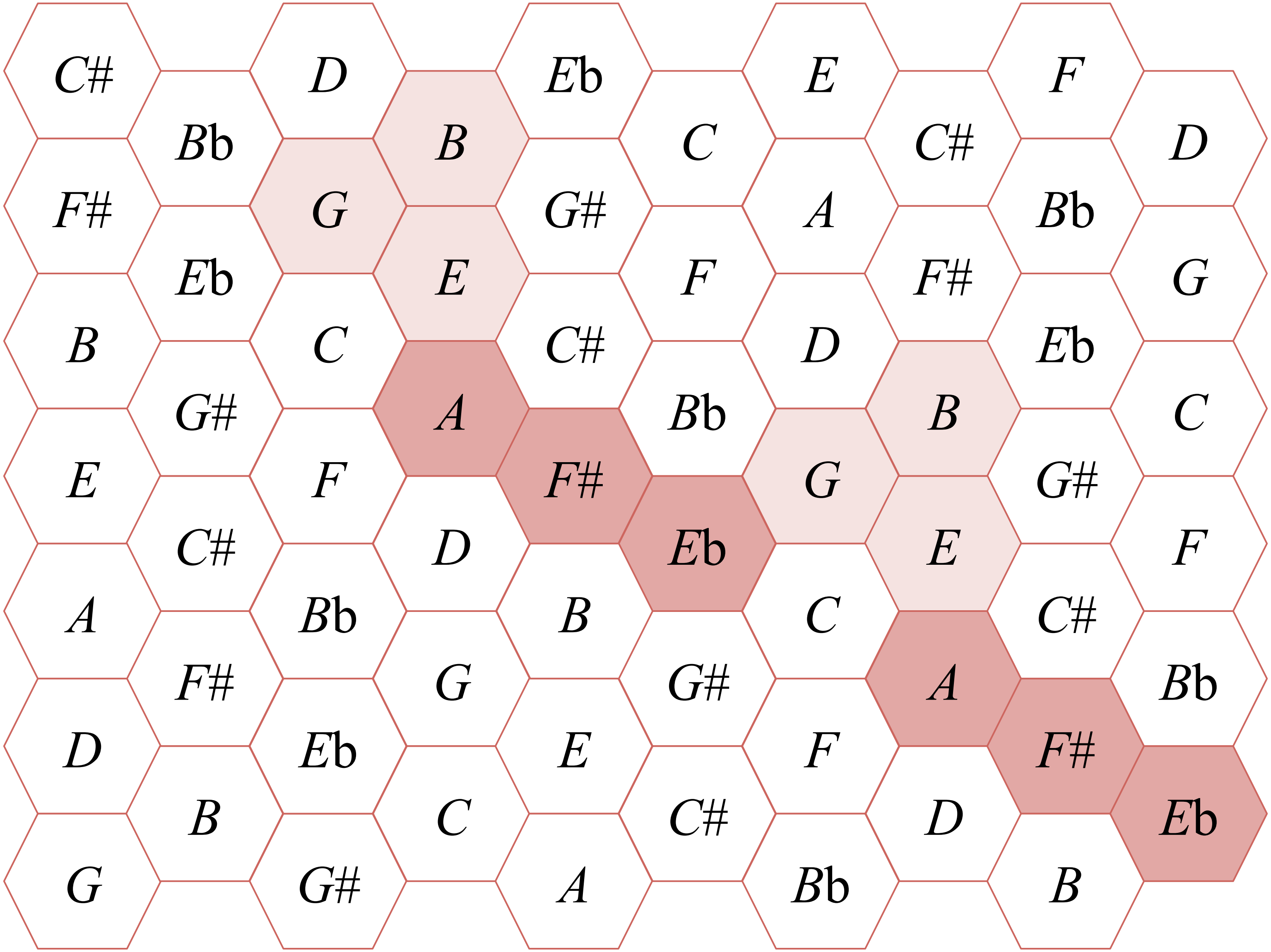
Extract of the Prelude Op.28 N.4
(F. Chopin)



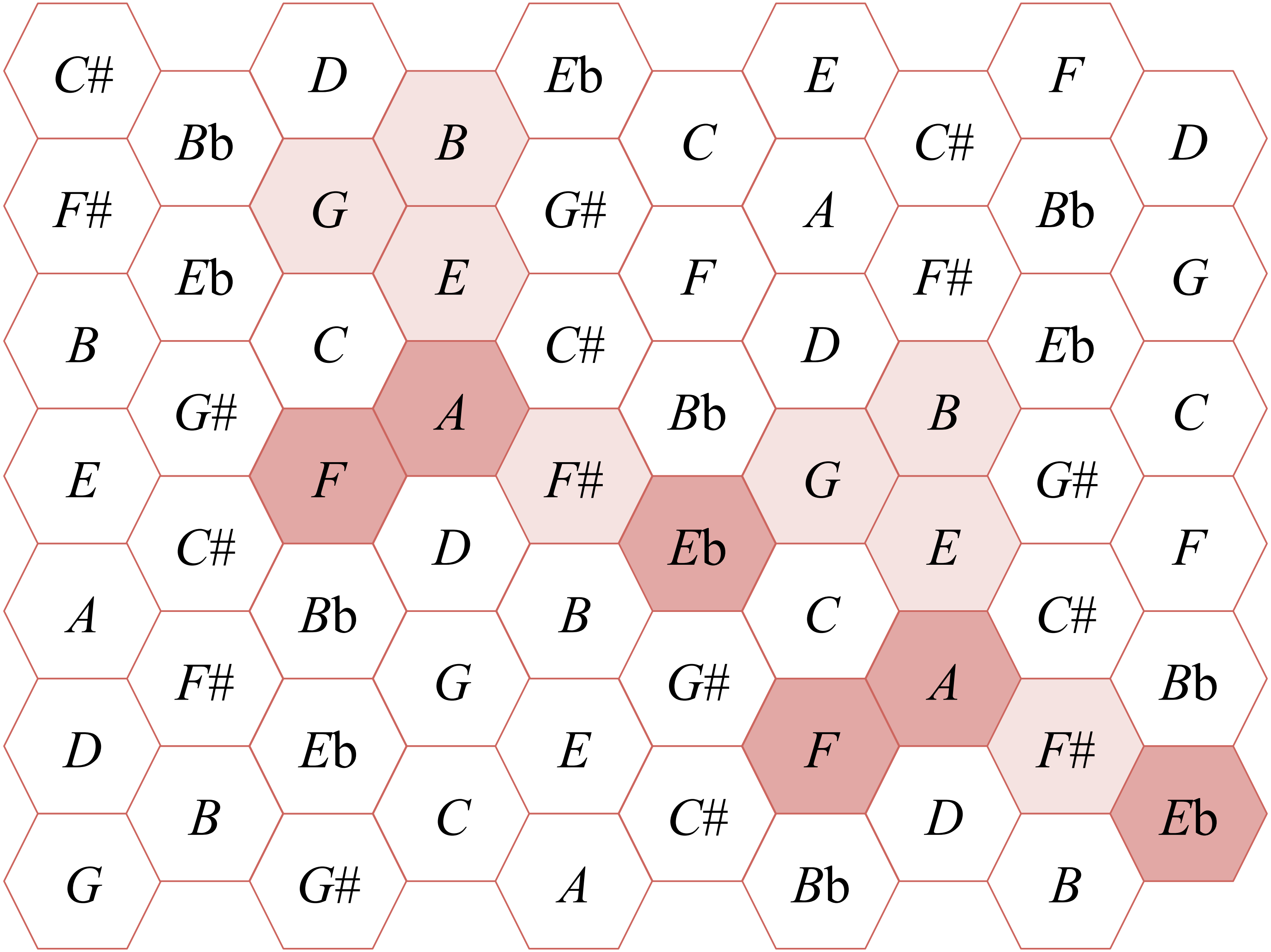
Extract of the Prelude Op.28 N.4
(F. Chopin)



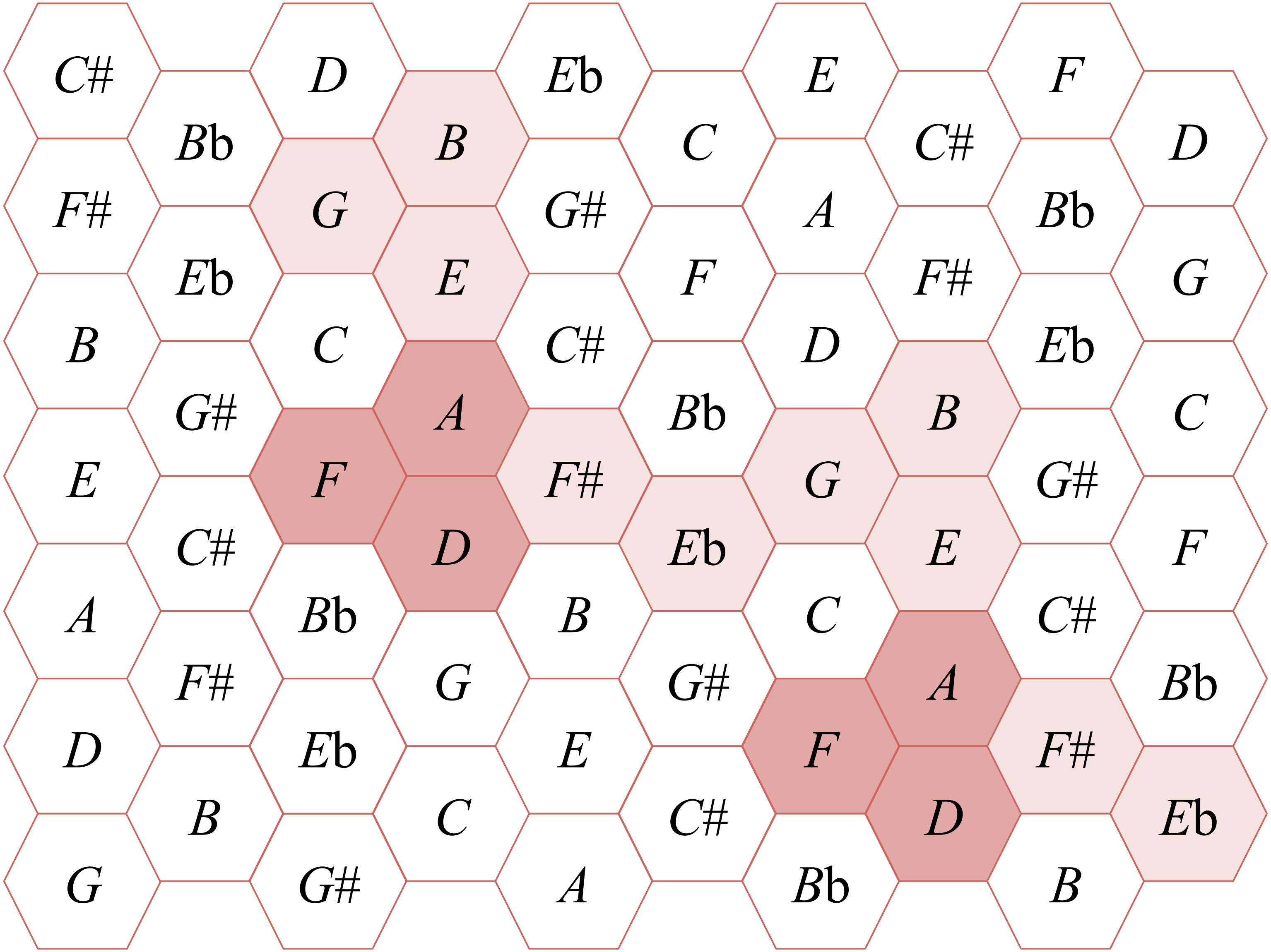
Extract of the Prelude Op.28 N.4
(F. Chopin)



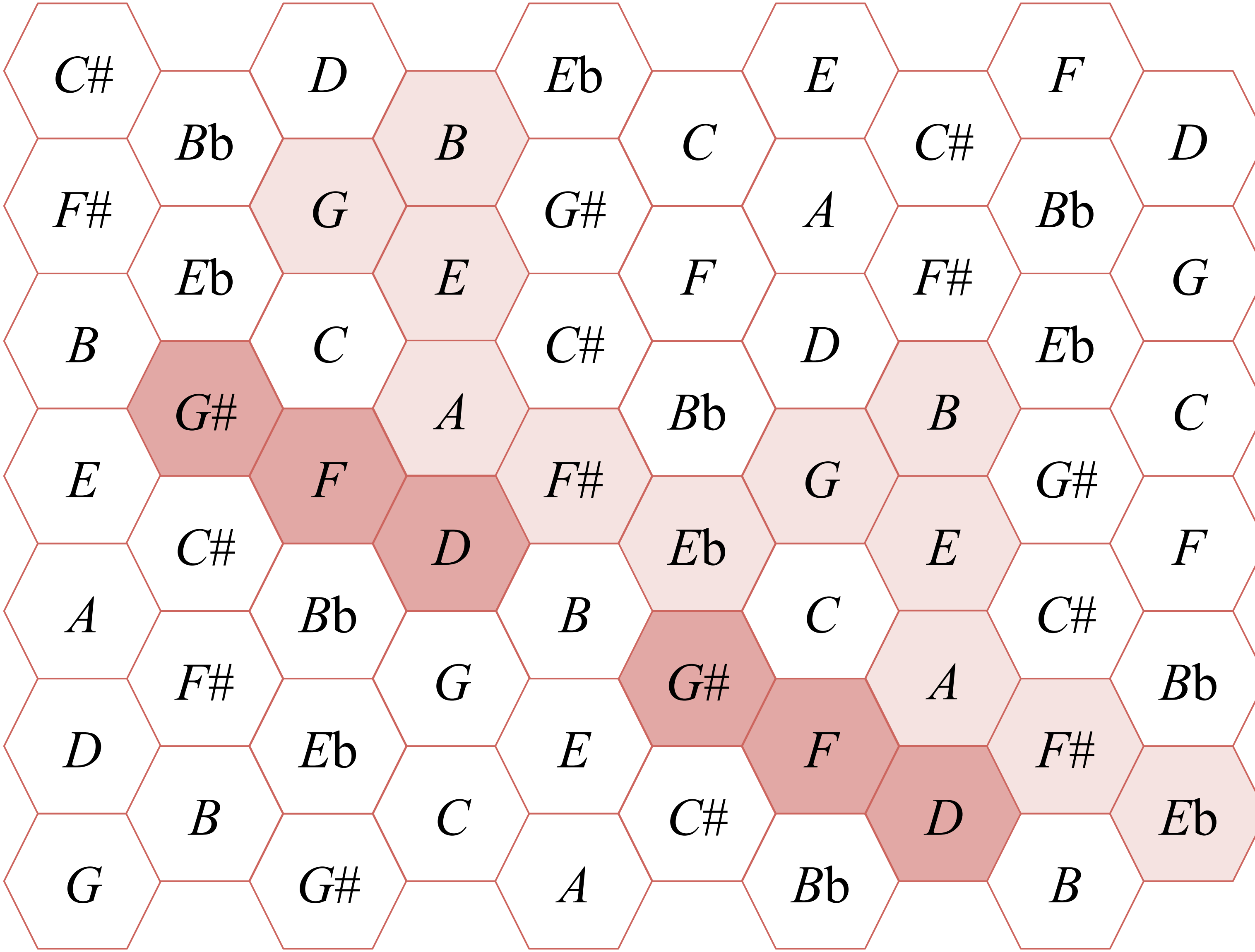
Extract of the Prelude Op.28 N.4
(F. Chopin)



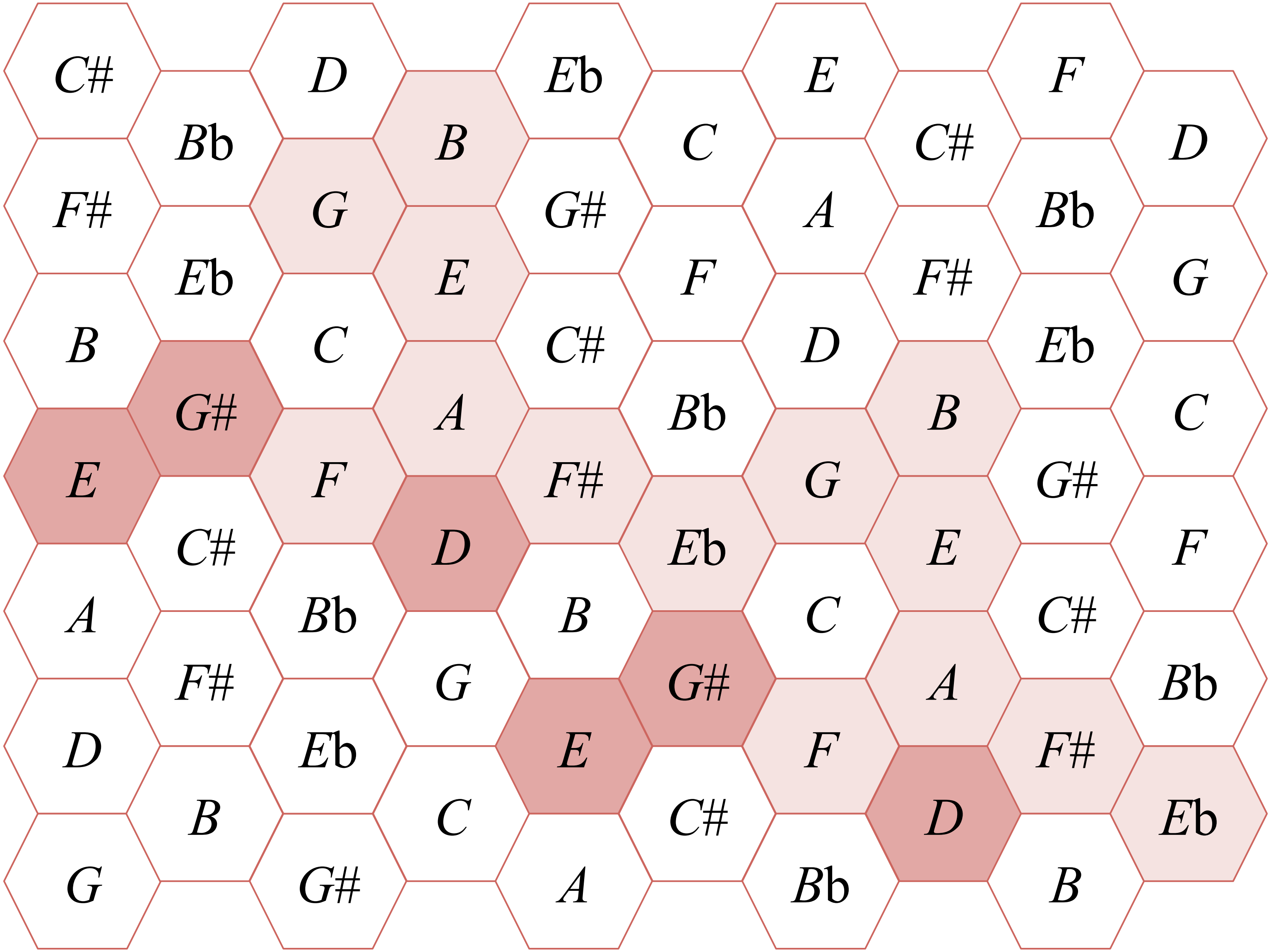
Extract of the Prelude Op.28 N.4
(F. Chopin)



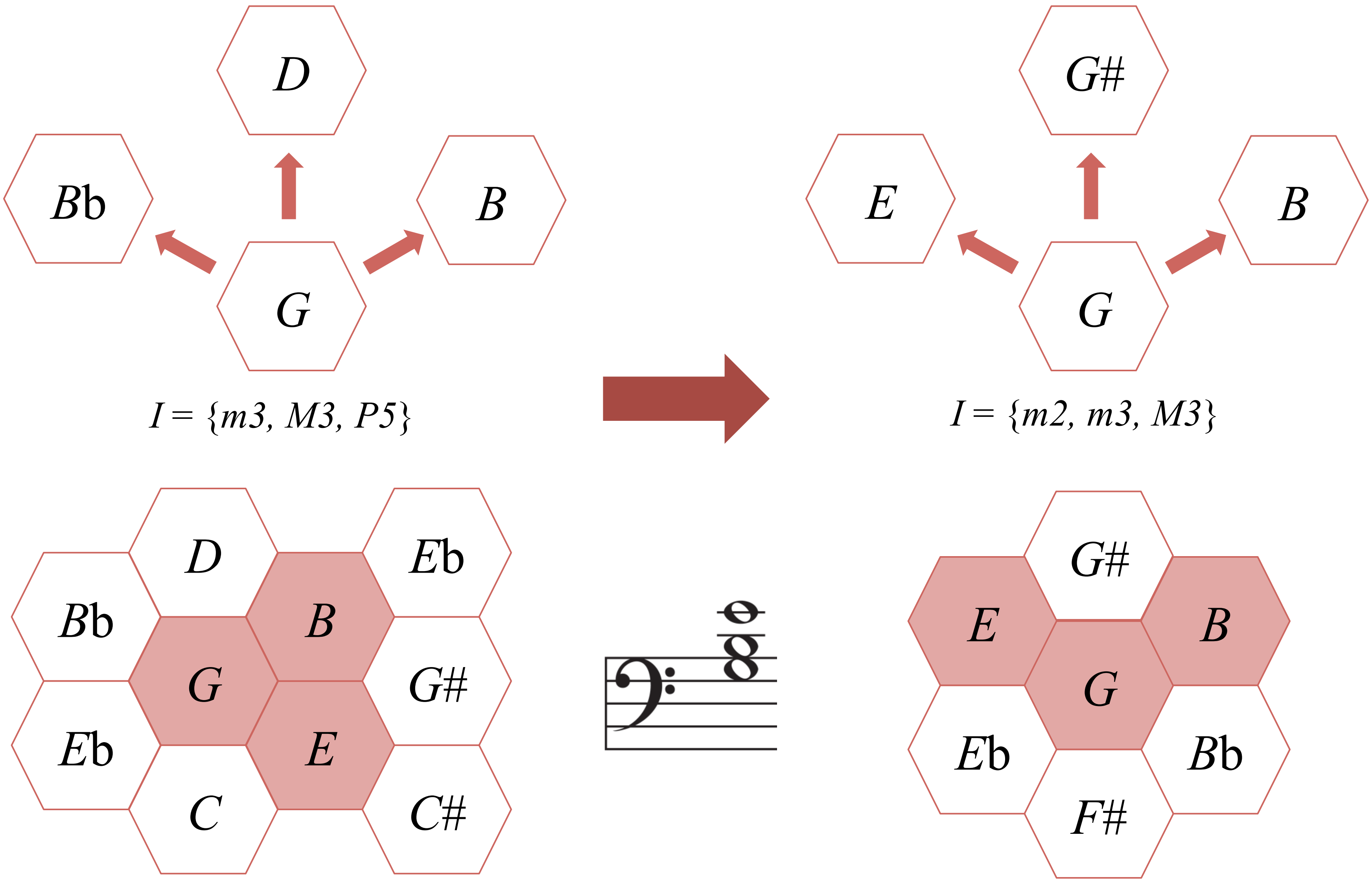
Extract of the Prelude Op.28 N.4
(F. Chopin)



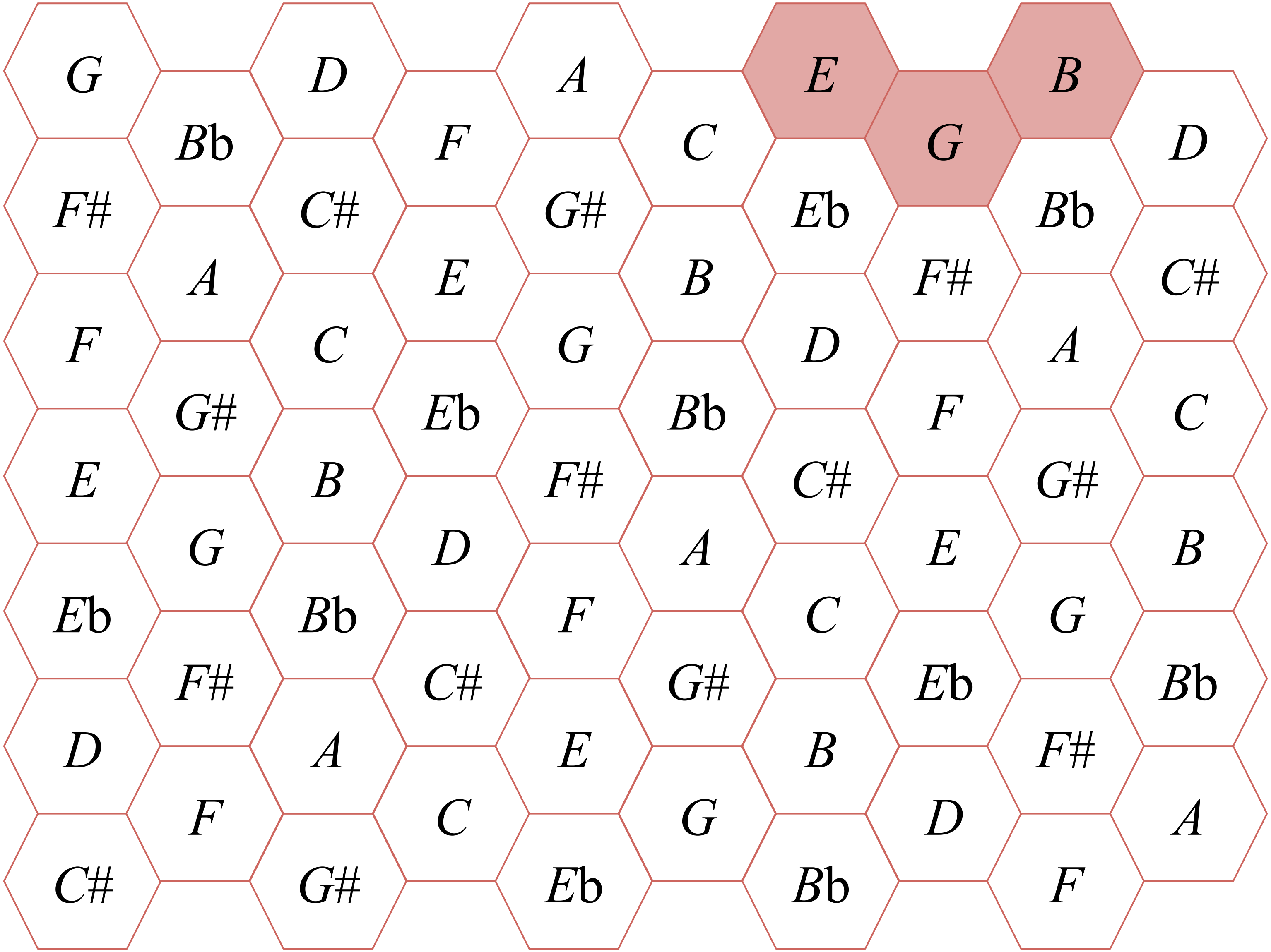
Extract of the Prelude Op.28 N.4
(F. Chopin)



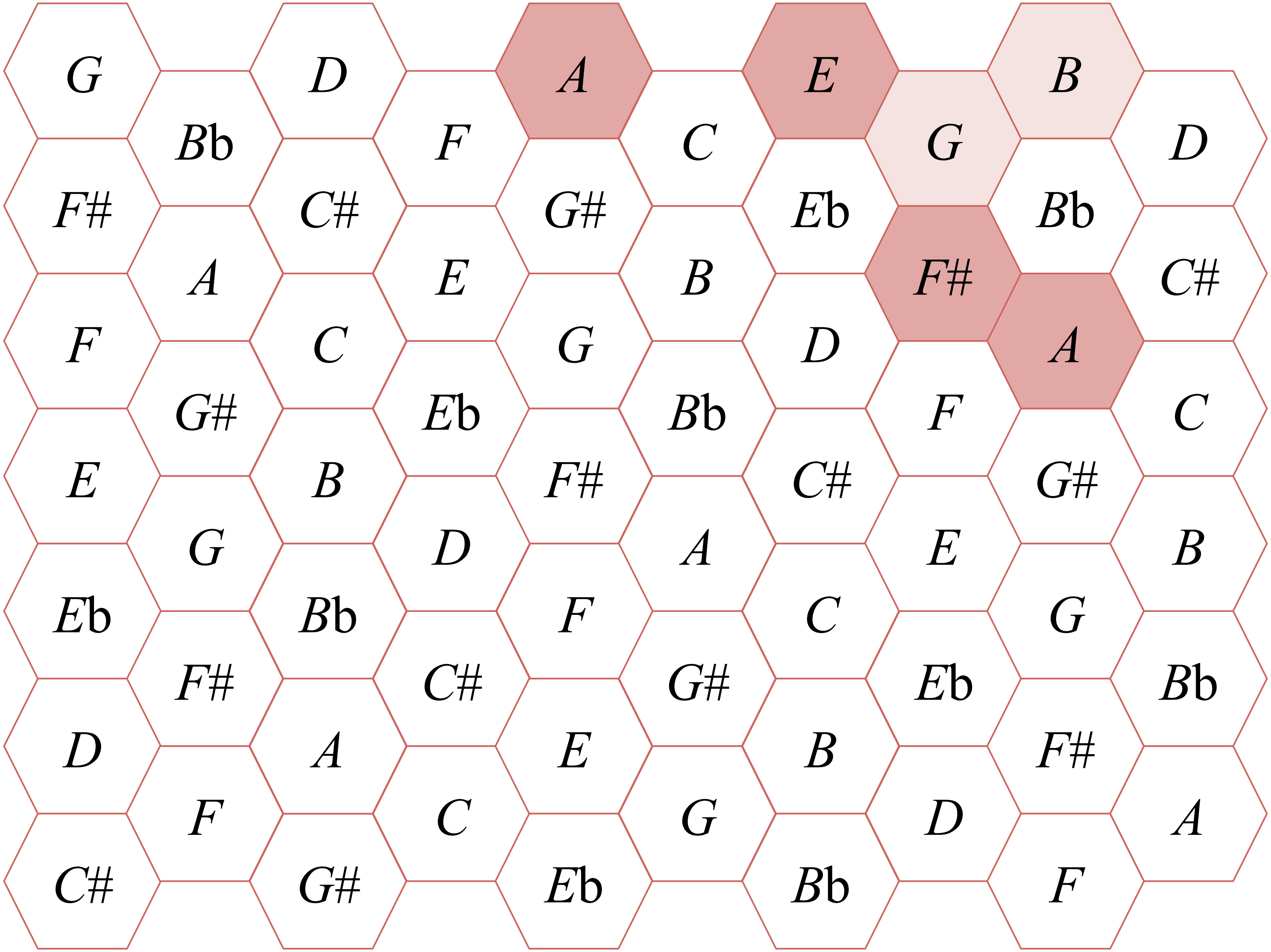
Extract of the Prelude Op.28 N.4
(F. Chopin)



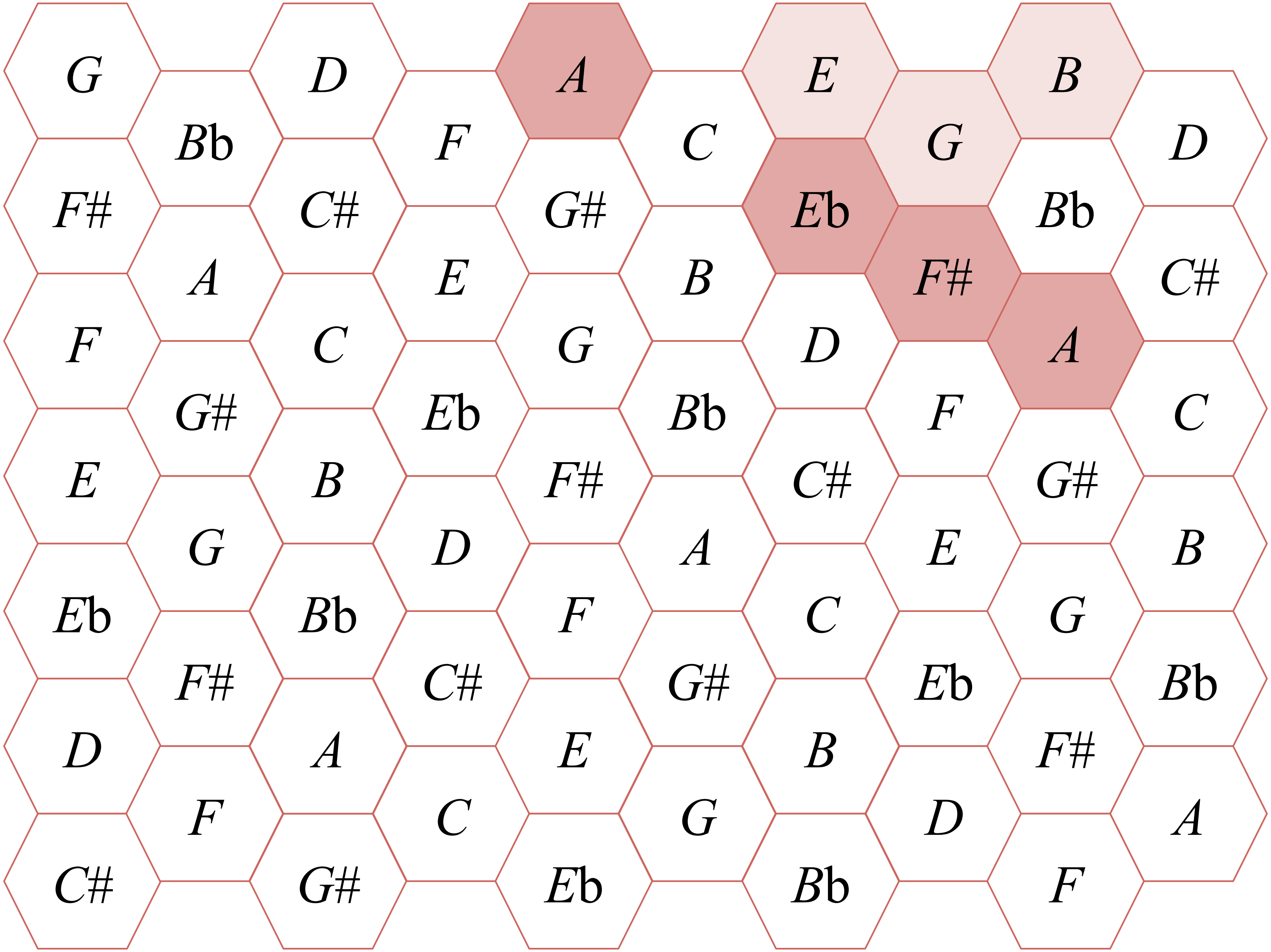
Extract of the Prelude Op.28 N.4
(F. Chopin)



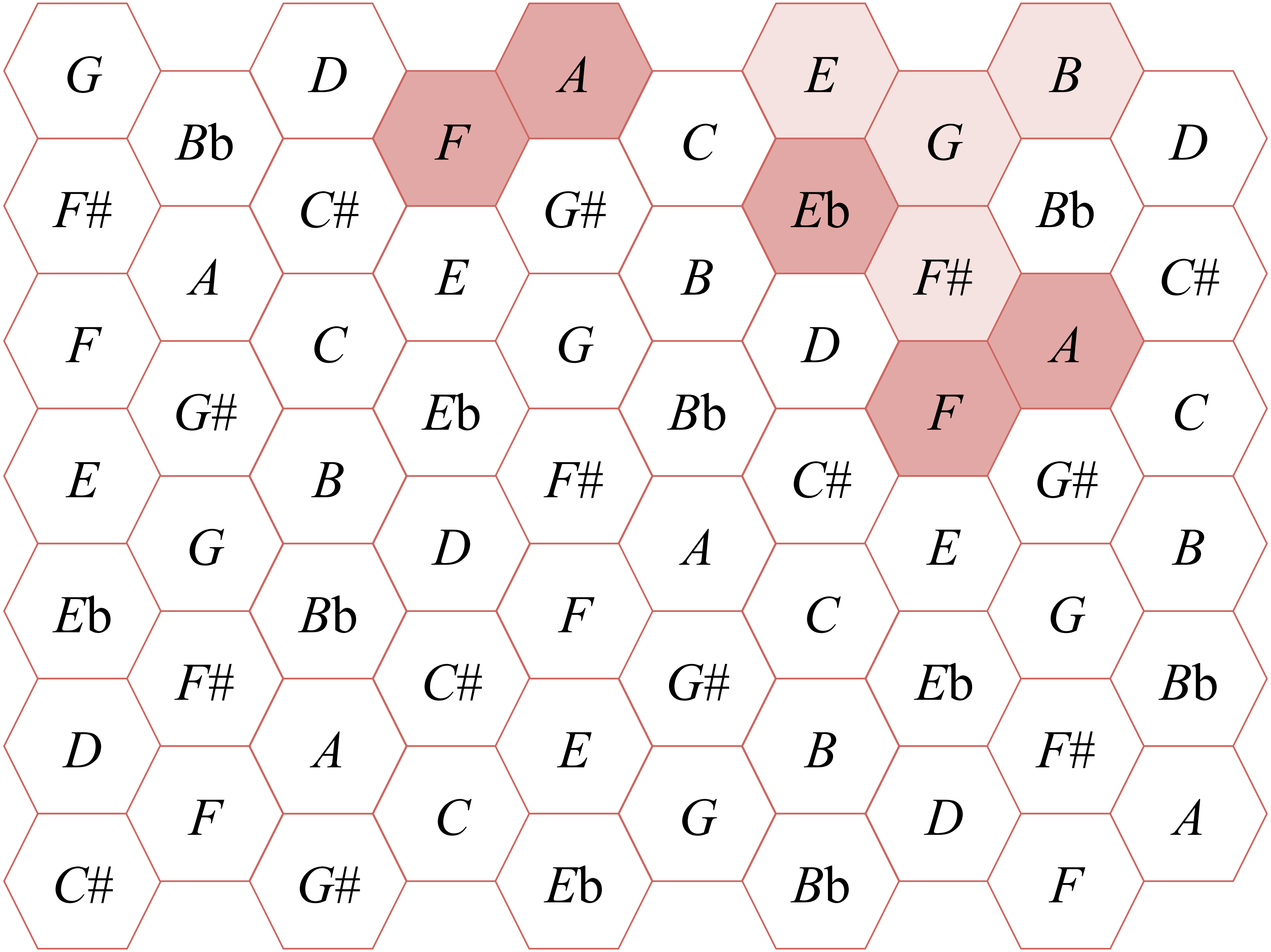
Extract of the Prelude Op.28 N.4
(F. Chopin)



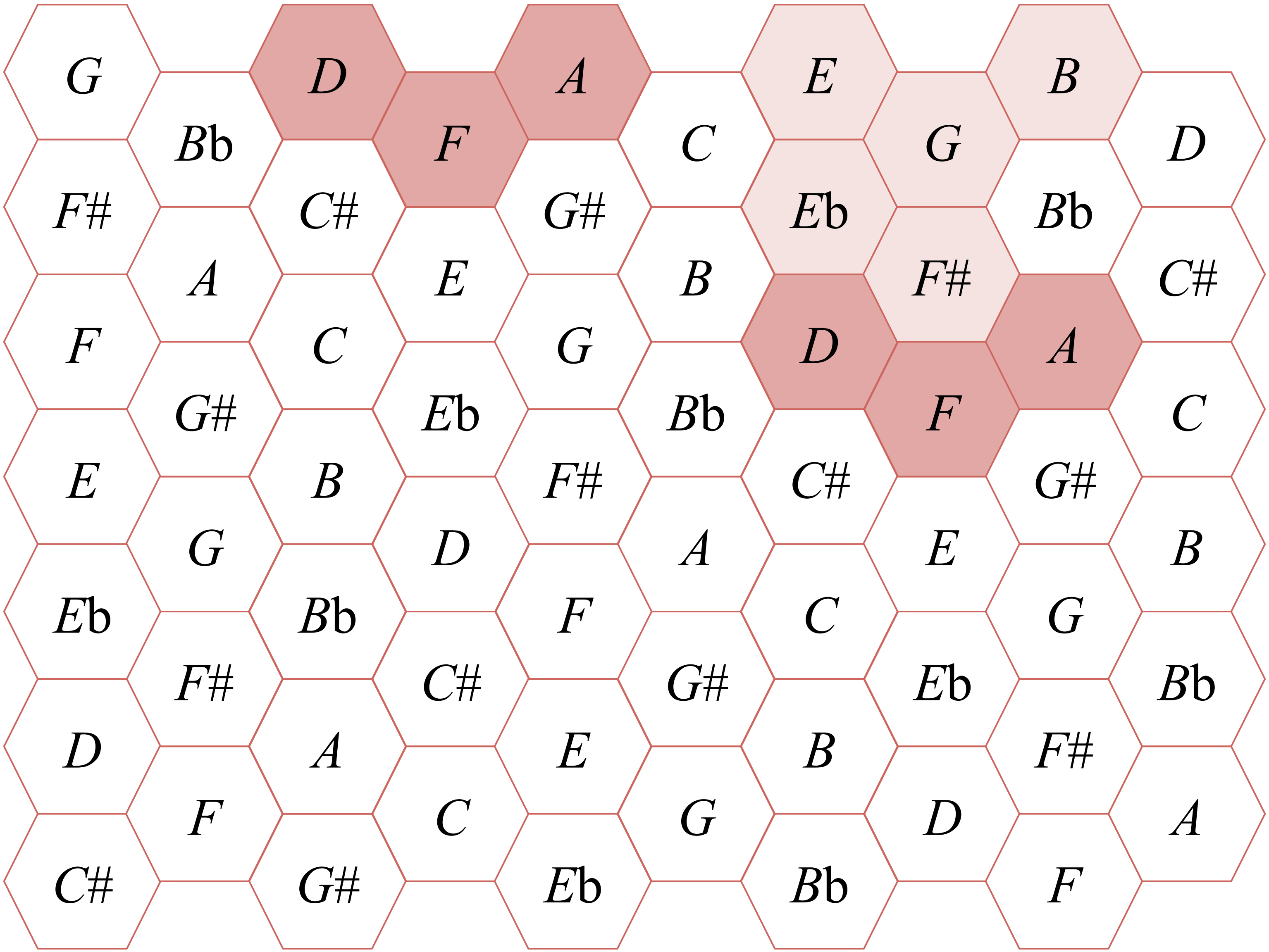
Extract of the Prelude Op.28 N.4
(F. Chopin)



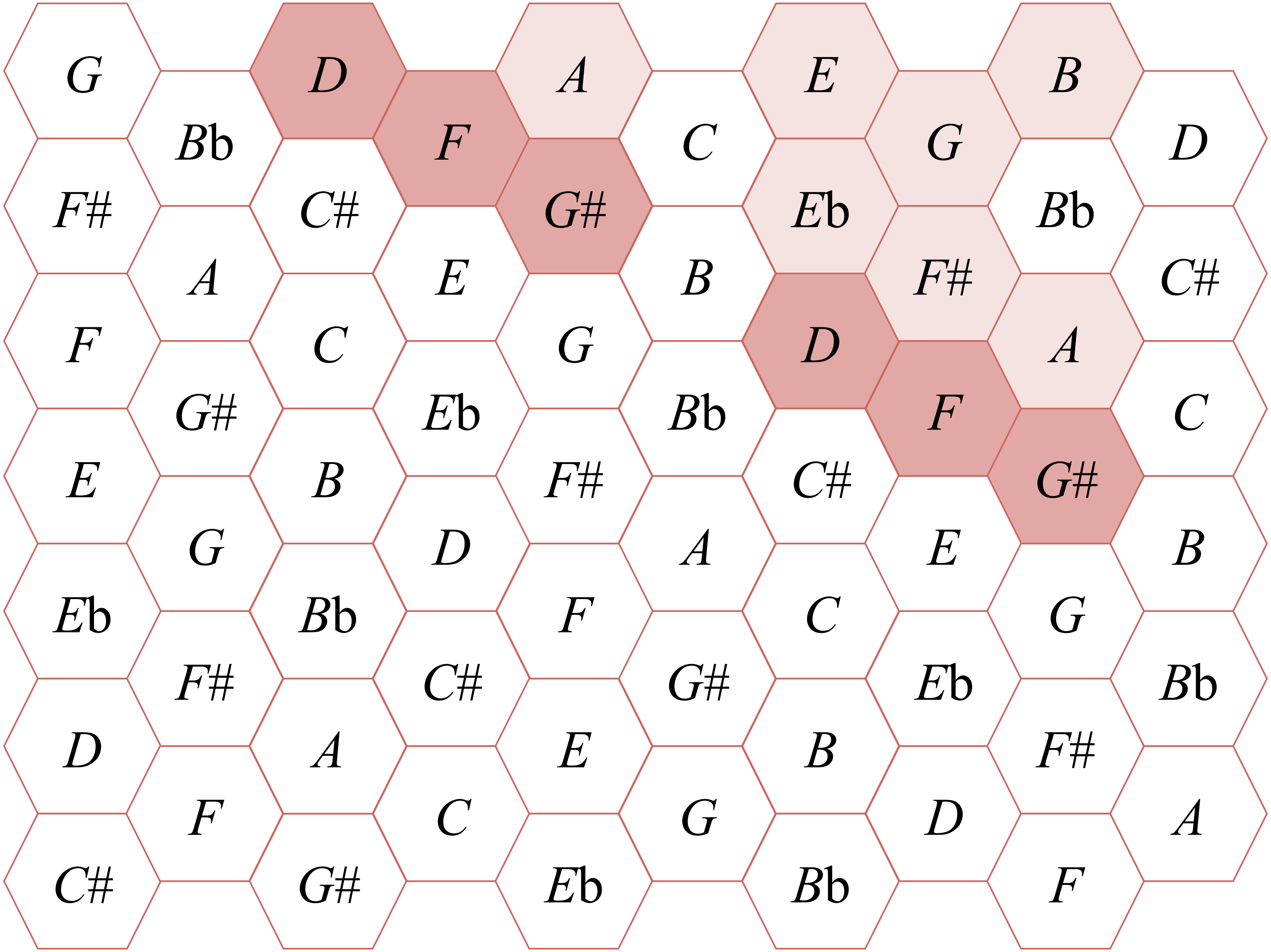
Extract of the Prelude Op.28 N.4
(F. Chopin)



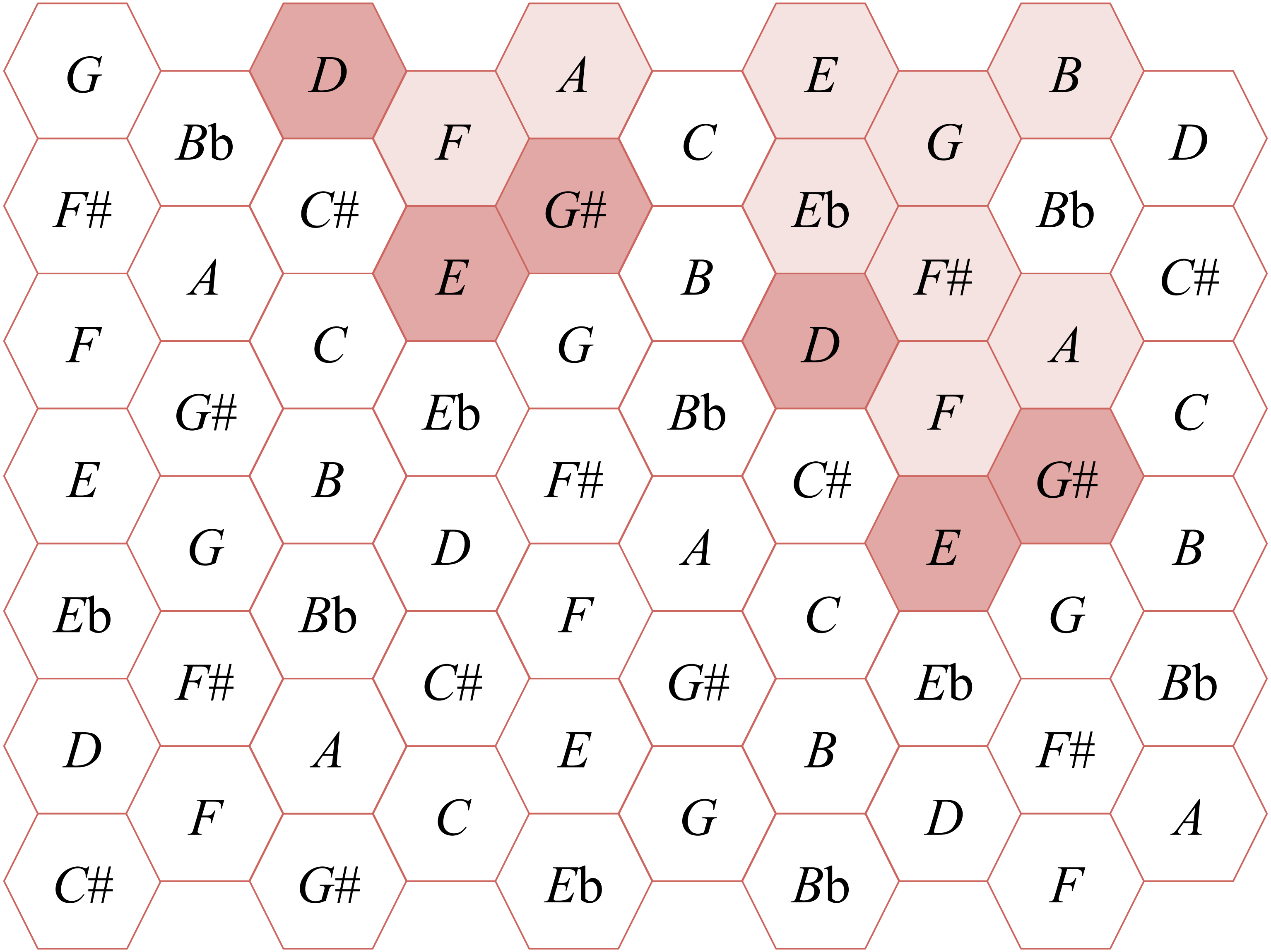
Extract of the Prelude Op.28 N.4
(F. Chopin)



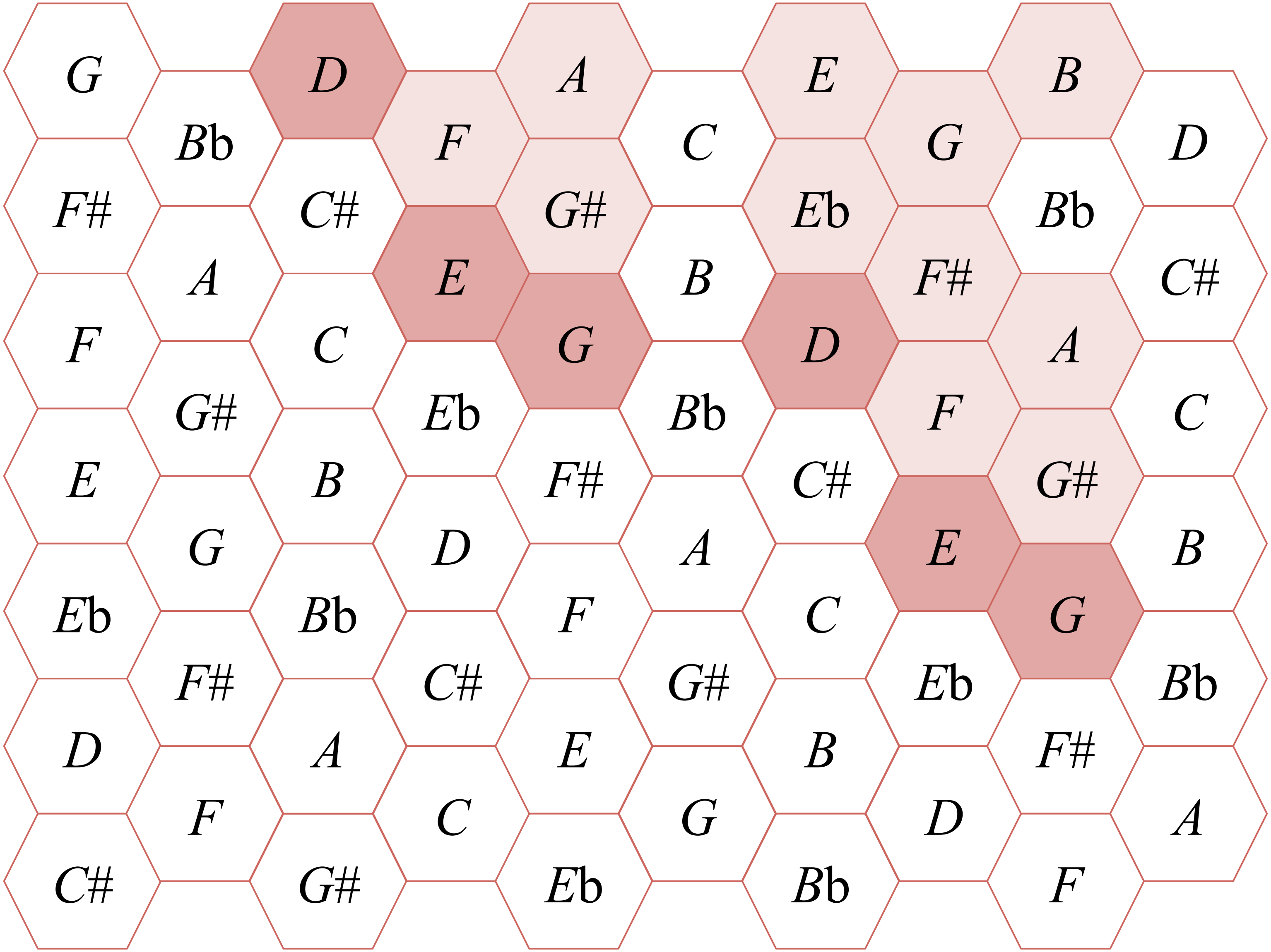
Extract of the Prelude Op.28 N.4
(F. Chopin)



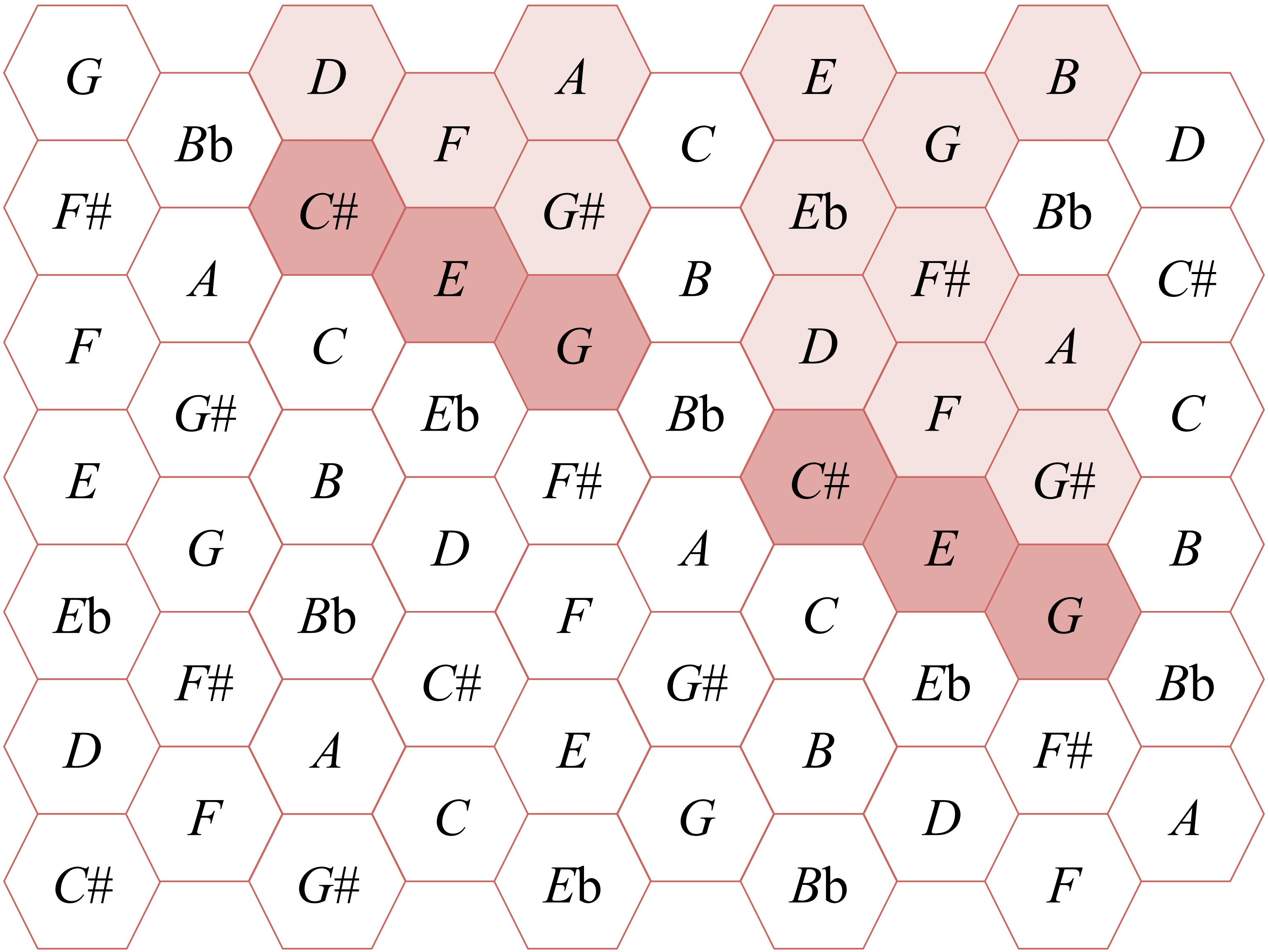
Extract of the Prelude Op.28 N.4
(F. Chopin)



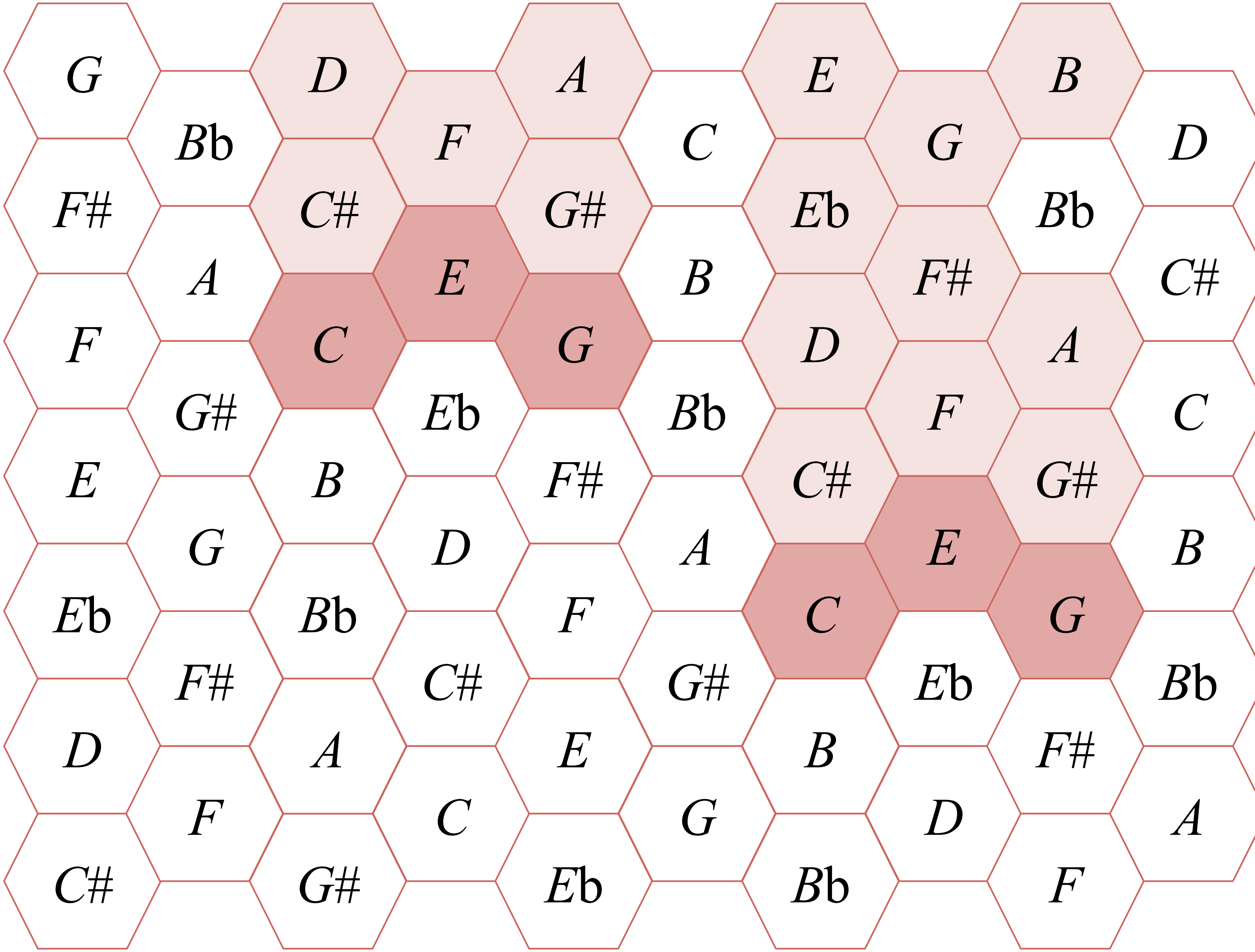
Extract of the Prelude Op.28 N.4
(F. Chopin)



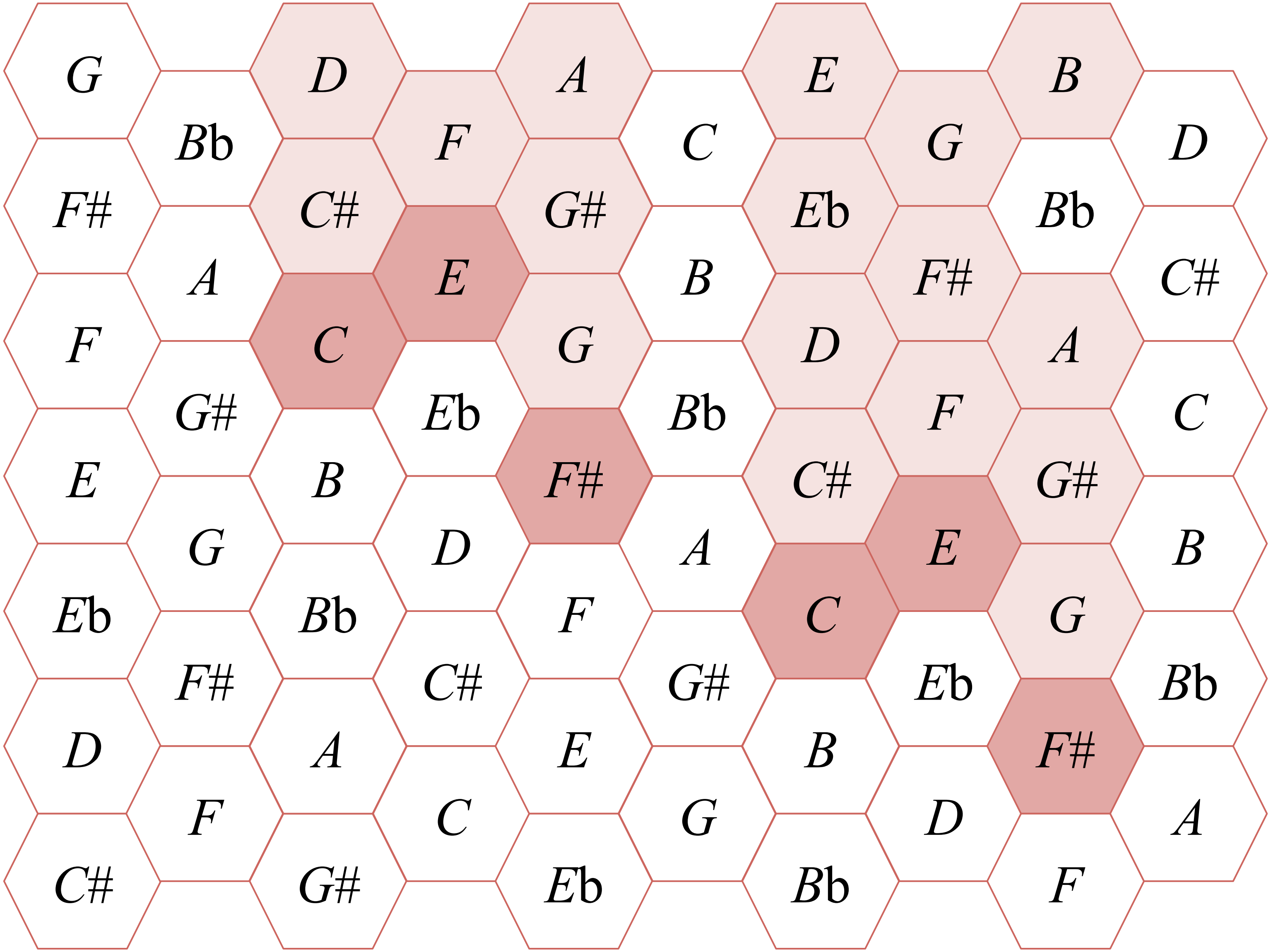
Extract of the Prelude Op.28 N.4
(F. Chopin)



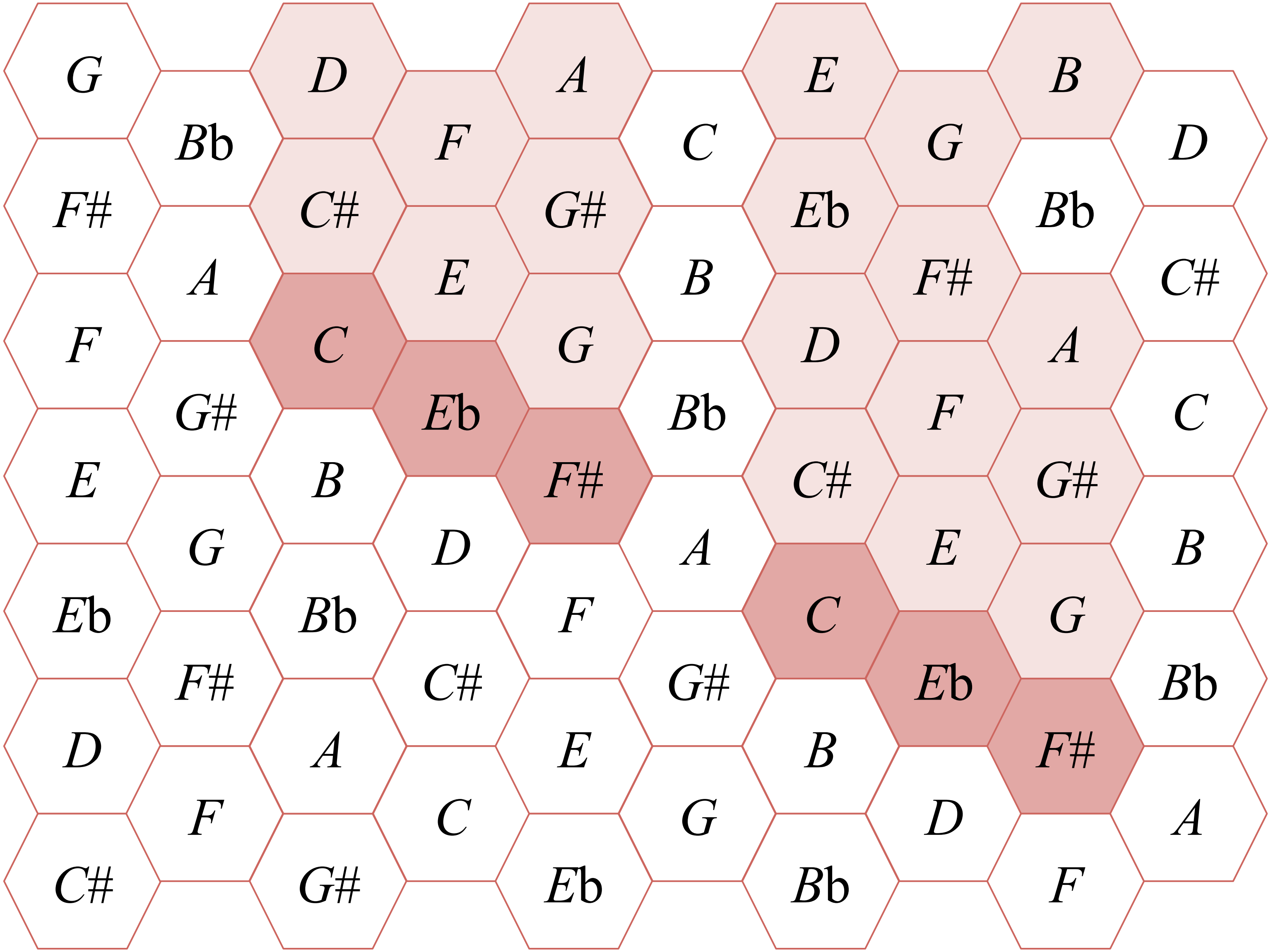
Extract of the Prelude Op.28 N.4
(F. Chopin)



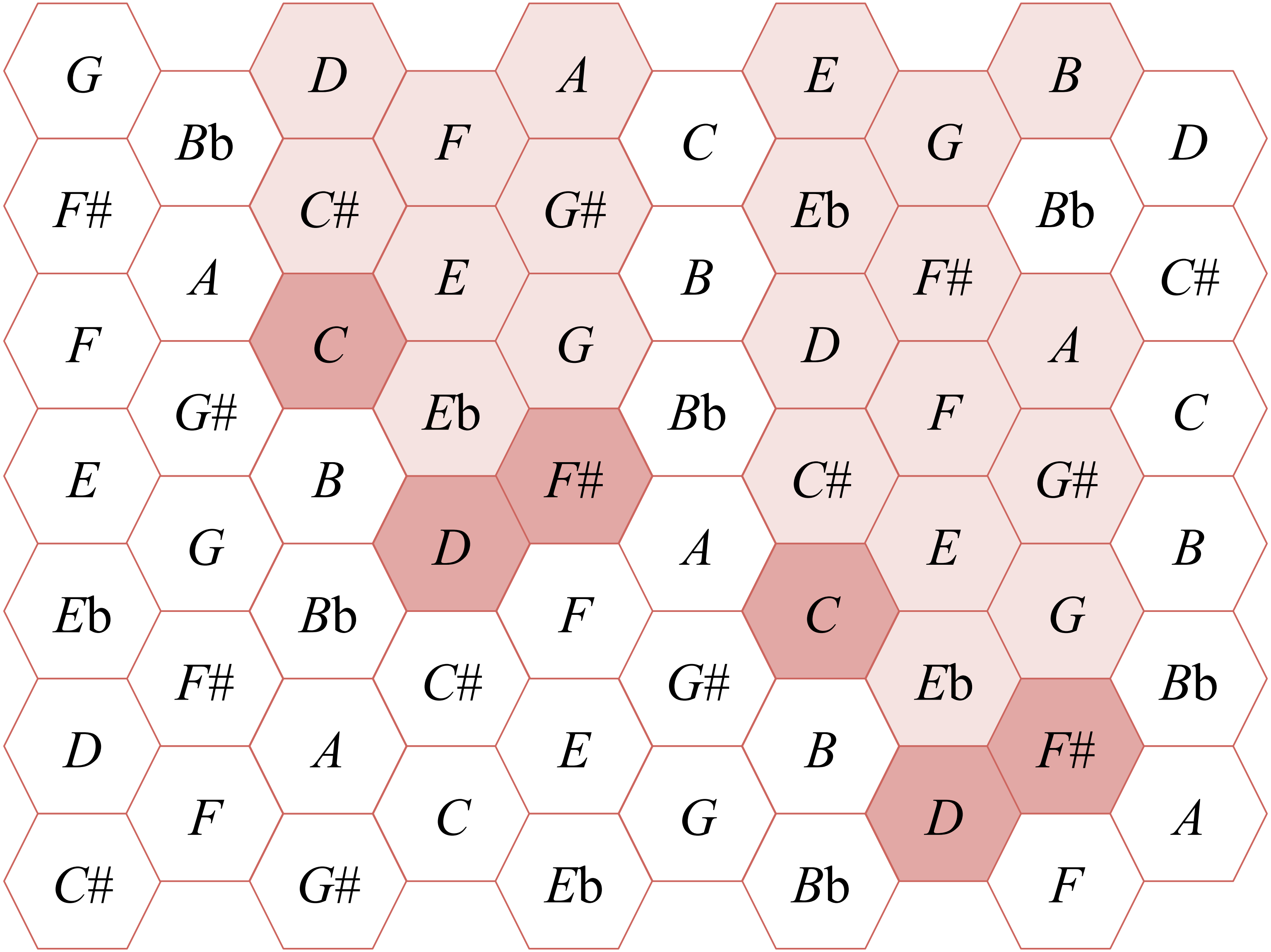
Extract of the Prelude Op.28 N.4
(F. Chopin)



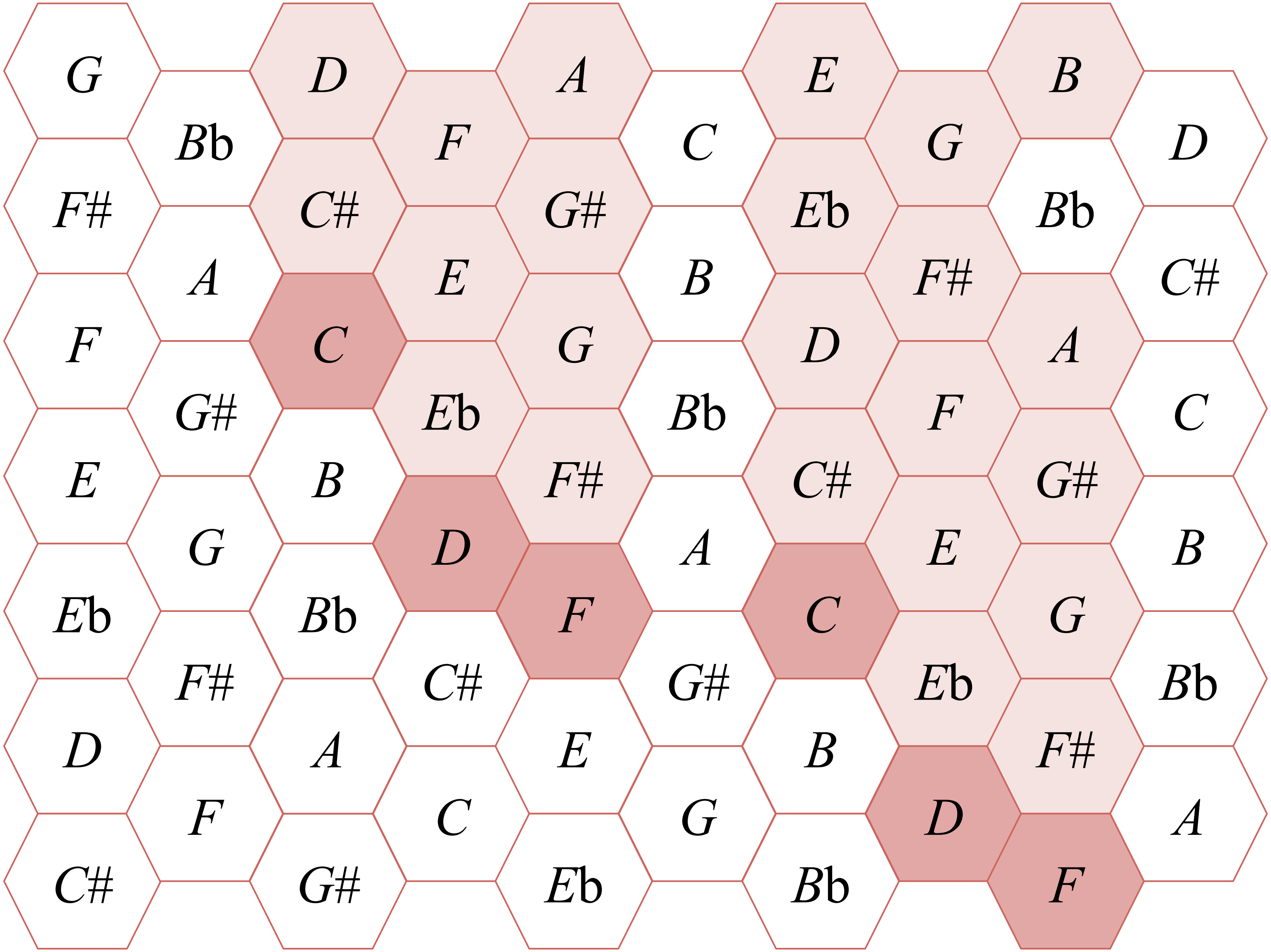
Extract of the Prelude Op.28 N.4
(F. Chopin)



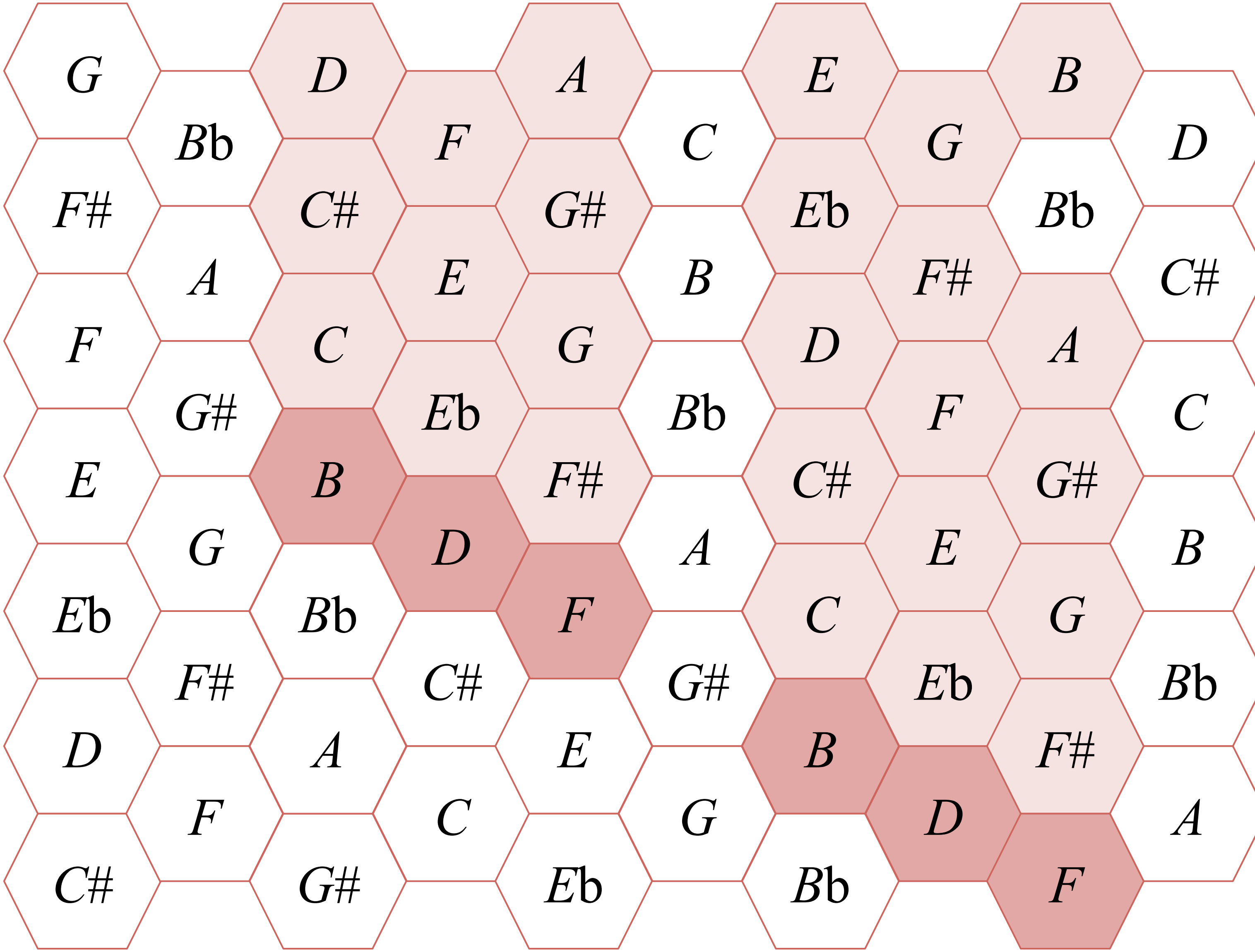
Extract of the Prelude Op.28 N.4
(F. Chopin)

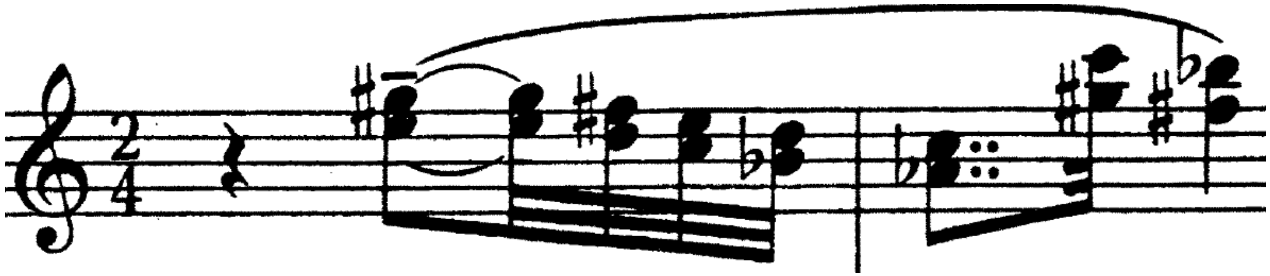


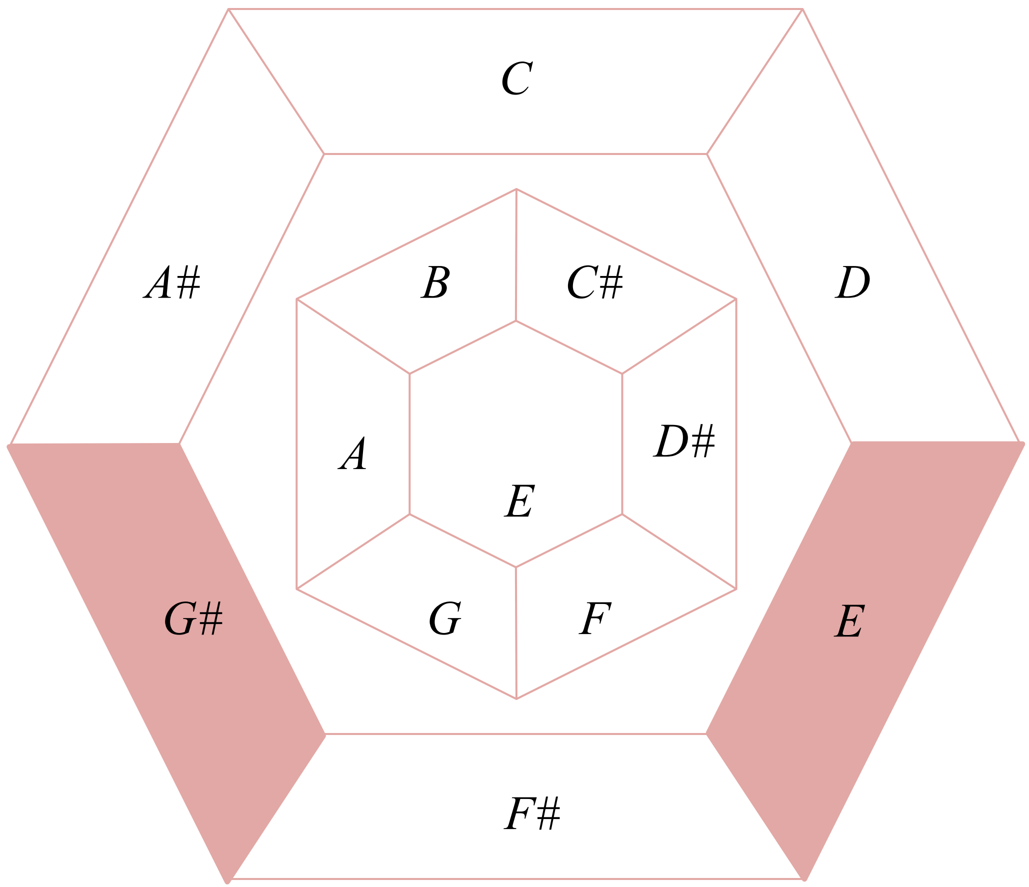
Extract of the Prelude Op.28 N.4
(F. Chopin)

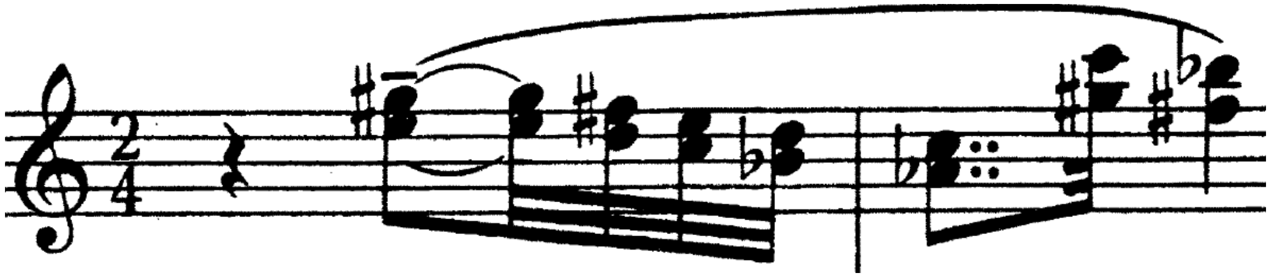


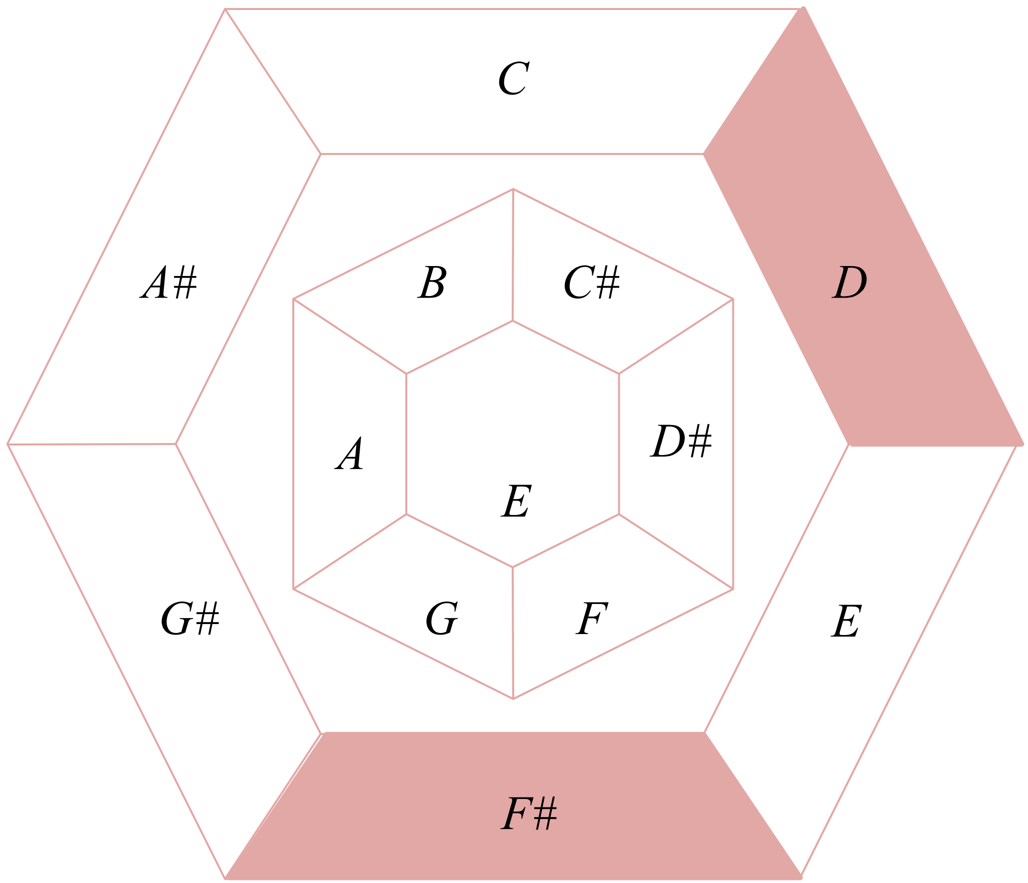
Extract of the Prelude Op.28 N.4
(F. Chopin)

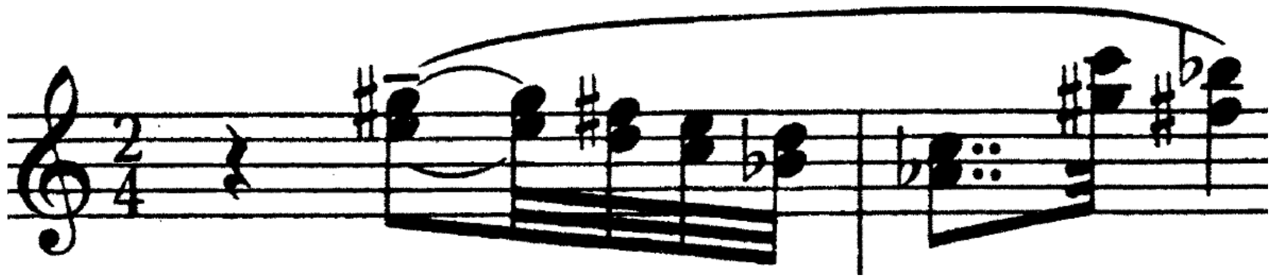


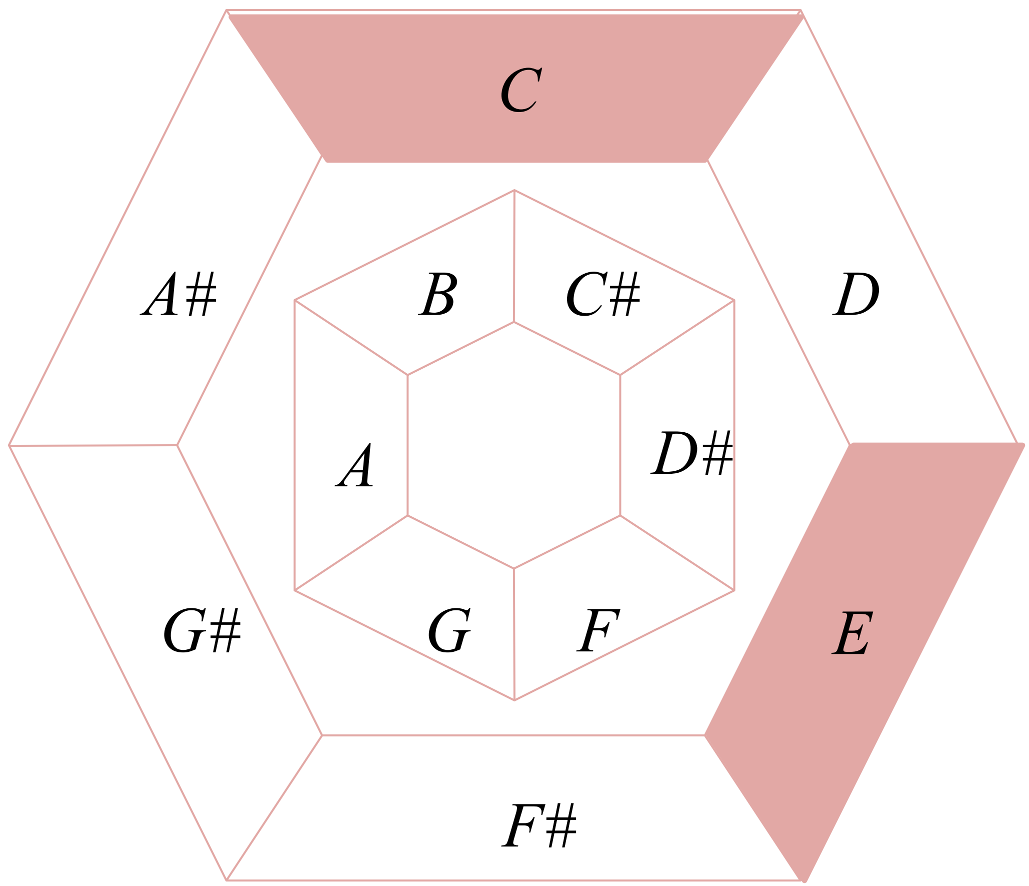



C
A#
B
C#
D
A
E
D#
G#
G
F
E
F#




C
A#
D
B
C#
A
E
D#
G#
G
F
E
F#




C
A#
B
C#
D
A
D#
G#
G
F
E
F#

Spatial Computing & MGS

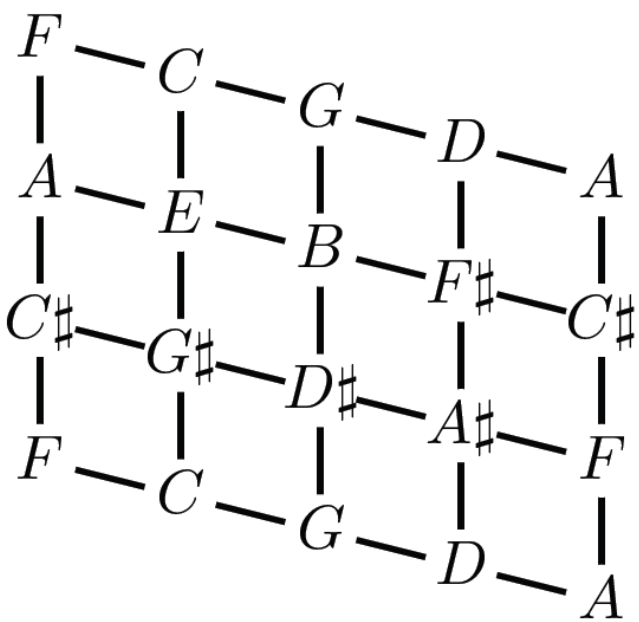
- Spatial computing
  - Compute in space
  - Compute space

- MGS: a programming language for spatial computing
  - Introduction of topological concepts in a programming language
  - Two main principles
    - Data structure: topological collection
    - Control structure: transformation

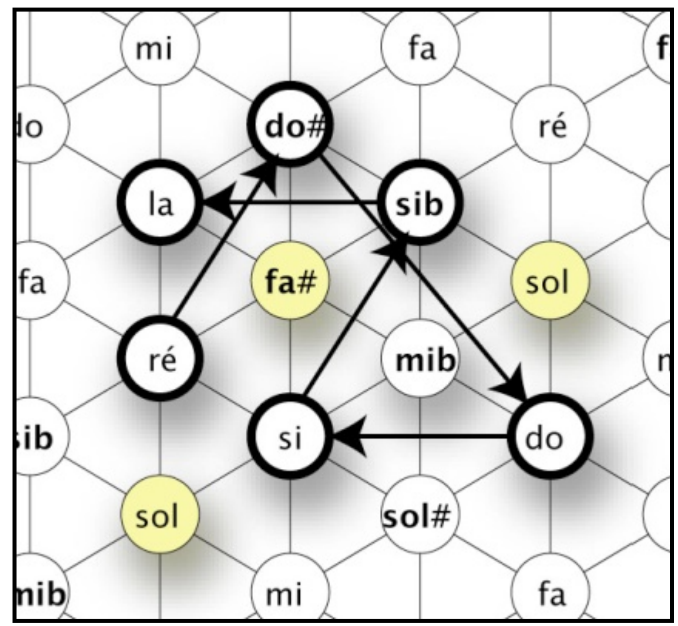
Neo-Riemannian Problematic

- Examples

  Neo-Riemannian approach for tonal music



Euler's tonnetz



Hexagonal network of notes
(J.-M. Chouvel)