

# A DATA SET FOR COMPUTATIONAL STUDIES OF SCHENKERIAN ANALYSIS

Phillip B. Kirlin

Department of Mathematics and Computer Science, Rhodes College  
kirlinp@rhodes.edu

## ABSTRACT

Schenkerian analysis, a kind of hierarchical music analysis, is widely used by music theorists. Though it is part of the standard repertoire of analytical techniques, computational studies of Schenkerian analysis have been hindered by the lack of available data sets containing both musical compositions and ground-truth analyses of those compositions. Without such data sets, it is difficult to empirically study the patterns that arise in analyses or rigorously evaluate the performance of intelligent systems for this kind of analysis. To combat this, we introduce the first publicly available large-scale data set of computer-processable Schenkerian analyses. We discuss the choice of musical selections in the data set, the encoding of the music and the corresponding ground-truth analyses, and the possible uses of these data. As an example of the utility of the data set, we present an algorithm that transforms the Schenkerian analyses into hierarchically-organized data structures that are easily manipulated in software.

## 1. CORPUS-DRIVEN RESEARCH

Corpus-driven research is now commonplace in the music informatics community. With the wealth of raw musical information now available in digital form, in many cases, it is straightforward to construct and use data sets containing numerous musical compositions. However, the problem of collecting ground-truth metadata about the content of the music still exists, especially where high-level features are concerned. This is a problem that affects researchers working with music in audio or symbolic formats.

Ground-truth data sets that include features specifically relating to music theory or music analysis are particularly labor-intensive to construct. Information about the high-level harmonic or melodic structure of compositions is often only found scattered throughout textbooks or individual research publications, and so there are few publicly-available corpora containing such information in a computer-processable format. Some data sets are created only for specific research projects and then discarded,

are not in an easy-to-use format, or are simply never made widely available.

The lack of varied ground-truth musical metadata relating to theory and analysis — especially data sets specifically designed to align with symbolic music data — hinders corpus-driven research studies because time must be spent collecting data. Sometimes the researchers must perform the music analysis themselves, possibly inadvertently introducing biases into the data. Without widely available comprehensive data sets, it is extremely difficult to conduct large-scale experiments on the structure of musical compositions in symbolic form, or quantitatively evaluate the performance of computational systems that emulate a music analysis process.

There is a particular dearth of empirical data available in the realm of *Schenkerian analysis*, a widely used analytical system that illustrates a hierarchical structure among the notes of a composition. Though Schenkerian analysis is one of the most comprehensive methods for music analysis that we have available today [1], there are no large-scale digital repositories of analyses available to researchers. In addition to the reasons stated above for the lack of corpora, Schenkerian analysis presents a number of unique challenges to creating a useful data set. First, a Schenkerian analysis for a composition is illustrated using the musical score of the composition itself, and commonly requires multiple staves to show the hierarchical structure uncovered. This requires substantial space on the printed page and thus is a deterrent to retaining large sets of analyses. Second, there is no established computer-interpretable format for Schenkerian analysis storage, and third, even if there were a format, it would take a great deal of effort to encode a number of analyses into processable computer files.

The lack of data has kept the number of computational studies of Schenkerian analysis requiring ground-truth data to a bare minimum; some examples include studies using corpora with six [7] or eight [6] pieces. Though these studies are useful, the results would likely carry more weight if the data sets used were larger.

With all of these ideas in mind, in this paper we introduce the first large-scale data set of musical compositions along with corresponding ground-truth Schenkerian analyses, called SCHENKER41<sup>1</sup>. The 41 musical selections included constitute the largest-known corpus of Schenkerian analyses in a machine-readable format. The musical selec-

<sup>1</sup> Available at [www.cs.rhodes.edu/~kirlinp/schenker41](http://www.cs.rhodes.edu/~kirlinp/schenker41).



© Phillip B. Kirlin.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Phillip B. Kirlin. “A Data Set For Computational Studies of Schenkerian Analysis”, 15th International Society for Music Information Retrieval Conference, 2014.

tions are standardized in mode, length, and instrumentation, and the analyses are stored in a novel text-based representation designed to be easily processed by a computer. We created these data with the hope that they would be useful to researchers (a) studying the Schenkerian analysis process itself from a quantitative standpoint (for instance, detecting patterns in the way analysis is done), (b) needing a data set of analyses for use with supervised machine learning techniques, and (c) performing any sort of quantitative evaluation requiring ground-truth hierarchical music analyses.

## 2. THE DATA SET

### 2.1 Creation and Content

In order to create a data set of musical compositions and corresponding ground-truth Schenkerian analyses that would be useful to researchers with a wide variety of goals, we restricted ourselves to music from the common practice period of European art music, and selected 41 excerpts from works by J. S. Bach, G. F. Handel, Joseph Haydn, M. Clementi, W. A. Mozart, L. van Beethoven, F. Schubert, and F. Chopin. All of the compositions were either for a solo keyboard instrument (or arranged for such an instrument) or for voice with keyboard accompaniment. All were in major keys and did not modulate.

The musical excerpts were also selected for the ease of locating a Schenkerian analysis for each excerpt done by an outside expert. Analyses for the 41 excerpts chosen came from four places: Forte and Gilbert's textbook *Introduction to Schenkerian Analysis* [4] and the corresponding instructor's manual [3], Cadwallader and Gagné's textbook *Analysis of Tonal Music* [2], Pankhurst's handbook *SchenkerGUIDE* [9], and a professor of music theory who teaches a Schenkerian analysis class. These four sources are denoted by the labels F&G, C&G, SG, and Expert in Table 1, which lists the excerpts in the corpus.

From a Schenkerian standpoint, we also chose excerpts such that the analyses of the excerpts would all share some commonalities. All the analyses contained a single linear progression as the fundamental background structure: either an instance of the *Ursatz* or a rising linear progression. Some excerpts contained an *Ursatz* with an *interruption*: a Schenkerian construct that occurs when a musical phrase ends with an incomplete instance of the *Ursatz*, then repeats with a complete version.

We put these restrictions on the musical content in place because we expected that if SCHENKER41 were to be used for supervised machine learning, such algorithms would be able to better model a corpus with less variability among the pieces.

Overall, SCHENKER41 contains 253 measures of music and 907 notes. The lengths of individual excerpts ranged from 6 to 53 notes.

### 2.2 Encoding

With our selected musical excerpts and our corresponding analyses in hand, we needed to translate the musical in-

formation into machine-readable form. Musical data has many established encoding schemes; we used MusicXML, a format that preserves more information from the original score than say, MIDI.

Translating the Schenkerian analyses proved harder because there is no current standard for storing such analyses in a format that a computer could easily process and manipulate. Therefore, we devised a text-based encoding scheme to represent the various notations found in a Schenkerian analysis. Each analysis is stored in a single text file that is linked to a specific MusicXML file containing the musical excerpt being analyzed.

Schenkerian analyses are primarily based on the concept of a *prolongation*, a situation where an analyst determines that a group of notes is elaborating a group of more structurally fundamental notes. Consider the descending melodic pattern D–C–B–F $\sharp$ –G all occurring over G major harmony, as is shown in Figure 1. We could imagine that an analyst would determine that this passage outlines a descending G-major triad (D–B–G), with the second note C (a passing tone) serving to melodically connect the preceding D to the following B. We would say the note C *prolongs* the motion from D to B. Similarly, the F $\sharp$  prolongs the motion from B to G. Schenkerian analysis hypothesizes that any tonal composition is structured as a nested collection of prolongations; identifying them is an important component of the analysis procedure.

Every prolongation identified in an analysis is encoded in the analysis text file using the syntax  $X(Y)Z$ , where  $X$  and  $Z$  are individual notes in the score and  $Y$  is a non-empty list of notes. Such a statement means that the notes in  $Y$  prolong the motion from note  $X$  to note  $Z$ . Additionally, we permit incomplete prolongations in the text file representation: one of  $X$  or  $Z$  may be omitted. The notes of  $X$ ,  $Y$  and  $Z$  are transcribed in the text file as is shown in Figure 1, with a measure number, followed by a pitch and octave, followed by an integer to distinguish between repeated notes in the same measure. Figure 2 shows how the prolongations of Figure 1 would be encoded. Note that the prolongation involving the F $\sharp$  is encoded with no  $X$  component; this tells us that there is no strong melodic connection from the B to the F $\sharp$ , only from the F $\sharp$  to the G.

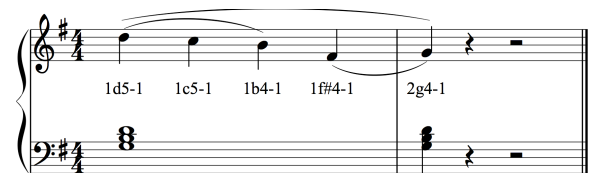


Figure 1. A melodic sequence with note names.

```
1d5-1 (1c5-1) 1b4-1
(1f#4-1) 2g4-1
1d5-1 (1b4-1) 2g4-1
```

Figure 2. An encoding of the prolongations present.

This text format easily supports encoding prolongations at differing hierarchical levels in the music. We can see

how Figure 2 encodes both the “surface-level” prolongations D–C–B and F $\sharp$ –G, but also the deeper prolongation D–B–G which outlines the fifth relationship in the G-major chord.

Aside from prolongations, the encoding system supports describing repetitions of notes that may be omitted in the analysis on the printed page; any linear progressions, including instances of the *Ursatz*; and the harmonic context present at any point in the analysis.

### 2.3 Compromises

An additional challenge not previously mentioned in creating the SCHENKER41 analyses is choosing an appropriate level of detail of the material to encode. Because the main objects in analyses are prolongations, it is natural to attempt to group them into categories like “neighbor tone” and “passing tone.” However, not all prolongations identified in analyses are easily categorized, and so category labels are often omitted in analyses not found in an educational context. This raises the question of whether or not to attempt to encode the category of prolongations in this corpus. To avoid the risk of incorrectly interpreting analyses, we have chosen to encode only what is *directly observable on the printed page* — the hierarchical relationship between groups of notes — and not categorize the prolongations found in the analyses. We recognize that this is a compromise between staying true to the data and encoding all potentially useful information.

## 3. USAGE OF THE DATA

The SCHENKER41 data set enables the undertaking of a wide variety of tasks and studies. In addition to the already-discussed endeavors of using the corpus for supervised machine learning or for quantitative evaluation, we theorize that with these data it could be possible to address the following questions:

- Do analysts identify certain types of prolongations more often than others under certain circumstances? These circumstances may involve the composer, musical genre, or even the analysis source.
- Does Schenkerian analysis align well with other forms of music analysis, such as Narmour’s implication-realization model of melodic expectation [8]?
- How well do Schenkerian analyses align with expressive performances of the music [10]? Do features of a performance such as phrasing, volume, or other quantifiable measures of musicality correspond to various Schenkerian annotations in an analysis?

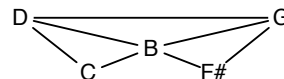
Besides answering questions about Schenkerian analysis itself, we hope that the availability of SCHENKER41 will spur others to study the utility of Schenkerian analysis in other areas of music informatics. For instance, we suspect hierarchical analyses could prove useful in constructing musical similarity metrics, because Schenkerian

analyses may highlight a common melodic pattern residing under the surface in two different musical excerpts.

Though the SCHENKER41 analyses can be directly processed by software, the nature of the flat text file format in which the data are encoded makes it difficult to see hierarchical relationships between notes not directly related by a single prolongation. Therefore, in this section we describe an algorithm to translate the analysis text files into hierarchical graph structures known as MOPs. It is possible to use the SCHENKER41 data in MOP form to automatically learn characteristics of Schenkerian analysis [5].

### 3.1 Maximal Outerplanar Graphs

Maximal outerplanar graphs, or *MOPs*, were first proposed by Yust [11] as elegant structures for representing a set of musical prolongations in a Schenkerian-style hierarchy. A MOP represents a hierarchy of melodic intervals located in a monophonic sequence of notes, though Yust proposed some extensions for polyphony. For example, the prolongations mentioned in Figures 1 and 2 are represented by the MOP shown in Figure 3.



**Figure 3.** A MOP representation of the music in Figure 1.

Formally, a MOP is a complete triangulation of a polygon, where the vertices of the polygon are notes and the outer perimeter of the polygon consists of the melodic intervals between consecutive notes of the original music, except for the edge connecting the first note to the last, which is called the *root edge*. Each triangle in the polygon specifies a prolongation. For instance, in Figure 3, the presence of triangle D–C–B means that the melodic motion from D to B is prolonged by the C. By expressing the hierarchy in this fashion, each edge  $(x, y)$  carries the interpretation that notes  $x$  and  $y$  are “consecutive” at some level of abstraction of the music. Edges closer to the root edge express more abstract relationships than edges farther away.

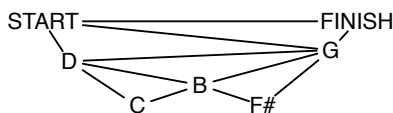
*Outerplanarity* is a property of a graph that can be drawn such that all the vertices are on the perimeter of the graph. Such a condition is necessary for us to enforce the strict hierarchy among the prolongations. A *maximal* outerplanar graph cannot have any additional edges added to it without destroying the outerplanarity; such graphs are necessarily polygon triangulations, and under this interpretation, all prolongations must occur over triples of notes.

There are three representational issues with MOPs we must address before discussing the algorithm to convert an analysis text file into MOPs. First, Schenkerian analyses as commonly encountered often include prolongations involving more than three notes. The analysis sources used in SCHENKER41 are no exception. For this reason, we relax the “maximal” qualifier for MOPs and permit prolongations involving any number of notes in our MOP representation. A prolongation involving more than three notes

will be translated into a polygon with more than three edges in the MOP representation.

Second, MOPs do not have a direct way to represent a prolongation with only a single “parent” note. Because MOPs model prolongations as a way of moving *from* one musical event *to* another event, every prolongation must have two parent notes. Music sometimes presents situations, however, that an analyst would model with a one-parent prolongation, such as an incomplete neighbor tone (we encountered this situation in Figure 2). Yust interprets such prolongations as having a “missing” origin or goal note that has been elided with a nearby structural note, which substitutes in the MOP for the missing note.

The third representational issue stems from trying to represent prolongations involving the first or last note in the music. Prolongations necessarily take place over time, and in a MOP, we interpret the temporally middle notes as prolonging the motion from the earliest note (the left parent) to the latest (the right parent). Following this temporal logic, we can infer that the root edge of a MOP must therefore necessarily be between the first note of the music and the last, implying these are the two most structurally important notes of a composition. As this is not always true in compositions, we add two pseudo-events to every MOP: an initiation event that is located temporally before the first note of the music, and a termination event, which is temporally positioned after the last note. The root edge of a MOP is fixed to always connect the initiation event and the termination event. These extra events allow for any melodic interval — and therefore any pair of notes in the music — to be represented as the most structural event in the composition. For instance, in Figure 4, which shows the D–C–B–F#–G pattern with initiation and termination events (labeled START and FINISH), the analyst has indicated that the G is the most structurally significant note in the passage, as this note prolongs the motion along the root edge.



**Figure 4.** A MOP containing initiation and termination events.

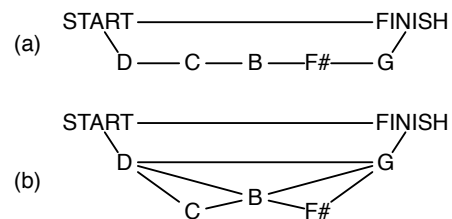
### 3.2 Converting the Corpus to MOPs

We now present an algorithm to convert a text file analysis like those in SCHENKER41 to a collection of MOPs. Because a single MOP only represents a monophonic sequence of notes, we may need multiple MOPs to store all of the prolongations in a single text file analysis. Most of the analyses in SCHENKER41 contain at least two MOPs, one representing the structure of the main melody, and one representing structure of the bass line.

The algorithm operates in three phases. In the first phase, we make a pass through the analysis text file to identify which notes will belong to which MOPs. We do this by creating a temporary graph structure consisting of all the

notes present in the analysis and initially no edges. For each prolongation in the analysis file  $X (Y) Z$ , we add the edges  $(X, Z)$  and  $(i, Z)$  for each note  $i$  in the set of notes  $Y$ . After processing every prolongation, every connected component in the graph will correspond to a single MOP.

Phase two adds edges to the MOP for all two-parent prolongations. For each MOP graph identified in phase one, we first remove all the edges, then create a “skeleton” MOP structure consisting of edges connecting only consecutive notes in the music, plus the additional edges involving the START and FINISH vertices. Figure 5(a) illustrates this skeletal structure for the prolongations described in Figure 2. We then create edges in the MOP corresponding to all prolongations in the analysis text file that have two parent notes. Adding appropriate edges is straightforward: for a prolongation  $X (Y) Z$ , we add an edge from note  $X$  to the first note of the set of notes  $Y$ , an edge from the last note of  $Y$  to note  $Z$ , and an edge from  $X$  to  $Z$ . If the consecutive notes of  $Y$  are not already connected to each other by edges, we also add such edges. At the end of phase two, we would have a structure like in Figure 5(b).



**Figure 5.** The (a) beginning and (b) end of phase two of creating a MOP.

Phase three involves adding edges in the MOP for one-parent prolongations, i.e., prolongations in the analysis text file of the form  $X (Y)$  or  $(Y) Z$ . We begin by adding edges between consecutive notes of  $Y$  as in phase two. The next step is identifying any additional edges necessary to enforce that the notes of  $Y$  should be lower in the hierarchy than  $X$  or  $Z$ , whichever parent note is present. Fortunately, it is guaranteed that every one-parent prolongation will fall into one of the six categories described below, each of which we handle separately. We briefly describe the six categories here, and their processing steps are fully described in the pseudocode of Algorithm 1. The code refers to the “smallest interior polygon” for a one-parent prolongation  $p$ , which is the smallest polygon in the MOP containing all the notes of  $p$  (the parent note and all of the child notes). This interior polygon will always exist in a MOP because MOPs express a strict hierarchy among the notes, and therefore all the notes of a prolongation will be found within a single polygon.

Category 1 corresponds to a one-parent prolongation missing a right parent, where the MOP already contains an edge connecting the left parent  $X$  to the first note of  $Y$ , and the edge in question already implies a hierarchical relationship between  $X$  and  $Y$ . In this situation, there are no extra edges to add because the necessary hierarchical relationship already exists. Category 2 corresponds to the same situation as Category 1, but reversed for a missing

**Algorithm 1**


---

```

1: procedure PROCESS-ONE-PARENT-PROLONGATIONS
2:   Let  $S$  be the set of one-parent prolongations.
3:   while  $S \neq \emptyset$  do
4:      $p \leftarrow$  shortest length prolongation in  $S$ 
5:      $I \leftarrow$  identify smallest interior polygon containing all notes of  $p$ 
6:     Assume vertices of  $I$  are numbered  $0 \dots m - 1$ 
7:     if  $\text{leftParent}(p) = I[0]$  and  $\text{firstChildNote}(p) = I[1]$  then ▷ Category 1
8:        $S \leftarrow S - \{p\}$  ▷ No additional edges needed;  $p$ 's children are already lower in the hierarchy
9:     else if  $\text{rightParent}(p) = I[m - 1]$  and  $\text{lastChildNote}(p) = I[m - 2]$  then ▷ Category 2
10:       $S \leftarrow S - \{p\}$  ▷ No additional edges needed;  $p$ 's children are already lower in the hierarchy
11:     else if  $\text{leftParent}(p) = I[0]$  then ▷ Category 3
12:       Add edge ( $\text{leftParent}(p)$ ,  $\text{firstChildNote}(p)$ ) to MOP;  $S \leftarrow S - \{p\}$ 
13:     else if  $\text{rightParent}(p) = I[m - 1]$  then ▷ Category 4
14:       Add edge ( $\text{rightParent}(p)$ ,  $\text{lastChildNote}(p)$ ) to MOP;  $S \leftarrow S - \{p\}$ 
15:     else if  $\text{rightParent}(p)$  is missing then ▷ Category 5
16:        $\text{newRight} \leftarrow$  earliest  $I[x]$  such that  $I[x]$  is later than all of  $p$ 's children
17:       if choice of  $\text{newRight}$  increases length of prolongation  $p$  then
18:         Update  $p$ 's length in  $S$ ; defer processing
19:       else
20:         Add edge ( $\text{leftParent}(p)$ ,  $\text{newRight}$ ) to MOP;  $S \leftarrow S - \{p\}$ 
21:     else if  $\text{leftParent}(p)$  is missing then ▷ Category 6
22:        $\text{newLeft} \leftarrow$  latest  $I[x]$  such that  $I[x]$  is earlier than all of  $p$ 's children
23:       if choice of  $\text{newLeft}$  increases length of prolongation then
24:         Update  $p$ 's length in  $S$ ; defer processing
25:       else
26:         Add edge ( $\text{newLeft}$ ,  $\text{rightParent}(p)$ ) to MOP;  $S \leftarrow S - \{p\}$ 

```

---

left parent note.

Category 3 corresponds to a one-parent prolongation missing a right parent, where the the MOP does *not* contain an edge connecting the left parent  $X$  to the first note of  $Y$ , but other nearby edges already imply a hierarchical relationship between  $X$  and  $Y$ . Here, we only need to add an edge from  $X$  to the first child note of  $Y$ . Category 4 corresponds to the same situation as Category 3, but reversed for a missing left parent.

Category 5 corresponds to a one-parent prolongation missing a right parent, where the the MOP does *not* contain an edge connecting the left parent  $X$  to the first note of  $Y$ , and *no other edges* in the MOP already imply a hierarchical relationship between  $X$  and  $Y$ . In this situation we must explicitly find a suitable right parent note, which we choose to be the temporally earliest note on the interior polygon that is later than all the notes of  $Y$ . Category 6 corresponds to the same situation as Category 5, but reversed for a missing left parent.

#### 4. CONCLUSIONS

In this paper, we presented SCHENKER41, the first large-scale data set of musical compositions and corresponding Schenkerian analyses in a computer-processable format. We anticipate that with the rise of corpus-driven research in music informatics, this data set will be of value to researchers investigating various characteristics of Schenkerian analysis, using machine learning techniques to study the analytical procedure, or harnessing the analyses for use in other music informatics tasks. We also presented an algorithm for translating the analyses into MOPs, which serve as useful data structures for representing the hierarchical organization of the analyses.

#### 5. REFERENCES

- [1] Matthew Brown. *Explaining Tonality*. University of Rochester Press, 2005.
- [2] Allen Cadwallader and David Gagné. *Analysis of Tonal Music: A Schenkerian Approach*. Oxford University Press, Oxford, 1998.
- [3] Allen Forte and Steven E. Gilbert. *Instructor's Manual for Introduction to Schenkerian Analysis*. W. W. Norton and Company, New York, 1982.
- [4] Allen Forte and Steven E. Gilbert. *Introduction to Schenkerian Analysis*. W. W. Norton and Company, New York, 1982.
- [5] Phillip B. Kirlin. *A Probabilistic Model of Hierarchical Music Analysis*. PhD thesis, University of Massachusetts Amherst, 2014.
- [6] Phillip B. Kirlin and David D. Jensen. Probabilistic modeling of hierarchical music analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 393–398, 2011.
- [7] Alan Marsden. Schenkerian analysis by computer: A proof of concept. *Journal of New Music Research*, 39(3):269–289, 2010.
- [8] Eugene Narmour. *Beyond Schenkerism: The Need for Alternatives in Music Analysis*. University of Chicago Press, 1977.
- [9] Tom Pankhurst. *SchenkerGUIDE: A Brief Handbook and Website for Schenkerian Analysis*. Routledge, New York, 2008.
- [10] Christopher Raphael. Symbolic and structural representation of melodic expression. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pages 555–560, 2009.
- [11] Jason Yust. *Formal Models of Prolongation*. PhD thesis, University of Washington, 2006.

Composer	Excerpt name	Analysis source
Bach	Minuet in G major, BWV Anh. 114, mm. 1–16	Expert
Bach	Chorale 233, Werde munter, mein Gemute, mm. 1–4	Expert
Bach	Chorale 317 (BWV 156), Herr, wie du willst, so schicks mit mir, mm. 1–5	F&G manual
Beethoven	Seven Variations on a Theme by P. Winter, WoO 75, Variation 1, mm.1–8	C&G
Beethoven	Seven Variations on a Theme by P. Winter, WoO 75, Theme, mm. 1–8	C&G
Beethoven	Ninth Symphony, Ode to Joy theme from finale (8 measures)	SG
Beethoven	Piano Sonata in F minor, Op. 2, No. 1, Trio, mm. 1–4	SG
Beethoven	Seven Variations on God Save the King, Theme, mm. 1–6	SG
Chopin	Mazurka, Op. 17, No. 1, mm. 1–4	SG
Chopin	Grande Valse Brilliante, Op. 18, mm. 5–12	SG
Clementi	Sonatina for Piano, Op. 38, No. 1, mm. 1–2	SG
Handel	Trio Sonata in B-flat major, Gavotte, mm. 1–4	Expert
Haydn	Divertimento in B-flat major, Hob. 11/46, II, mm. 1–8	F&G
Haydn	Piano Sonata in C major, Hob. XVI/35, I, mm. 1–8	F&G
Haydn	Twelve Minuets, Hob. IX/11, Minuet No. 3, mm. 1–8	SG
Haydn	Piano Sonata in G major, Hob. XVI/39, I, mm. 1–2	SG
Haydn	Hob. XVII/3, Variation I, mm. 19–20	SG
Haydn	Hob. I/85, Trio, mm. 39–42	SG
Haydn	Hob. I/85, Menuetto, mm. 1–8	SG
Mozart	Piano Sonata 11 in A major, K. 331, I, mm. 1–8	F&G
Mozart	Piano Sonata 13 in B-flat major, K. 333, III, mm. 1–8	F&G manual
Mozart	Piano Sonata 16 in C major, K. 545, III, mm. 1–8	F&G manual
Mozart	Six Variations on an Allegretto, K. Anh. 137, mm. 1–8	F&G manual
Mozart	Piano Sonata 7 in C major, K. 309, I, mm. 1–8	C&G
Mozart	Piano Sonata 13 in B-flat major, K. 333, I, mm. 1–4	F&G
Mozart	7 Variations in D major on “Willem van Nassau,” K. 25, mm. 1–6	SG
Mozart	Twelve Variations on “Ah vous dirai-je, Maman,” K. 265, Var. 1, mm. 23–32	SG, C&G
Mozart	12 Variations in E-flat major on “La belle Françoise,” K. 353, Theme, mm. 1–3	SG
Mozart	Minuet in F for Keyboard, K. 5, mm. 1–4	SG
Mozart	8 Minuets, K. 315, No. 1, Trio, mm. 1–8	SG
Mozart	12 Minuets, K. 103, No. 4, Trio, mm. 15–16	SG
Mozart	12 Minuets, K. 103, No. 3, Trio mm. 7–8,	SG
Mozart	Untitled from the London Sketchbook, K. 15a, No. 1, mm. 12–14	SG
Mozart	9 Variations in C major on “Lison dort,” K. 264, Theme, mm. 5–8	SG
Mozart	12 Minuets, K. 103, No. 12, Trio, mm. 13–16	SG
Mozart	12 Minuets, K. 103, No. 1, Trio, mm. 1–8	SG
Mozart	Piece in F for Keyboard, K. 33B, mm. 7–12	SG
Schubert	Impromptu in B-flat major, Op. 142, No. 3, mm. 1–8	F&G manual
Schubert	Impromptu in G-flat major, Op. 90, No. 3, mm. 1–8	F&G manual
Schubert	Impromptu in A-flat major, Op. 142, No. 2, mm. 1–8	C&G
Schubert	Wanderer’s Nachtlied, Op. 4, No. 3, mm. 1–3	SG

**Table 1.** The musical excerpts contained in SCHENKER41.