A Mathematically Verified Approach to Three-Note Chord Identification

Methodology Behind the Intervallo Triad Calculator

Manuel Hernandez

May 2026

Abstract

This paper documents the methodology used to construct the chord identification engine in *Intervallo*, a music theory calculator for iOS. We approach three-note chord identification as a finite enumeration problem rather than a heuristic pattern-matching problem. Within the constraint of three distinct pitch classes contained in one octave, the universe of possible interval structures is exactly fifty-five pairs. We catalog all fifty-five, identify which correspond to named chord structures in Western tonal and post-tonal harmony, and document the Swift implementation derived from this catalog. We discuss the role of multiple AI models in cross-validating the catalog, the value of computational verification in resolving model disagreements, and the philosophical position that informs which structures the calculator names and which it correctly identifies as unnamed.

1 Introduction

Three notes within one octave produce a finite set of possible interval relationships. That set can be enumerated completely, mapped to its musical interpretations through reference to established music theory, and implemented as a single comprehensive lookup table. The result is a calculator with a defensible answer for every possible input and no hidden gaps in coverage.

2 The Combinatorial Space

Given three distinct pitch classes within one octave, normalized so that the lowest note is the reference, we can express any three-note structure as a pair of intervals measured in semitones:

$$(s_1, s_2) \text{ where } 1 \leq s_1 < s_2 \leq 11 \tag{1}$$

where:

- s_1 = semitones from lowest to middle note
- s_2 = semitones from lowest to highest note

The constraint that $s_1 \geq 1$ and $s_2 \leq 11$ ensures three distinct pitch classes within one octave. The constraint that $s_1 < s_2$ ensures we count each structure once regardless of voicing order.

Counting these pairs gives us:

$$\sum_{i=1}^{10} i = 55 \text{ distinct interval pairs} \tag{2}$$

This is the complete universe of three-note pitch-class structures within an octave. Every chord identification problem reduces to a question about one of these 55 pairs.

3 Methodology

We sought to produce an exhaustive reference identifying which of the 55 pairs correspond to named chord structures in Western harmony, which have multiple valid interpretations, and which represent unnamed clusters or fragments. Our methodology employed three independent processes.

3.1 Multi-Model Cross-Reference

We submitted the same prompt to four large language models: Claude Opus, Claude Opus with extended thinking, Gemini Pro, and ChatGPT in deliberation mode. Each was asked to produce a complete catalog of every named three-note chord structure expressible as a normalized semitone pair, with chord names, abbreviations, alternative interpretations, and musical context.

The four outputs were then cross-referenced for agreement and disagreement. Pairs where all four models agreed on the chord identity were treated as ground truth. Pairs where models disagreed were flagged for investigation. Pairs where only one model proposed a name were treated with skepticism.

3.2 Computational Verification

For each chord structure proposed by the models, we wrote Python code to compute the actual semitone pair produced by the chord in each of its inversions. This served as the ultimate arbiter when models disagreed about which interval pair corresponded to which inversion.

This verification step proved critical. One model's quick-reference table erroneously listed the major triad first inversion as the pair $(4, 9)$, when computational verification confirmed that $(4, 9)$ is actually the minor triad first inversion. The major triad first inversion is $(3, 8)$. Without computational verification, this error could have entered the implementation.

3.3 Theoretical Categorization

Following verification, we organized the named structures into categories aligned with their function in Western harmony:

- Standard triads in root position and both inversions (10 structures)
- Suspended chords and their voicings (4 structures)
- Altered triads (1 structure)
- Shell voicings derived from seventh chords (7 structures)
- Power chord voicings with octave (2 structures)

This produces 24 named pair-quality assignments mapping to a smaller number of distinct chord qualities. The remaining pairs were classified as either ambiguous, post-tonal cluster sonorities, or genuinely unnamed.

4 Findings

4.1 Multi-Model Reliability

The four models showed strong agreement on standard triadic structures. All four correctly identified the four primary triad qualities and their three inversion variants. All four agreed on the suspended chord pair assignments, on the augmented triad's symmetrical property, and on the shell voicings derived from common seventh chords.

Disagreements clustered in three areas: less common altered triads, quartal voicings, and the appropriate names for non-functional cluster sonorities. These disagreements reflect genuine ambiguity in music theory rather than model error.

4.2 Counter-Intuitive Result on Compute Allocation

We observed that allocating more computational resources to this problem produced worse results, not better. Extended thinking modes timed out or returned to their original prompts without producing complete tables. The models that performed best were those that treated the task as a knowledge retrieval and organization problem rather than a deep reasoning problem.

This is consistent with the structure of the problem. The complete catalog of three-note chord structures already exists in music theory references. Constructing the catalog is a cataloging task, not a reasoning task. The deep reasoning machinery that helps with multi-step problems becomes a liability when applied to enumeration.

4.3 Inversion Detection

A key implementation insight emerged from the verification work. The conventional approach to inversion detection involves rotating the input notes through three positions and testing each rotation against a small set of root-position chord patterns. This works for the four primary triads but generates ambiguity when the same interval pair can result from multiple chord identities.

A cleaner approach is to recognize that each interval pair uniquely encodes both the chord quality and its inversion. The pair $(3, 8)$ is always the first inversion of a major triad. The pair $(5, 9)$ is always the second inversion of a major triad. By embedding inversion detection directly in the pattern matcher rather than relying on rotation, we eliminate one source of error and produce more transparent code.

4.4 The Augmented Triad Symmetry

The augmented triad presents a special case worth noting. Because it consists of two stacked major thirds, all three rotations of an augmented triad produce the same interval pair $(4, 8)$. The chord is symmetrical: C-E-G#, E-G#-C, and G#-C-E all produce identical pitch-class content with no distinguishing intervals.

This means the question of which note is the root cannot be answered from pitch content alone. It must be answered from spelling, key context, or surrounding harmony.

5 Spelling-Awareness Versus Sound-Awareness

A subtle but important architectural decision concerns whether the calculator should be spelling-aware or sound-aware. These approaches diverge on enharmonically equivalent notes.

Consider the chord C-E-G♭. Computed by sound, this is identical to C-E-F#. Computed by spelling, the two are different chords with different theoretical identities. C-E-G♭ is most naturally read as a major triad with a flattened fifth, while C-E-F# would be read in a Lydian context.

The Intervallo interval calculator is spelling-aware: it treats F→B as an Augmented 4th and F→C♭ as a Diminished 5th, despite their identical sound. This reflects the theoretical reality that spelling carries musical meaning beyond sound.

The Intervallo triad calculator, by contrast, is sound-aware: it normalizes notes to their pitch class before identification. C-E#-G correctly identifies as a C suspended fourth chord because C-E#-G sounds identical to C-F-G. This decision was made because triadic identification depends on the bass-relative interval relationships, which are properties of sound rather than spelling.

These two design decisions are deliberately different. Each calculator reflects the theoretical conventions appropriate to its task.

6 The Question of Coverage

The 55 possible interval pairs partition into three groups under our analysis.

The first group consists of pairs that map to well-named chord structures with broad agreement across music theory sources. These number 24 pairs. Intervallo names each of these with its standard chord symbol: the four standard triads across all inversions (12 pairs), suspended chords and quartal structures (4 pairs), shell voicings derived from seventh chords (7 pairs), and power chord voicings (1 pair).

The second group consists of pairs that have valid musical interpretations but where those interpretations are context-dependent. The pair $(5, 10)$ might be a quartal voicing in one context and a suspended chord inversion in another. Intervallo names these with their most common interpretation but the result display can show alternatives.

The third group consists of pairs that have no standard chord name in Western harmony. These are typically clusters, dissonant fragments of larger structures, or post-tonal sonorities used for color rather than functional harmony. Intervallo correctly identifies these as unnamed rather than forcing a name.

This third group is essential to the calculator's integrity. A chord identifier that names every possible input is a chord identifier that lies. The presence of "Not a triad" as an honest answer for genuinely non-triadic structures distinguishes Intervallo from tools that prioritize completeness of output over correctness.

7 Implementation

The complete chord identification logic compiles to a single Swift function of approximately 30 lines. Every named pair is one switch case. Every unnamed pair falls through to the default. The total mapping is small enough to read in a single screen and exhaustive enough to handle every possible three-note input.

```
private func triadQuality(semisToThird: Int, semisToFifth: Int) ->
    TriadQuality? {
     switch (semisToThird, semisToFifth) {

     // Major triad    all inversions
     case (4, 7):  return .major
     case (3, 8):  return .major
     case (5, 9):  return .major

     // Minor triad    all inversions
     case (3, 7):  return .minor
     case (4, 9):  return .minor
     case (5, 8):  return .minor

     // Diminished triad    all inversions
     case (3, 6):  return .diminished
     case (3, 9):  return .diminished
     case (6, 9):  return .diminished
```

```
    // Augmented triad (symmetrical)
    case (4, 8):  return .augmented

    // Suspended and quartal
    case (2, 7):  return .sus2
    case (5, 7):  return .sus4
    case (2, 9):  return .sus4
    case (5, 10): return .quartal

    // Altered
    case (4, 6):  return .majorFlat5

    // Shell voicings
    case (4, 10): return .dom7Shell
    case (3, 10): return .min7Shell
    case (4, 11): return .maj7Shell
    case (3, 11): return .minMaj7Shell
    case (6, 10): return .halfDimShell
    case (7, 10): return .dom7NoThird
    case (7, 11): return .maj7NoThird

    // Power chord
    case (7, 12): return .powerChord
    case (5, 12): return .powerChord

    default: return nil
    }
}
```

Inversion identification is handled by a parallel switch on the same interval pairs, returning the appropriate inversion classification. Root note identification for inverted chords is handled by indexing into the input note array based on the detected inversion.

This entire system is deterministic, exhaustive within its declared scope, and returns honest unnamed status for inputs outside that scope.

8 Limitations

The current implementation handles three-note structures within one octave. It does not address four-note structures such as full seventh chords, extended chords with ninths and elevenths, or chords spanning more than one octave with internal voicings.

Extension to four-note chord identification follows the same methodology with a larger combinatorial space. The number of unique four-note structures within one octave is significantly larger than 55, but the same approach of exhaustive enumeration, multi-model cross-reference, and computational verification scales naturally.

The current implementation also does not perform key-context analysis. A chord identified as “C Major” might function as a tonic in C major, a dominant in F major, or a chromatic mediant in E major depending on surrounding harmony. Intervallo identifies the chord but does not classify its function.

9 Conclusion

Three-note chord identification is fundamentally a finite cataloging problem rather than an open-ended reasoning problem. By treating it as such, we produced a calculator with complete coverage, transparent logic, and honest behavior on inputs that lie outside the named chord vocabulary.

The methodology of multi-model cross-reference combined with computational verification produced a more reliable result than any single model could have produced alone. The disagreements between models highlighted genuine ambiguities in music theory. The agreements identified the structural facts. The Python verification resolved the cases where models conflicted with each other or with themselves.

The resulting implementation occupies approximately 30 lines of Swift and handles every possible three-note input with theoretical justification. The calculator can be examined, criticized, and extended by anyone with sufficient music theory background, and its answers can be defended on first principles rather than appeal to authority.

This is the basis on which Intervallo presents its triad identification: a complete, verified, and honest enumeration of three-note chord structures in Western harmony.

A The 55 Interval Pairs

The complete combinatorial space of three-note structures within one octave, with named structures indicated in bold.

s_1	s_2	Chord Identity
1	2	Chromatic cluster
1	3	Cluster
1	4	Cluster
1	5	Cluster
1	6	m2 + tritone
1	7	m2 + P5
1	8	m2 + minor sixth
1	9	m2 + major sixth
1	10	m2 + m7
1	11	m2 + M7
2	3	Cluster
2	4	Whole-tone fragment
2	5	Suspended fragment
2	6	Lydian fragment
2	**7**	**Suspended 2nd**
2	8	Minor sixth fragment
2	**9**	**Suspended 4th, second voicing**
2	10	Dominant ninth fragment
2	11	Major ninth fragment
3	4	Split-third cluster
3	5	Minor add4 fragment
3	**6**	**Diminished triad, root**
3	**7**	**Minor triad, root**
3	**8**	**Major triad, first inversion**
3	**9**	**Diminished triad, first inversion**
3	**10**	**Minor 7th shell**
3	**11**	**Minor-Major 7th shell**
4	5	Major add4 cluster
4	**6**	**Major flat 5**
4	**7**	**Major triad, root**
4	**8**	**Augmented triad**
4	**9**	**Minor triad, first inversion**
4	**10**	**Dominant 7th shell**
4	**11**	**Major 7th shell**
5	6	Diminished sus4
5	**7**	**Suspended 4th, root**
5	**8**	**Minor triad, second inversion**
5	**9**	**Major triad, second inversion**
5	**10**	**Quartal trichord**
5	11	Major 7 sus4 shell
6	7	Tritone cluster
6	8	Augmented fragment
6	**9**	**Diminished triad, second inversion**
6	**10**	**Half-diminished shell**
6	11	Major 7 flat 5 shell
7	8	Fifth cluster
7	9	Power chord plus sixth
7	**10**	**Dominant 7th no third**
7	**11**	**Major 7th no third**
8	9	Sixth cluster
8	10	Dominant flat 13 fragment
8	11	Major flat 13 fragment
9	10	Sixth seventh cluster
9	11	Sixth major seventh cluster
10	11	Seventh cluster

Twenty-four of fifty-five pairs receive names in Intervallo's current implementation. The remaining thirty-one pairs return unnamed status because they do not correspond to standard chord structures in Western harmony, though many of them have meaningful interpretations in post-tonal, modal jazz, or texture-based composition contexts.

B Two Pairs Deserving Special Discussion

B.1 The Pair $(3, 5)$

The pair $(3, 5)$ represents a minor third followed by a perfect fourth. Some sources name this "minor flat five," treating it as a minor triad with a flattened fifth. Other sources treat it as an incomplete fragment of a half-diminished seventh chord.

Intervallo does not name this pair. The reasoning is that "minor flat five" is not a stable functional chord in Western tonal music, and the half-diminished interpretation requires a fourth note that is not present. Naming the pair would force one interpretation when neither has clear priority. Returning unnamed status preserves the user's ability to make their own contextual judgment.

B.2 The Pair $(5, 7)$

The pair $(5, 7)$ represents a perfect fourth followed by a perfect fifth. The conventional name is "suspended fourth chord." However, the same three pitch classes can be read as a suspended second chord built on the fourth above the bass note. C-F-G can be read as Csus4 or as Fsus2 in second inversion.

Intervallo names this pair as Csus4, prioritizing the bass-relative reading. A future version may surface the alternative interpretation as a secondary result.

References

[1] Forte, Allen. *The Structure of Atonal Music*. Yale University Press, 1973.

[2] Piston, Walter. *Harmony*. W.W. Norton & Company, 1987.

[3] Kostka, Stefan M., Dorothy Payne, and Byron Almén. *Tonal Harmony*. McGraw-Hill, 2017.

This paper is a living document. Corrections, disagreements, and proposed additions are welcome at `https://www.intervallo.app`.