

USING HARMONIC AND MELODIC ANALYSES TO AUTOMATE THE INITIAL STAGES OF SCHENKERIAN ANALYSIS

Phillip B. Kirlin

Department of Computer Science, University of Massachusetts Amherst

pkirlin@cs.umass.edu

ABSTRACT

Structural music analysis is used to reveal the inner workings of a musical composition by recursively applying reductions to the music, resulting in a series of successively more abstract views of the composition. Schenkerian analysis is the most well-developed type of structural analysis, and while there is a wide body of research on the theory, there is no well-defined algorithm to perform such an analysis. A automated algorithm for Schenkerian analysis would be extremely useful to music scholars and researchers studying music from a computational standpoint. The first major step in producing a Schenkerian analysis involves selecting notes from the composition in question for the primary soprano and bass parts of the analysis. We present an algorithm for this that uses harmonic and melodic analyses to accomplish this task.

1. INTRODUCTION

Numerous tasks in music information retrieval could be accomplished more effectively if information about musical structure were readily available. For example, in the task of retrieving musical passages that are similar to a given passage, having structural analyses available would allow similarity metrics to be based on the underlying musical structure of a composition as well as on the musical surface. An algorithm for structural analysis of music would therefore be an indispensable resource in music information research.

Schenkerian analysis [1] is a type of music analysis that emphasizes finding structural relationships among the notes of a composition. Developed by the Austrian music theorist Heinrich Schenker, Schenkerian analysis differs from other types of analysis that focus on a single aspect of music, such as the harmony or melody, to the exclusion of other aspects. Schenkerian analysis harnesses all aspects of a piece together to create an analysis that explains how various notes in the piece function in relation to others.

Of particular importance in Schenkerian analysis is the identification of *structural dependences* among groups of

notes. If the way in which a note X functions in a musical passage is due to the presence of another note or group of notes Y , then X is said to be dependent upon Y , and Y is said to be at a higher structural level than X . The process of finding structural dependences proceeds recursively during an analysis. The final set of dependences can be depicted as a tree, with the surface-level notes as the leaves. With each structural dependence located, the more structurally important notes are elevated to higher levels.

Though a tree theoretically can show all the hierarchical levels of a Schenkerian analysis, typically analyses are illustrated through a sequence of *Schenker graphs*. These graphs are visual depictions of a few contiguous levels of the note hierarchy, using staves with notes as in common music notation, but using other notation symbols such as stems, beams, and slurs to show relationships among notes rather than timing or phrasing information.

Because Schenkerian analysis primarily focuses on the main melodic line and the main harmonic bass line of the music in question, Schenker graphs are often presented on two staves, with the primary melodic line on the upper staff and the supporting bass harmony tones on the lower staff. Notes of inner voices are occasionally shown on the graphs, but are sometimes omitted when they serve to only fill out the harmony. We focus on *foreground graphs*, the graphs that show the structural levels closest to the musical surface. A foreground graph is usually the first graph constructed when completing a Schenkerian analysis; all subsequent graphs are based — directly or indirectly — on the foreground graph. Therefore, it is critical to choose the correct set of notes to appear in the foreground graph. We will call a foreground graph, after notes have been selected for its staves but before any reductions have been applied, a *preliminary foreground graph*. Consider the first eight measures of Schubert's *Impromptu No. 2 in A-flat major*, shown in Figure 1. A preliminary foreground Schenker graph for the *Impromptu*, with appropriate notes in the soprano and bass parts, would look like Figure 2.

In this paper, we present and analyze an algorithm, FOREGRAPH, for identifying which notes in a score should belong on the soprano and bass staves of a preliminary foreground Schenker graph, based on analyses of harmony and voice leading. We build on the work of Kirlin and Utgoff [2], whose IVI system requires, as a first step, isolation of the primary soprano and bass parts prior to analysis. Automating Schenkerian analysis has been studied recently by Marsden [3–5] and Marsden and Wiggins [6]. These

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

© 2009 International Society for Music Information Retrieval.



Figure 1. An excerpt from Schubert's *Impromptu No. 2* in A-flat major.

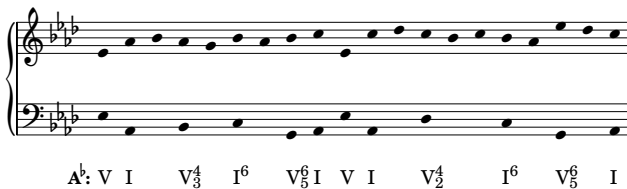


Figure 2. A preliminary foreground graph constructed by hand from the *Impromptu*.

lines of work are promising, but they have only been tested on short, sometimes synthetic, musical phrases. Lerdahl and Jackendoff developed a grammatical approach to musical structure in [7], which Hamanaka, et. al. [8] turned into an algorithm. Their system, however, requires manual adjustment of many parameters that differ for each musical composition. Older work by Kessler [9] and Smoliar [10] demonstrated understanding of the principles involved in automating analysis, but did not provide any algorithms.

2. COMPUTATIONAL METHODS FOR HARMONIC AND MELODIC ANALYSIS

Schenkerian analysis is based on the principles of harmony and voice leading. These two aspects of a composition must be examined prior to beginning an analysis. Since we desire a fully-automated system for producing foreground graphs, we must examine various algorithms for determining the harmony at various points in a composition, and the voice leading possibilities for any note in a piece.

We have chosen MusicXML as our representation of choice. MusicXML is a file format that represents common Western music notation by encoding the pitches and durations of notes. Though the MIDI representation is more widely used than MusicXML, the latter format encodes an additional wealth of information that the former does not supply, such as key and time signatures, stem and beaming information, and slurs and phrase marks.

One can look at harmonic analysis as occurring in two phases. First, a chord-labeling component assigns chord labels (such as “C Major”) to segments of a composition. A second pass then uses the chord labels to assign functional Roman numerals to segments.

2.1 Chord Labeling

A chord labeling component must divide a composition temporally into segments, where each segment corresponds

to a single harmony. We use a variant of Pardo and Birmingham's HARMAN algorithm [11] to accomplish this.

HARMAN uses two separate algorithms to perform chord labeling. The *labeling algorithm* is concerned with determining the best chord label for a given segment of music (a segment being an interval of time with fixed starting and ending times), and the *segmentation algorithm* determines the points in the music where the harmony changes. A harmony can change at a *partition point*: any place in the music where a note starts or stops.

While HARMAN does a very good job “out of the box,” we use a modified version of the algorithm and detail our changes below.

- **Meter** — HARMAN does not take the meter of the piece into account, and sometimes it chooses a partition point in a metrically weak position that is adjacent to a metrically stronger one. Because it is preferable to have a change of harmony in an analysis at a metrically strong position [12], we force HARMAN choose each measure boundary as a partition point.
- **Octave doubling** — Because HARMAN analyzes each note in a segment independently, often notes that are doubled at the octave exert too much of an influence over the chord labeling algorithm. Therefore, when analyzing a segment, we consider multiple instances of notes with the same pitch class and duration as a single note. For example, in Figure 1, the notes of the melody in measures 5–7 that are doubled at the octave will not be counted twice.
- **Neighbor tones** — Our version of HARMAN ignores “obvious” neighbor tones within segments. An obvious neighbor tone is a note Y that occurs in a note sequence $X - Y - Z$ where X and Z have the same pitch, and are separated from Y by a half-step. Without this correction, HARMAN has trouble distinguishing between chord tones and non-chord tones in heavily figured contexts.

2.2 Assignment of Roman Numerals

Given a chord labeling, the remaining task in harmonic analysis is mapping the chord labels (such as “C Major”) to Roman numeral labels (such as “V⁶”). While we have investigated algorithms for computing the key of a composition and the locations of any modulations and tonicizations, we restrict ourselves for the remainder of this dis-

cussion to non-modulating pieces whose key is encoded correctly in their MusicXML representation. As FOREGRAPH, the foreground Schenker graph generation algorithm presented in the next section, relies on a correct Roman numeral analysis, placing this restriction on the input music makes us more certain that we are supplying correct Roman numerals to FOREGRAPH.

The second phase detects tonicizations by looking for consecutive chords where the first chord functions as a temporary dominant to the second. For example, in the key of C major, this would detect the chord sequence “D Major – G Major” and change the harmonic analysis of “II – V” to “V/V – V.” We stipulate that the first chord cannot occur normally in the original key, to eliminate the possibility of the common “I – IV” chord sequence being reinterpreted as a tonicization of the IV chord.

2.3 Voice Leading Analysis

A voice leading analysis determines, for every note in the piece, which notes could logically follow from that note, according to the principles of voice leading [12]. Algorithms for determining voice leading, however, can differ in their interpretations of *implicit polyphony* [13]. For example, given the four notes in the first measure of Figure 3, some algorithms would determine that all four notes belong to a single voice, whereas others would find two voices and interpret the four notes as standing for the triads shown in the second measure. The second interpretation is an example of implicit polyphony.



Figure 3. An example where the voice leading is ambiguous.

Schenkerian analysis, as it gives primary consideration to the linear connections in music [14], requires a voice leading analysis that uncovers implicit polyphony. A reasonable way to handle this is to permit a voice-leading connection between two notes only if the motion between them is stepwise.

If one takes this stance, it is easy to construct an algorithm for determining the voice leading for a given composition. For a note n in a piece, we examine the set of notes that begin at times later than the ending of n (there cannot be a voice-leading connection between two notes that overlap in time). Note n may have up to three voice leading connections: (1) a step-down connection, (2) a step-up connection, and (3) a same-pitch connection. For each type of connection, we find the earliest note that satisfies the criteria for that kind of connection. We also require that if n has a same-pitch voice-leading connection to a note m , then n may not have any stepwise voice-leading connections to notes that begin later than m . This is because voice-leading connections between notes of identical pitch are typically stronger than stepwise connections.

3. PRODUCING PRELIMINARY FOREGROUND GRAPHS

Recall that our goal is to produce preliminary foreground Schenker graphs like the one in Figure 2. Since the purpose of a foreground graph is to capture the primary soprano and bass tones of the piece, constructing such a graph reduces to selecting notes for the soprano and bass parts.

In most circumstances, the primary melody (soprano) tone is the highest one heard at any point in time, and the primary bass tone is the lowest. Therefore, FOREGRAPH is based on the idea of selecting the highest pitch for the soprano line and the lowest for the bass line. However, complications arise in situations where the primary bass or soprano tones persist in time even though they may have stopped sounding. Consider an Alberti bass line, such as in Figure 4. Because this figure is outlining a chord, only the lowest note of the chord belongs to the primary bass line (the other notes belong to inner voices). The low Cs, though they are only represented on the page as eighth notes, persist in the musical mind through the entire measure as if they were sounding constantly; the true bass line does not skip between the notes of the chord. This is the reason why we require a voice leading algorithm that can detect cases of implicit polyphony, not just in cases of arpeggiation, but in any case where the bass or soprano part may move between voices.



Figure 4. An Alberti bass line.

Still, there are cases where voices start and stop mid-composition, and an algorithm that blindly follows the initial bass and soprano lines stepwise from the start of the piece to the end would not suffice in cases, for example, of register transfer. Therefore, FOREGRAPH chooses appropriate bass and soprano tones for each harmonic segment defined by the harmonic analysis algorithm, and then follows the tones via voice-leading connections to fill out the segment; pseudocode is given in Figure 5. For each harmonic segment in a composition, FOREGRAPH finds the lowest and highest pitched notes that belong to the current harmony; these notes are added to the primary bass and soprano parts. The FILLRANGE procedure then adds additional notes by following voice-leading connections from the initial notes added in the segment; connections are followed both backwards and forwards in time, and notes are only added if they do not overlap in time with any other notes already added to the segment.

The EXTENDVOICE procedure then allows the musical line in a harmonic segment to be extended into following segments, stopping only upon reaching a note that is consonant in the prevailing harmony for the segment. Because the primary notes are determined independently for each harmonic segment, it is possible that the soprano or bass lines fleshed out by FILLRANGE will not connect musically over a segment break. EXTENDVOICE permits each

line to be followed to a logical conclusion without adding too many notes of what may develop into an irrelevant inner voice. Because EXTENDVOICE halts upon adding a consonant note in the prevailing harmony, leaps are possible in the computed musical lines over segment breaks.

After choosing the notes for the soprano and bass lines, they are displayed as noteheads on staves as a preliminary foreground graph. FOREGRAPH produced the output shown in Figure 6 for the Schubert Impromptu in Figure 1.

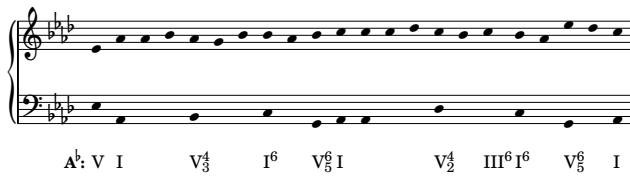


Figure 6. A preliminary foreground graph produced by FOREGRAPH.

If one compares the hand-constructed graph in Figure 2 to the one produced by FOREGRAPH in Figure 6, only a few differences are apparent. One is that the computer-constructed graph contains instances of adjacent notes of identical pitch. FOREGRAPH does not reduce these cases to single notes because although this occurs frequently in foreground graphs, it is not always done consistently.

The only other differences in the computer-generated analysis are the omitted “V” chord near the middle of the analysis, and the added “III⁶” chord. Both of these differences derive from the harmonic analysis component used as a preliminary step to FOREGRAPH. The V chord in the hand-constructed graph was not generated in the computer analysis as it was absorbed into the I chords on either side. Similarly, the first-inversion III chord arises from a misinterpretation of chord tones and non-chord tones.

4. EVALUATION AND ANALYSIS

In order to evaluate the correctness of FOREGRAPH, we require a set of input music scores and correct foreground graphs for them. We turned to a standard Schenkerian analysis textbook [14], and encoded the first twelve musical examples that had correct analyses provided, and whose analyses contained soprano and bass parts (two of the examples were monophonic, and so were omitted). The examples are all multi-measure excerpts from common practice period works.

Our method of evaluation is based on the standard metrics of precision and recall. If one views each note in a composition as an individual document, then constructing a preliminary foreground graph is equivalent to executing two queries: one query to retrieve all notes belonging to the soprano part, and a second query to retrieve all notes belonging to the bass part. We also need to define what it means for a note to be “relevant” and “retrieved” to compute precision and recall. We consider a note “retrieved” for a query if it appears in the corresponding part (soprano or bass) for the computer-constructed foreground graph. Defining “relevant” is complicated because the foreground

graphs as they appear in the textbook (1) often contain pertinent pitches of inner voices along with the primary soprano and bass parts, and (2) already have had some reductions applied in most cases, which removes some notes from the ground-truth that would appear in the computer-generated graphs.

Therefore, we have two notions of “relevant” and compute statistics based on each definition. In our first set of calculations, we consider a note to be relevant for the soprano (bass) query if it is present on the upper (lower) staff of the Schenker graph in the textbook analysis. This definition, however, considers many notes as relevant that will not be present in the computer-generated analyses as they belong to inner voices. To remedy this, our second definition considers a note to be relevant for the soprano (bass) query if it is present on the upper (lower) staff in the Schenker graph in the textbook analysis, and has a stem pointing up (down). If it is clear that stem direction in a graph is *not* being used to indicate to which voice a note belongs (and the direction is only determined by aesthetics), the restriction on stem direction is ignored, and only the presence of the stem is considered. Stems in graphs are indications of structural importance, and therefore these are notes that we are particularly interested in having FOREGRAPH identifying correctly.

We ran the FOREGRAPH algorithm on each example and compared the resulting graphs to the textbook’s graphs. For each example, and for each part (soprano and bass), we computed precision (the fraction of retrieved notes that were also relevant) and recall (the fraction of relevant notes that were also retrieved). To provide a baseline for comparison with FOREGRAPH, we evaluated a second foreground graph creation algorithm, RANDOM, that selects notes for the soprano and bass parts from the input music randomly. RANDOM always chooses the same number of notes for the soprano and bass parts for each example as were selected by FOREGRAPH for the same example. We calculated average precision and recall for RANDOM over 500 runs. All of the precision and recall statistics for FOREGRAPH and RANDOM are displayed in Table 1. To show more clearly the improvement of FOREGRAPH over RANDOM, Figure 7 compares the F1 measure (harmonic mean of precision and recall) for each musical example for the two algorithms.

One of the excerpts deserves special mention. The excerpt from Schubert’s *Symphony in B minor* confused FOREGRAPH as the accompaniment part is pitched higher than the primary melody. The analysis constructed by FOREGRAPH contained a harmony line in the soprano part, and the true melody was not present at all. Because this single example distorted the statistics for the soprano part, Table 1 contains entries for the aggregate precision and recall with and without the *Symphony* included.

Overall, we are encouraged by the results of the evaluation. We are especially pleased with the recall values for the stemmed notes definition of relevance; disregarding the Schubert *Symphony*, FOREGRAPH retrieved almost 90% of the relevant bass notes, and almost 80% of the relevant soprano notes. Figure 7 clearly indicates that FOREGRAPH

procedure FOREGRAPH

Let $V(x, y)$ be true if there is a voice leading connection between notes x and y .

Let S be a set of notes for the primary soprano part.

Let B be a set of notes for the primary bass part.

for each harmonic segment H in the composition **do**

Let n be the lowest pitched note in H that is a member of H 's harmony.

Add n to B

FILLRANGE(n, B, H)

EXTENDVOICE(B, H)

Let n be the highest pitched note in H that is a member of H 's harmony.

Add n to S

FILLRANGE(n, S, H)

EXTENDVOICE(S, H)

procedure FILLRANGE(note n , part P , harmonic segment H)

Initialize queue Q to contain just n

while Q is not empty **do**

Remove the top note from the queue, call it m

Let N be the set of all notes such that if $x \in N$, then either $V(m, x)$ or $V(x, m)$, and x is in H .

Sort N by increasing length of time between m and each note in N

if N is empty, **then return**

for each note $x \in N$ **do**

if x does not conflict with any notes in P **then** add x to P and add x to Q

procedure EXTENDVOICE(part P , harmonic segment H)

Let $curr$ be the last note in H that is also in P

while $curr$ is not consonant in H 's harmony **do**

Let N be the set of all notes such that if $x \in N$, then $V(m, x)$

if N is empty, **then return**

Let n be the note in N with the minimum length of time to $curr$

Add n to P

$curr \leftarrow n$

Figure 5. The FOREGRAPH algorithm.

is a large improvement over choosing notes randomly.

The two issues mentioned earlier that complicated choosing an appropriate definition of relevance cause the precision and recall values to not represent the true quality of the graphs produced by FOREGRAPH. The first issue is that many of the ground-truth analyses contain notes of inner voices on the upper and lower staves, as well as notes from the primary soprano and bass parts. The bass part of the Chopin *Nocturne*, for example, contains arpeggiated chords. FOREGRAPH only included the lowest note of each chord in the primary bass part, while the textbook included all of the notes of each chord, with all but the lowest given as inner voices. This lowered the recall value for all bass notes in this example to 23.5%.

The second issue is that many of the textbook's graphs have already had simple reductions applied to the musical surface; repeated notes in the textbook's graphs have also been removed in many cases. Because FOREGRAPH only *selects* notes for the foreground graphs and does not perform any reductions, many of the precision values are lower than they would be if those reductions had not been done in the textbook's graphs. For example, in the *French Suite*; FOREGRAPH placed many notes in the soprano part that were not present in the textbook's graph because reductions had already been applied to them.

We are confident that FOREGRAPH is ready to be used as a precursor to an actual Schenkerian reduction algorithm. Because we are only selecting notes to be placed in the soprano and bass parts, the output of FOREGRAPH is ready for processing to search for reductions, and any low precision statistics should not be alarming.

5. REFERENCES

- [1] Heinrich Schenker. *Der Freie Satz*. Universal Edition, Vienna, 1935. Published in English as *Free Composition*, translated and edited by E. Oster, Longman, 1979.
- [2] Phillip B. Kirlin and Paul E. Utgoff. A framework for automated Schenkerian analysis. In *Proceedings of the Ninth International Conference on Music Information Retrieval*, pages 363–368, Philadelphia, September 2008.
- [3] Alan Marsden. Automatic derivation of musical structure: A tool for research on Schenkerian analysis. In *Proceedings of the Eighth International Conference on Music Information Retrieval*, pages 55–58, 2007. Extended Version.
- [4] Alan Marsden. Generative structural representation

Excerpt	All notes				Stemmed notes			
	Precision		Recall		Precision		Recall	
	Sop.	Bass	Sop.	Bass	Sop.	Bass	Sop.	Bass
J. S. Bach, <i>Aria variata</i> (BWV 989)	0.320	0.438	0.889	0.700	0.200	0.250	1.000	0.667
Beethoven, <i>Ninth Symphony</i> , III	0.818	0.800	0.818	1.000	0.455	0.300	0.714	1.000
Haydn, <i>Symphony in D major</i> , No. 104	0.667	0.538	0.706	0.875	0.389	0.462	0.875	0.857
Chopin, <i>Prelude in A major</i> , Op. 28/7	0.636	0.333	0.636	1.000	0.091	0.333	0.500	1.000
Haydn, <i>Divertimento in B-flat</i> , II	0.643	0.615	1.000	1.000	0.286	0.308	1.000	1.000
Schubert, <i>Standchen</i> from <i>Schwanengesang</i>	0.778	0.571	0.280	1.000	0.556	0.429	0.556	1.000
J. S. Bach, Chorale No. 149	0.800	0.857	1.000	0.857	0.600	0.429	1.000	0.750
J. S. Bach, <i>French Suite in C minor</i> , Sarabande	0.550	0.750	0.786	0.900	0.250	0.333	0.714	0.800
Schubert, <i>Symphony in B minor</i> , No. 8, I	0.000	0.667	0.000	1.000	0.000	0.667	0.000	1.000
Mozart, <i>Symphony in C major</i> , K. 425, IV	0.636	0.174	0.609	0.500	0.273	0.130	1.000	1.000
Chopin, <i>Nocturne in D-flat major</i> , Op. 27/2	0.889	0.400	0.533	0.235	0.333	0.400	0.500	1.000
Brahms, <i>Rhapsody in E-flat major</i> , Op. 119/4	1.000	1.000	0.917	1.000	0.364	0.636	1.000	1.000
FOREGRAPH, all excerpts	0.568	0.557	0.568	0.772	0.279	0.364	0.729	0.911
FOREGRAPH, all except for Schubert's <i>Symphony</i>	0.650	0.547	0.675	0.753	0.319	0.336	0.797	0.896
RANDOM, all excerpts	0.255	0.154	0.255	0.213	0.104	0.083	0.273	0.208
Improvement of FOREGRAPH over RANDOM	0.313	0.403	0.313	0.559	0.175	0.281	0.456	0.703

Table 1. Precision and recall for evaluation.

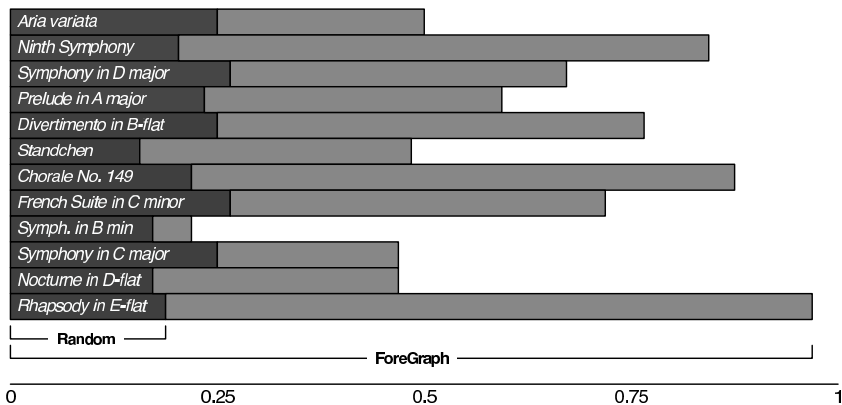


Figure 7. A graph comparing the F1 measures for RANDOM (dark bars) and for FOREGRAPH (light bars).

- of tonal music. *Journal of New Music Research*, 34(4):409–428, December 2005.
- [5] Alan Marsden. Extending a network-of-elaborations representation to polyphonic music: Schenker and species counterpoint. In *Proceedings of the First Sound and Music Computing Conference*, pages 57–63, 2004.
 - [6] Alan Marsden and Geraint A. Wiggins. Schenkerian reduction as search. In *Proceedings of the Fourth Conference on Interdisciplinary Musicology*, Thessaloniki, Greece, July 2008.
 - [7] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, Massachusetts, 1983.
 - [8] Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo. ATTA: Implementing GTTM on a computer. In *Proceedings of the Eighth International Conference on Music Information Retrieval*, pages 285–286, 2007.
 - [9] Michael Kassler. APL applied in music theory. *APL Quote Quad*, 18(2):209–214, 1987.
 - [10] Stephen W. Smoliar. A computer aid for Schenkerian analysis. *Computer Music Journal*, 2(4):41–59, 1980.
 - [11] Bryan Pardo and William P. Birmingham. Algorithms for chordal analysis. *Computer Music Journal*, 26(2):27–49, 2002.
 - [12] Edward Aldwell and Carl Schachter. *Harmony and Voice Leading*. Harcourt Brace & Company, Fort Worth, Texas, second edition, 1989.
 - [13] David Temperley. *The Cognition of Basic Musical Structures*. MIT Press, Cambridge, Massachusetts, 2001.
 - [14] Allen Forte and Steven E. Gilbert. *Introduction to Schenkerian Analysis*. W. W. Norton and Company, New York, 1982.