Two Conditions Suffice to Derive the Seven Modes from the Twelve-Tone-System

Nicolas Hamel

The seven diatonic modes exhibit three well-known properties that appear to be particularly important and implicating their strong tonal coherence within our modern musical system: (1) All scale tones can be arranged in successive fifths; (2) Only exactly one triad (exclusively constructed by notes of the scale) can be built on each scale tone; (3) The interval of the tritone occurs exactly once and has exactly one possible resolution within the scale. In this study, these properties were imposed as conditions on the set of 2048 transposition-classes representing all distinct scales that can possibly be built by using one to twelve tones from the twelve-tone system. Interestingly, the imposition of any two of the three conditions turns out to be sufficient to derive the seven modes from that set of distinct scales. This result is due to the fact that no other scale within the twelve-tone system simultaneously satisfies any two of the three conditions. These findings might implicate a fundamental role of those three properties in the formation and universality of the diatonic system and the seven modes.

Keywords: modes; scales; diatonic-system; derivation; twelve-tone system; algorithmic music theory

1. Introduction

The seven diatonic modes (Ionian, Aeolian, Mixolydian, Lydian, Phrygian, Dorian and Locrian) constitute one of the most influential pitch organizations in the history of Western music. Their structure and persistence have been examined from a wide range of perspectives, including acoustics, perception, historical practice, and theoretical systematization. Within modern music theory, these modes are commonly understood as specific subsets of the twelve-tone equal temperament system, whose internal organization gives rise to characteristic patterns of stability, tension, and tonal orientation.

Numerous well-established approaches have been proposed to explain the emergence and dominance of the diatonic system and the modes, most notably those grounded in acoustics and auditory perception. In particular, the work of Hermann von Helmholtz [1] has provided a foundational account by relating musical scales to the physical properties of sound and the physiology of hearing. Such approaches have proven highly influential and continue to shape contemporary understanding of tonal systems.

The present study does not seek to replace, refute, or compete with these established derivations. Instead, it adopts a complementary perspective by investigating all possible scales within the twelve-tone system and examining which of them satisfy certain empirically motivated structural properties. The resulting characterization of the seven modes is intended as an independent and self-contained result, offering an alternative viewpoint that may contribute additional insight into the structural distinctiveness of the diatonic system.

2. Methods

2.1. The Properties of the Modes

In this study, we investigate scales within the twelve-tone equal temperament system with respect to a set of three explicitly defined structural properties. To this end, we first generate the complete set of all possible distinct scales, and subsequently evaluate each of them against the selected properties. These properties are empirically motivated and are intended to capture aspects of tonal coherence, internal consistency, and perceptual stability of a scale. The underlying assumption is that such properties can, in some way, justify or explain the degree of tonal stability exhibited by a given scale. Notably, all of these properties are satisfied by the diatonic modes, which are widely regarded as possessing a high degree of tonal stability. This observation serves as an important point of reference for the present analysis. The three properties used in this study will now be introduced.

The first property of the seven modes is the possibility to arrange all scale tones of each mode in successive fifths [4]. Therefore, for each tone – except the last one in the sequence of fifths – the corresponding fifth is also contained within the mode. The fifth, as the first non-unison overtone of a fundamental, provides a certain degree of stability within the system; for this reason, the modes exhibit a strong internal coherence. This property was mainly established by the ancient greek philosopher Pythagoras (500 BCE), from which he developed a tuning system, known today as the

Pythagorean tuning [1]. Any scale that shows this property of a possible arrangement of successive fifths of its scale tones fulfills the first condition and is referred to as exhibiting a fifth-relation (FR).

The next property refers to the scale-degrees of the modes. In the diatonic system and therefore in all seven modes, each scale tone has exactly one associated third, either major or minor, from which a unique triad can be formed [3]. Although the seven modes include only major, minor, and diminished triads as diatonic harmonies, the augmented triad will also be regarded as a theoretically valid chord within a scale in this paper. These triads are referred to as the scale degrees of the mode. In each of the seven modes, it is only possible to construct exactly one triad on every scale degree using only the notes of the mode. It is therefore impossible within the modes to form both a major and a minor triad or any other two different triads on the same scale degree. For example, a scale consisting of the tones C–Eb–E–F–G–A–B permits the construction of both a C minor (C–Eb–G) and a C major (C–E–G) triad on the first scale degree, and therefore does not exhibit the same property as the seven modes.. Likewise, there is no scale tone within the modes on which no triad at all can be built using only the notes of the mode. For instance, in a scale consisting of the tones C–D–F–G–A–B, no triad can be constructed on the first scale degree, C, since neither its major nor its minor third is contained within the scale. Instances like this do not occur in the seven modes. Each degree of the mode can thus be assigned a specific chord quality (major, minor, diminished, or augmented), depending on the type of triad it forms. This unambiguous assignment allows for harmonic motion between scale degrees within the mode that does not produce ambiguities or uncertainties in voice leading or harmonic direction. This property is of particular importance for the tonal coherence of the seven modes, since ambiguity in harmonic progression may disrupt their hierarchical structure and thereby weaken the tonal center. In what follows, we shall refer to a scale in which each scale tone can be assigned exactly one triad as exhibiting unambiguous-degree-formation (UDF). (It can be shown that the property UDF is completely equivalent to the property that for each scale tone only one associated third, either major or minor, is contained within the scale.)

Another property that is considered relevant to the tonal coherence of the seven modes concerns the interval of the diminished fifth – also known as the tritone. Due to its dissonant character, the tritone possesses a strong tendency toward resolution [3]. Its resolution is achieved either by an outward semitone motion of the lower and upper tones or, conversely, by an inward semitone motion. No other interval exhibits such a high degree of tension and desire for resolution, while at the same time establishing its goal interval as a particularly satisfying point of arrival. The tritone is thus fundamentally responsible for the tonal tension structure within a scale. In the seven modes, this interval occurs exactly once and can be resolved only to a single, unique goal interval within the mode. There is no mode among the seven in which a tritone cannot be formed between two scale tones, nor one that contains more than a single such tritone. Likewise, there is no mode among the seven in which both possible resolutions of the tritone are available within the scale. As a result, the tension structure of the seven modes is uniquely determined: there are neither multiple centers of tension, in the form of several tritones, nor multiple tonal centers arising from multiple possible resolutions of a tritone within the same mode. Any scale that shows this same property concerning the tritone is referred to as exhibiting unambiguous-tritone-relation (UTR).

It should be emphasized that the selected properties are not intended to constitute an exhaustive or definitive characterization of tonal stability. Rather, they represent a limited set of criteria chosen for their practical relevance and illustrative power. The present framework is therefore not restricted to these properties and is explicitly open to extension: additional or alternative properties may be formulated and applied to the same set of scales. Exploring which scales satisfy such further criteria, and how these results relate to established or novel musical practices, may yield additional and potentially insightful perspectives on the structural organization of pitch collections. Beyond these properties, the seven modes exhibit several additional structural characteristics, notably the stepwise progression consisting of alternating whole and semitone intervals. The major (Ionian) and natural minor (Aeolian) modes, for example, exhibit a stricter form of the UTR property, which may also be examined: In these modes, the tritone does not merely resolve to two pitches that belong to the scale; rather, the resulting resolution tones are members of the tonic triad. This additional constraint is not satisfied by the remaining modes.

2.2. Mathematical Description of Pitches and Scales

In order to provide a mathematical treatment of the various pitch-classes of the twelve-tone system and to compute intervals between them, we follow the approach by Allen Forte in his work The Structure of Atonal Music [3].

With respect to octave equivalence, the twelve tones of the chromatic scale are represented as twelve pitch classes, each assigned an integer from 0 to 11: C is denoted by 0, C-sharp (which is enharmonically equivalent to D-flat) by 1, D by 2, and so on up to B, which corresponds to 11 [3, p. 2-3]. Intervals between two pitch-classes x and y are computed modulo 12 (represented as % 12 in the python code), because the twelve pitch-classes are arranged circular, so that after pitch-class 11 (or in letter-notation B) follows 0 (or in letter-notation C) and not 12 [3, p. 6]. Therefore, the interval I between note B and note C is computed by $I = |0 - 11| \mod 12 = 11 \mod 12 = 1$. The result 1 is not to be interpreted as the pitch-class of the corresponding letter-name C#, but as the number of semitones between the two pitch-classes that form the interval.

A pitch-class set (pc-set) is a set of distinct pitch-classes considered independently of order and duplication [3, p. 2]. Consequently the pitch-class sets $A = [0,2,4,5]$, $B = [2,0,4,5]$, $C = [0,0,2,4,5]$ are all equivalent because the pitch-classes of A and B are merely arranged in a different order, while C just contains a duplicate of the pitch-class 0. In the context of this paper a pc-set is understood as a scale within the twelve-tone system. Therefore the pc-sets A, B and C all represent the same scale, which can be expressed in ascending pitch-class order as 0–2–4–5 or in conventional letter-notation C–D–E–F. Furthermore, only the ordered pc-set $A = [0,2,4,5]$, which does not contain any duplicate pitch-classes and which is in ordered arrangement, would be considered here.

Some pc-sets differ from one another only by a transposition, meaning that the intervallic structure of the corresponding scale is identical but shifted by a constant number of semitones [3, p. 5-7]. In this paper, we are not concerned with the various transpositions of a given set, but rather with pc-sets that exhibit a unique interval-sequence. Therefore we only consider the equivalence classes formed by grouping together all pc-sets that can be transformed into one another through transposition, referred to as transposition-classes. Each transposition-class is then represented by a corresponding pc-set in which the first pitch class is set to 0, effectively establishing the note C as the root of the corresponding scale. This convention allows for a systematic enumeration of distinct interval-sequences, independent of absolute pitch. Therefore each of the seven modes for example will be interpreted as a distinct specific interval-sequence rooted on pitch-class 0 (or in letter-notation the root-note C) and which represents a specific transposition-class within the twelve-tone system.

To compute all these distinct transposition-classes in the programming language python, we define the function `compute_all_tc_in_12_EDO()` using the library `itertools` shown in Appendix [A.1.]. This function generates all subsets within the set of the twelve pitch-classes `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]` that include pitch-class 0. In this way, the function generates the complete set of transposition-classes, with each class represented by a single pc-set in which the pitch-class 0 appears as the root of the corresponding scale.

2.3. Algorithmic Implementation of the Three Properties

After that three further functions are defined to compute all scales that exhibit one of the three properties of the seven modes: the first function `compute_all_tc_with_FR(list_of_scales)` returns all scales from an input list that show the first property of a fifth-relation (FR) [A.2.]. The function itself calls another function `build_chain_of_fifths(first_note, residual_notes)` that tries to arrange the pitch-classes of an input list `residual_notes` in successive fifths and returns this arrangement if it is possible.

The next function is defined as `compute_all_tc_with_UDF(list_of_scales)` which gets an input list of scales (or pc-sets) and returns every scale, that exhibits the second property of the unambiguous-degree-formation (UDF) (see Appendix [A.3.]). In this function an algorithm is implemented, that tries to build all four triads on each tone for every scale out of the input list only with tones from the respective scale. It also counts the number of possible triad formations on each tone and only returns those scales, in which there is exactly one possibility for each tone.

The last function is defined as `compute_all_tc_with_UTR(list_of_scales)` and returns every scale out of an input list, that shows the third property of the unambiguous-tritone-relation (UTR) (see Appendix [A.4.]). The algorithm used in this function first of all filters those scales which contain more or less than one tritone interval by computing all possible intervals between the notes of the scale. The property UTR also demands an unambiguous resolution of the tritone within the scale. Therefore the algorithm continues by counting all possible resolutions of the single tritone in the remaining scales and returns those, with one possible resolution only.

3. Results

3.1. Computation of all Transposition–Classes that exhibit one of the three Properties

By executing the function `compute_all_transposition_classes()` 2048 transposition-classes are generated, where each transposition-class is represented by a corresponding pc-set starting on 0. In other words there exists a total of 2048 distinct interval-sequences (or scales) of a minimum of one and a maximum of twelve non-duplicate pitch-classes, that are rooted on 0.

These 2048 distinct transposition-classes are inserted in the list `all_transposition_classes` by the function. An excerpt of the list is shown in Appendix [B.1.]. This list can now be used as the input for the three functions `compute_all_scales_with_FR(list_of_scales)`, `compute_all_scales_with_UDF(list_of_scales`, `compute_all_scales_with_UTR(list_of_scales)` to compute all transposition-classes (or distinct scales) with one of the three properties each.

An excerpt of the list which contains all transposition-classes that exhibit the property FR is shown in Appendix [B.2.]. This list contains 67 transposition-classes in total. The seven modes can be found at position 22 to 28.

The list which contains all transposition-classes that exhibit the property UDF is shown in Appendix [B.3.]. It contains 33 transposition-classes in total. The seven modes can be found at position 27 to 33.

An excerpt of the list which contains all transposition-classes that exhibit the property UTR is shown in Appendix [B.4.]. This list contains 162 transposition-classes in total. The seven modes can be found at positions 128, 129, 144, 145, 149, 150 and 151.

3.2. Computation of all Transposition-Classes that exhibit any two of the three Properties

In each case the seven modes are not the only transposition-classes that exhibit the respective property. It is therefore necessary, to conduct a more detailed analysis of each of the three lists of transposition-classes, that exhibit at least one of the three properties. The aim is to identify all transposition-classes that simultaneously exhibit at least two of the three properties. This can be achieved by using each of the lists generated by one function as the input for a different function. In doing so, all transposition classes can be determined that exhibit the properties of:

1. FR and UDF
2. FR and UTR
3. UDF and UTR

The list generated for the first case is shown in Appendix [C.1.], for the second case in Appendix [C.2.] and for the third case in Appendix [C.3.].

4. Discussion

The results shown in Appendix [B] proof that other transposition-classes than the seven modes exhibit at least one of the three properties as well. In the case of the properties FR and UTR this result can be interpreted as trivial, since other transposition-classes consisting of a different number of successive fifths than seven or other transposition-classes with only one tritone and a single resolution for it can easily be constructed manually. Although all 67 transposition-classes that exhibit the property FR can systematically be computed by considering different successions of alternating numbers of fifths that contain the root pitch 0, the 162 transposition-classes that exhibit the property UTR are not so easily computed without an algorithm.

The most striking of the three computed lists is probably the one, which contains all transposition-classes which exhibit the property UDF. Only 33 transposition-classes in total exhibit this property. It follows that next to the seven modes there exist only 26 other transposition-classes in which only exactly one triad (major, minor, diminished or augmented) can be assigned to each scale-tone. This is an interesting result on its own which motivates further analysis of these transposition-classes and the importance of the property UDF of scales within the Western tonal framework.

Because the seven modes are not the only transposition classes that exhibit at least one of the three properties, this result alone is not sufficient to justify their dominance within the Western musical system. However, when the additional results presented in Appendix C are taken into account, one may argue that the prominence of the seven modes can be explained by the fact that they appear to be the only transposition classes that simultaneously exhibit any two of the three properties. Moreover, the seven modes are, consequently, the only transposition classes that satisfy all three properties at once. Since the presence of any two properties already suffices to derive the seven modes, it follows that, in principle, one of the three properties is dispensable.

Any historical inference concerning the initial establishment of the seven modes would require an examination of the extent to which the properties FR, UDF, and UTR were already implicitly operative within ancient musical systems. Such an investigation would necessarily involve substantially deeper historical and musicological analyses. Conducting these analyses in full scope, however, lies beyond the intentions of the present study, which aims primarily at a structural and systematic exploration rather than a comprehensive historical account.

References

[1] Hermann von Helmholtz, *Die Lehre von den Tonempfindungen als Physiologische Grundlage für die Theorie der Musik,* Braunschweig: Friedrich Vieweg und Sohn, 1877.

[2] Allen Forte, *The Structures of Atonal Music,* New Haven and London: Yale University Press, 1973.

[3] Edward Aldwell, Carl Schachter, *Harmony and Voice Leading,* Boston: Schirmer Cengage Learning, 2011.

[4] John Vincent, *The Diatonic Modes in Modern Music,* Berkeley and Los Angeles: University of California Press, 1951.

Appendices

Appendix A.1.

```
def compute_all_tc_in_12_EDO():

    all_pc_sets_in_12_EDO = []

    for k in range(1, len(pitch_classes) + 1):
        combinations = itertools.combinations(pitch_classes, k)
        all_pc_sets_in_12_EDO.extend([list(combination) for combination in combinations])


    all_transposition_classes_in_12_EDO = []

    for pc_set in all_pc_sets_in_12_EDO:

        if 0 in pc_set:
            all_transposition_classes_in_12_EDO.append(pc_set)


    return all_transposition_classes_in_12_EDO


all_tc_in_12_EDO = compute_all_tc_in_12_EDO()
```

Appendix A.2.

```
def compute_all_tc_with_UTR(list_of_scales):

    all_scales_with_UTR = []

    for scale in list_of_scales:
        number_tritones = 0
        notes_of_tritone = []

        for i in range(len(scale)):
            for j in range(i + 1, len(scale)):
                if abs(scale[i] - scale[j]) % 12 == 6:
                    number_tritones += 1
                    notes_of_tritone = [scale[i], scale[j]]

        if number_tritones == 1:
            a, b = notes_of_tritone
            number_of_resolutions = 0

            if ((a - 1) % 12 in scale) and ((b + 1) % 12 in scale):
                number_of_resolutions += 1
            if ((a + 1) % 12 in scale) and ((b - 1) % 12 in scale):
                        number_of_resolutions += 1

            if number_of_resolutions == 1:
                all_scales_with_UTR.append(scale)

    return all_scales_with_UTR
```

Appendix A.3.

```python
def build_chain_of_fifths(first_note, remaining_notes):

    chain_of_fifths = [first_note]
    current_note = first_note

    while True:

        next_note = (current_note + 7) % 12

        if next_note in remaining_notes:
            chain_of_fifths.append(next_note)
            remaining_notes.remove(next_note)
            current_note = next_note
        else:
            break

    return chain_of_fifths



def compute_all_tc_with_FR(list_of_scales):

    all_scales_with_FR = []

    for scale in list_of_scales:

        for first_note in scale:
            residual_notes = scale.copy()
            residual_notes.remove(first_note)
            chain_of_fifths = build_chain_of_fifths(first_note, residual_notes)

            if sorted(chain_of_fifths) == sorted(scale):
                all_scales_with_FR.append(sorted(chain_of_fifths))
                break

    return all_scales_with_FR
```

Appendix A.4.

```python
def compute_all_tc_with_UDF(list_of_scales):

    all_scales_with_UDF = []

    for scale in list_of_scales:

        one_triad_per_note = True

        for note1 in scale:

            number_of_triads_on_note = 0

            for note2 in scale:
                for note3 in scale:
                    if note2 == note1 or note3 == note1 or note2 == note3:
                        continue

                    interval1 = (note2 - note1) % 12
                    interval2 = (note3 - note2) % 12

                    if interval1 in [3, 4] and interval2 in [3, 4]:
                                    number_of_triads_on_note += 1

            if number_of_triads_on_note != 1:
                one_triad_per_note = False
                break

        if one_triad_per_note:
            all_scales_with_UDF.append(scale)


     return all_scales_with_UDF
```

Appendix B.1. (All transposition-classes)

```
Scale 1: [0]
Scale 2: [0, 1]
Scale 3: [0, 2]
Scale 4: [0, 3]
Scale 5: [0, 4]
Scale 6: [0, 5]
Scale 7: [0, 6]
Scale 8: [0, 7]
Scale 9: [0, 8]
Scale 10: [0, 9]
Scale 11: [0, 10]
Scale 12: [0, 11]
Scale 13: [0, 1, 2]
Scale 14: [0, 1, 3]
Scale 802: [0, 2, 3, 5, 7, 11]
Scale 803: [0, 2, 3, 5, 8, 9]
Scale 804: [0, 2, 3, 5, 8, 10]
Scale 805: [0, 2, 3, 5, 8, 11]
Scale 806: [0, 2, 3, 5, 9, 10]
Scale 807: [0, 2, 3, 5, 9, 11]
Scale 808: [0, 2, 3, 5, 10, 11]
Scale 809: [0, 2, 3, 6, 7, 8]
Scale 810: [0, 2, 3, 6, 7, 9]
Scale 811: [0, 2, 3, 6, 7, 10]
Scale 812: [0, 2, 3, 6, 7, 11]
Scale 813: [0, 2, 3, 6, 8, 9]
Scale 814: [0, 2, 3, 6, 8, 10]
Scale 815: [0, 2, 3, 6, 8, 11]
Scale 2035: [0, 2, 4, 5, 6, 7, 8, 9, 10, 11]
Scale 2036: [0, 3, 4, 5, 6, 7, 8, 9, 10, 11]
Scale 2037: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Scale 2038: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11]
Scale 2039: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11]
Scale 2040: [0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11]
Scale 2041: [0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11]
Scale 2042: [0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 11]
Scale 2043: [0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11]
Scale 2044: [0, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11]
Scale 2045: [0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11]
Scale 2046: [0, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11]
Scale 2047: [0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
Scale 2048: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

Appendix B.2. (All transposition-classes with FR)

```
Scale 1: [0]
Scale 2: [0, 5]
Scale 3: [0, 7]
Scale 4: [0, 2, 7]
Scale 5: [0, 5, 7]
Scale 6: [0, 5, 10]
Scale 7: [0, 2, 5, 7]
Scale 8: [0, 2, 7, 9]
Scale 9: [0, 3, 5, 10]
Scale 10: [0, 5, 7, 10]
Scale 11: [0, 2, 4, 7, 9]
Scale 12: [0, 2, 5, 7, 9]
Scale 13: [0, 2, 5, 7, 10]
Scale 14: [0, 3, 5, 7, 10]
Scale 15: [0, 3, 5, 8, 10]
Scale 16: [0, 1, 3, 5, 8, 10]
Scale 17: [0, 2, 3, 5, 7, 10]
Scale 18: [0, 2, 4, 5, 7, 9]
Scale 19: [0, 2, 4, 7, 9, 11]
Scale 20: [0, 2, 5, 7, 9, 10]
Scale 21: [0, 3, 5, 7, 8, 10]
Scale 22: [0, 1, 3, 5, 6, 8, 10]
Scale 23: [0, 1, 3, 5, 7, 8, 10]
Scale 24: [0, 2, 3, 5, 7, 8, 10]
Scale 25: [0, 2, 3, 5, 7, 9, 10]
Scale 26: [0, 2, 4, 5, 7, 9, 10]
Scale 27: [0, 2, 4, 5, 7, 9, 11]
Scale 28: [0, 2, 4, 6, 7, 9, 11]
Scale 29: [0, 1, 2, 3, 5, 7, 8, 10]
Scale 30: [0, 1, 2, 4, 6, 7, 9, 11]
Scale 31: [0, 1, 3, 5, 6, 7, 8, 10]
Scale 32: [0, 1, 3, 5, 6, 8, 10, 11]
Scale 33: [0, 2, 3, 4, 5, 7, 9, 10]
Scale 34: [0, 2, 3, 5, 7, 8, 9, 10]
Scale 35: [0, 2, 4, 5, 6, 7, 9, 11]
Scale 36: [0, 2, 4, 5, 7, 9, 10, 11]
Scale 37: [0, 1, 2, 3, 5, 6, 7, 8, 10]
Scale 38: [0, 1, 2, 3, 5, 7, 8, 9, 10]
Scale 39: [0, 1, 2, 4, 5, 6, 7, 9, 11]
Scale 40: [0, 1, 2, 4, 6, 7, 8, 9, 11]
Scale 41: [0, 1, 3, 4, 5, 6, 8, 10, 11]
Scale 42: [0, 1, 3, 5, 6, 7, 8, 10, 11]
Scale 43: [0, 2, 3, 4, 5, 7, 8, 9, 10]
Scale 44: [0, 2, 3, 4, 5, 7, 9, 10, 11]
Scale 45: [0, 2, 4, 5, 6, 7, 9, 10, 11]
Scale 46: [0, 1, 2, 3, 4, 5, 7, 8, 9, 10]
Scale 47: [0, 1, 2, 3, 4, 6, 7, 8, 9, 11]
Scale 48: [0, 1, 2, 3, 5, 6, 7, 8, 9, 10]
Scale 49: [0, 1, 2, 3, 5, 6, 7, 8, 10, 11]
Scale 50: [0, 1, 2, 4, 5, 6, 7, 8, 9, 11]
Scale 51: [0, 1, 2, 4, 5, 6, 7, 9, 10, 11]
Scale 52: [0, 1, 3, 4, 5, 6, 7, 8, 10, 11]
Scale 53: [0, 1, 3, 4, 5, 6, 8, 9, 10, 11]
Scale 54: [0, 2, 3, 4, 5, 6, 7, 9, 10, 11]
Scale 55: [0, 2, 3, 4, 5, 7, 8, 9, 10, 11]
Scale 56: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Scale 57: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11]
Scale 58: [0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11]
Scale 59: [0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11]
Scale 60: [0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11]
Scale 61: [0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 11]
Scale 62: [0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11]
Scale 63: [0, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11]
Scale 64: [0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11]
Scale 65: [0, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11]
Scale 66: [0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
Scale 67: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

Appendix B.3. (All transposition-classes with UDF)

```
Scale 1: [0, 4, 8]
Scale 2: [0, 1, 4, 8]
Scale 3: [0, 3, 6, 9]
Scale 4: [0, 3, 7, 11]
Scale 5: [0, 4, 5, 8]
Scale 6: [0, 4, 8, 9]
Scale 7: [0, 1, 4, 7, 10]
Scale 8: [0, 1, 4, 8, 10]
Scale 9: [0, 2, 3, 6, 9]
Scale 10: [0, 2, 3, 6, 10]
Scale 11: [0, 2, 4, 5, 8]
Scale 12: [0, 3, 5, 6, 9]
Scale 13: [0, 3, 6, 8, 9]
Scale 14: [0, 3, 6, 9, 11]
Scale 15: [0, 3, 7, 9, 11]
Scale 16: [0, 4, 6, 8, 9]
Scale 17: [0, 1, 4, 6, 7, 10]
Scale 18: [0, 1, 4, 6, 8, 10]
Scale 19: [0, 2, 3, 6, 8, 9]
Scale 20: [0, 2, 3, 6, 8, 10]
Scale 21: [0, 2, 4, 5, 8, 10]
Scale 22: [0, 2, 4, 6, 7, 10]
Scale 23: [0, 2, 4, 6, 8, 9]
Scale 24: [0, 2, 4, 6, 8, 10]
Scale 25: [0, 3, 5, 6, 9, 11]
Scale 26: [0, 3, 5, 7, 9, 11]
Scale 27: [0, 1, 3, 5, 6, 8, 10]
Scale 28: [0, 1, 3, 5, 7, 8, 10]
Scale 29: [0, 2, 3, 5, 7, 8, 10]
Scale 30: [0, 2, 3, 5, 7, 9, 10]
Scale 31: [0, 2, 4, 5, 7, 9, 10]
Scale 32: [0, 2, 4, 5, 7, 9, 11]
Scale 33: [0, 2, 4, 6, 7, 9, 11]
```

Appendix B.4. (All transposition-classes with UTR)

```
Scale 1: [0, 1, 5, 6]
Scale 2: [0, 1, 7, 8]
Scale 3: [0, 4, 5, 11]
Scale 4: [0, 6, 7, 11]
Scale 5: [0, 1, 2, 5, 6]
Scale 6: [0, 1, 2, 8, 9]
Scale 7: [0, 1, 3, 5, 6]
Scale 8: [0, 1, 3, 7, 8]
Scale 9: [0, 1, 4, 5, 6]
Scale 10: [0, 1, 4, 5, 11]
Scale 11: [0, 1, 4, 7, 8]
Scale 12: [0, 1, 5, 6, 8]
Scale 13: [0, 1, 5, 6, 9]
Scale 14: [0, 1, 5, 6, 10]
Scale 15: [0, 1, 5, 7, 8]
Scale 16: [0, 1, 7, 8, 9]
Scale 17: [0, 1, 7, 8, 10]
Scale 18: [0, 1, 7, 8, 11]
Scale 19: [0, 2, 3, 7, 8]
Scale 20: [0, 2, 3, 9, 10]
Scale 21: [0, 2, 4, 5, 11]
Scale 22: [0, 2, 6, 7, 11]
Scale 23: [0, 3, 4, 5, 11]
Scale 24: [0, 3, 4, 8, 9]
Scale 25: [0, 3, 4, 10, 11]
Scale 26: [0, 3, 6, 7, 11]
Scale 27: [0, 4, 5, 7, 11]
Scale 28: [0, 4, 5, 8, 11]
Scale 29: [0, 4, 5, 9, 10]
Scale 30: [0, 4, 5, 9, 11]
Scale 31: [0, 4, 6, 7, 11]
Scale 32: [0, 6, 7, 8, 11]
Scale 33: [0, 6, 7, 9, 11]
Scale 34: [0, 6, 7, 10, 11]
Scale 35: [0, 1, 2, 3, 5, 6]
Scale 36: [0, 1, 2, 3, 9, 10]
Scale 37: [0, 1, 2, 4, 5, 6]
Scale 38: [0, 1, 2, 4, 5, 11]
Scale 39: [0, 1, 2, 4, 8, 9]
Scale 40: [0, 1, 2, 5, 6, 9]
Scale 41: [0, 1, 2, 5, 6, 10]
Scale 42: [0, 1, 2, 5, 8, 9]
Scale 43: [0, 1, 2, 8, 9, 10]
Scale 44: [0, 1, 2, 8, 9, 11]
Scale 45: [0, 1, 3, 4, 5, 6]
Scale 46: [0, 1, 3, 4, 5, 11]
Scale 47: [0, 1, 3, 4, 7, 8]
Scale 48: [0, 1, 3, 4, 8, 9]
Scale 49: [0, 1, 3, 4, 10, 11]
Scale 50: [0, 1, 3, 5, 6, 8]
Scale 51: [0, 1, 3, 5, 6, 10]
Scale 52: [0, 1, 3, 5, 7, 8]
Scale 53: [0, 1, 3, 7, 8, 10]
Scale 54: [0, 1, 3, 7, 8, 11]
Scale 55: [0, 1, 4, 5, 6, 8]
Scale 56: [0, 1, 4, 5, 6, 9]
Scale 57: [0, 1, 4, 5, 7, 8]
Scale 58: [0, 1, 4, 5, 8, 11]
Scale 59: [0, 1, 4, 5, 9, 10]
Scale 60: [0, 1, 4, 5, 9, 11]
Scale 61: [0, 1, 4, 7, 8, 9]
Scale 62: [0, 1, 4, 7, 8, 11]
Scale 63: [0, 1, 5, 6, 8, 9]
Scale 64: [0, 1, 5, 6, 8, 10]
Scale 65: [0, 1, 5, 6, 9, 10]
Scale 66: [0, 1, 5, 7, 8, 9]
Scale 67: [0, 1, 5, 7, 8, 10]
Scale 68: [0, 1, 7, 8, 9, 10]
Scale 69: [0, 1, 7, 8, 9, 11]
Scale 70: [0, 1, 7, 8, 10, 11]
Scale 71: [0, 2, 3, 4, 5, 11]
Scale 72: [0, 2, 3, 4, 7, 8]
Scale 73: [0, 2, 3, 4, 10, 11]
Scale 74: [0, 2, 3, 5, 7, 8]
Scale 75: [0, 2, 3, 5, 9, 10]
Scale 76: [0, 2, 3, 6, 7, 11]
Scale 77: [0, 2, 3, 7, 8, 10]
Scale 78: [0, 2, 3, 7, 8, 11]
Scale 79: [0, 2, 3, 7, 9, 10]
Scale 80: [0, 2, 3, 9, 10, 11]
Scale 81: [0, 2, 4, 5, 7, 11]
Scale 82: [0, 2, 4, 5, 9, 10]
Scale 83: [0, 2, 4, 5, 9, 11]
Scale 84: [0, 2, 4, 6, 7, 11]
Scale 85: [0, 2, 6, 7, 9, 11]
Scale 86: [0, 2, 6, 7, 10, 11]
Scale 87: [0, 3, 4, 5, 7, 11]
Scale 88: [0, 3, 4, 5, 8, 9]
Scale 89: [0, 3, 4, 5, 8, 11]
Scale 90: [0, 3, 4, 6, 7, 11]
Scale 91: [0, 3, 4, 7, 8, 9]
Scale 92: [0, 3, 4, 7, 10, 11]
Scale 93: [0, 3, 4, 8, 9, 11]
Scale 94: [0, 3, 4, 8, 10, 11]
Scale 95: [0, 3, 6, 7, 8, 11]
Scale 96: [0, 3, 6, 7, 10, 11]
Scale 97: [0, 4, 5, 7, 8, 11]
Scale 98: [0, 4, 5, 7, 9, 10]
Scale 99: [0, 4, 5, 7, 9, 11]
Scale 100: [0, 4, 5, 8, 9, 10]
Scale 101: [0, 4, 5, 8, 9, 11]
Scale 102: [0, 4, 6, 7, 8, 11]
Scale 103: [0, 4, 6, 7, 9, 11]
Scale 104: [0, 6, 7, 8, 9, 11]
Scale 105: [0, 6, 7, 8, 10, 11]
Scale 106: [0, 6, 7, 9, 10, 11]
Scale 107: [0, 1, 2, 3, 4, 5, 6]
Scale 108: [0, 1, 2, 3, 4, 5, 11]
Scale 109: [0, 1, 2, 3, 4, 10, 11]
Scale 110: [0, 1, 2, 3, 5, 6, 10]
Scale 111: [0, 1, 2, 3, 5, 9, 10]
Scale 112: [0, 1, 2, 3, 9, 10, 11]
Scale 113: [0, 1, 2, 4, 5, 6, 9]
Scale 114: [0, 1, 2, 4, 5, 8, 9]
Scale 115: [0, 1, 2, 4, 5, 9, 10]
Scale 116: [0, 1, 2, 4, 5, 9, 11]
Scale 117: [0, 1, 2, 4, 8, 9, 11]
Scale 118: [0, 1, 2, 5, 6, 9, 10]
Scale 119: [0, 1, 2, 5, 8, 9, 10]
Scale 120: [0, 1, 2, 8, 9, 10, 11]
Scale 121: [0, 1, 3, 4, 5, 6, 8]
Scale 122: [0, 1, 3, 4, 5, 7, 8]
Scale 123: [0, 1, 3, 4, 5, 8, 9]
Scale 124: [0, 1, 3, 4, 5, 8, 11]
Scale 125: [0, 1, 3, 4, 7, 8, 11]
Scale 126: [0, 1, 3, 4, 8, 9, 11]
Scale 127: [0, 1, 3, 4, 8, 10, 11]
Scale 128: [0, 1, 3, 5, 6, 8, 10]
Scale 129: [0, 1, 3, 5, 7, 8, 10]
Scale 130: [0, 1, 3, 7, 8, 10, 11]
Scale 131: [0, 1, 4, 5, 6, 8, 9]
Scale 132: [0, 1, 4, 5, 7, 8, 9]
Scale 133: [0, 1, 4, 5, 8, 9, 10]
Scale 134: [0, 1, 4, 5, 8, 9, 11]
Scale 135: [0, 1, 4, 7, 8, 9, 11]
Scale 136: [0, 1, 5, 6, 8, 9, 10]
Scale 137: [0, 1, 5, 7, 8, 9, 10]
Scale 138: [0, 1, 7, 8, 9, 10, 11]
Scale 139: [0, 2, 3, 4, 5, 7, 8]
Scale 140: [0, 2, 3, 4, 5, 7, 11]
Scale 141: [0, 2, 3, 4, 6, 7, 11]
Scale 142: [0, 2, 3, 4, 7, 8, 11]
Scale 143: [0, 2, 3, 4, 7, 10, 11]
Scale 144: [0, 2, 3, 5, 7, 8, 10]
Scale 145: [0, 2, 3, 5, 7, 9, 10]
Scale 146: [0, 2, 3, 6, 7, 10, 11]
Scale 147: [0, 2, 3, 7, 8, 10, 11]
Scale 148: [0, 2, 3, 7, 9, 10, 11]
Scale 149: [0, 2, 4, 5, 7, 9, 10]
Scale 150: [0, 2, 4, 5, 7, 9, 11]
Scale 151: [0, 2, 4, 6, 7, 9, 11]
Scale 152: [0, 2, 6, 7, 9, 10, 11]
Scale 153: [0, 3, 4, 5, 7, 8, 9]
Scale 154: [0, 3, 4, 5, 7, 8, 11]
Scale 155: [0, 3, 4, 6, 7, 8, 11]
Scale 156: [0, 3, 4, 7, 8, 9, 11]
Scale 157: [0, 3, 4, 7, 8, 10, 11]
Scale 158: [0, 3, 6, 7, 8, 10, 11]
Scale 159: [0, 4, 5, 7, 8, 9, 10]
Scale 160: [0, 4, 5, 7, 8, 9, 11]
Scale 161: [0, 4, 6, 7, 8, 9, 11]
Scale 162: [0, 6, 7, 8, 9, 10, 11]
```

Appendix C.1. (All transposition-classes with FR and UDF

```
Scale 1: [0, 1, 3, 5, 6, 8, 10]
Scale 2: [0, 1, 3, 5, 7, 8, 10]
Scale 3: [0, 2, 3, 5, 7, 8, 10]
Scale 4: [0, 2, 3, 5, 7, 9, 10]
Scale 5: [0, 2, 4, 5, 7, 9, 10]
Scale 6: [0, 2, 4, 5, 7, 9, 11]
Scale 7: [0, 2, 4, 6, 7, 9, 11]
```

Appendix C.2. (All transposition-classes with FR and UTR)

```
Scale 1: [0, 1, 3, 5, 6, 8, 10]
Scale 2: [0, 1, 3, 5, 7, 8, 10]
Scale 3: [0, 2, 3, 5, 7, 8, 10]
Scale 4: [0, 2, 3, 5, 7, 9, 10]
Scale 5: [0, 2, 4, 5, 7, 9, 10]
Scale 6: [0, 2, 4, 5, 7, 9, 11]
Scale 7: [0, 2, 4, 6, 7, 9, 11]
```

Appendix C.3. (All transposition-classes with UTR and UDF)

```
Scale 1: [0, 1, 3, 5, 6, 8, 10]
Scale 2: [0, 1, 3, 5, 7, 8, 10]
Scale 3: [0, 2, 3, 5, 7, 8, 10]
Scale 4: [0, 2, 3, 5, 7, 9, 10]
Scale 5: [0, 2, 4, 5, 7, 9, 10]
Scale 6: [0, 2, 4, 5, 7, 9, 11]
Scale 7: [0, 2, 4, 6, 7, 9, 11]
```