

# TOWARDS SCHENKERIAN ANALYSIS BY COMPUTER: A REDUCTIONAL MATRIX

*Alan Marsden*

Lancaster Institute for the Contemporary Arts  
Lancaster University  
Lancaster LA1 4YW  
UK

## ABSTRACT

An approach to the implementation of Schenkerian analysis in a computer program is described. It would appear that a single piece of music has a number of possible analyses related exponentially to the size of the piece. The approach described differs from previous attempts at the computer implementation of Schenkerian analysis by not aiming to build an analysis directly from the notes of the score, but instead to construct a matrix which contains, in different paths through the matrix, all possible analyses. The size of the matrix is related to the square of the size of the piece, and the time-complexity of its generation to the cube. Generation of the matrix is therefore a tractable problem. An analysis can be derived from the matrix in time related to the square of the size of the piece, but the complexity of deriving analyses with particular properties has yet to be investigated.

## 1. INTRODUCTION

Schenkerian analysis is a common and extremely important tool in the academic study of music. The theory has been likened to transformational grammar in language because it describes the structure of music in terms of common, simpler structures underlying their manifestation in elaborated ways on the musical 'surface'. The analogy can be taken further in that a tool which automated the process of Schenkerian analysis could yield similar benefits to automatic parsing of language: segmentation into meaningful units, intelligent editing, discovery of hidden similarities, etc. Elsewhere I have demonstrated how a system similar to the one described here allows the representation of melodic patterns even where they are 'disguised' on the musical surface through processes such as variation [7]. In another project, the same system was used as the basis for software which generated melodies allowing explicit control over their degree of similarity [8].

Research on implementation of Schenkerian theory has previously been conducted by Kassler [4, 5] and Smoliar and others [1, 2, 13]. These did not result in an implementation capable of deriving a complete analysis from a musical score, though Kassler's software can derive an analysis from a 'middleground'.

The theory of Lerdahl and Jackendoff [6], which is indebted to Schenker in some of its reductional formulations, appears at first more immediately

implementable than Schenker's theory because of its formulation as a rule system. However, attempts at implementation of the theory (most recently [3]) have not yet resulted in a complete and useful analytical tool.

This paper reports on a project which aims to re-examine the problem of automatic Schenkerian analysis by computer. It differs fundamentally from earlier approaches by avoiding making decisions about which analytical interpretation to adopt from among alternatives. Instead, all possibilities are gathered into a matrix, reducing the apparent exponential complexity of the full analytical problem to polynomial complexity. A complete analysis can be derived from the matrix, though this aspect of the problem has yet to be fully addressed.

## 2. SCHENKERIAN THEORY

Schenkerian analysis represents a piece of music as a multi-levelled structure. From one perspective each 'higher' level is derived from the preceding level by a process of reduction, leaving only the structurally more important notes. From the other perspective, each 'lower' level builds upon the previous level by a process of elaboration. Only certain kinds of reduction/elaboration are possible, and they must follow certain harmonic and tonal constraints. The analytical process therefore consists essentially of deciding, for each small segment of a piece, and in a recursive process, what kind of elaboration is present, and therefore which notes should be retained at the higher level.

## 3. THE UNDERLYING REPRESENTATION SYSTEM

### 3.1. Notes and Elaborations

The representation system used here is described in detail in [7], but revised and extended as described in [9]. It is simplest to think of as a tree, with two alternating kinds of elements: notes (or rests) and elaborations (though in fact a representation is not necessarily a tree, but rather a directed acyclic graph, because of measures to allow the representation of polyphony). An elaboration has (usually) a single parent note, and generates two or more child notes (or a note and a rest) which occupy the time span of the parent note. This is illustrated in Figure 1. The detail of the

**Figure 1.** Examples of elaborations. The upper stave of each pair shows parents, the lower stave children.

children produced by an elaboration is influenced by the context of metre, harmony and tonality of the parent note. This is the reason for the differences between the two ‘repetition’ elaborations, and the two ‘consonant skip’ elaborations (the first of which could apply in a G-major harmonic context and the second in E minor). The same sequence of notes can often be produced by more than one kind of elaboration, as shown in the last example on the upper pair of staves, where an ‘upper neighbour note’ elaboration (E-D) produces the same sequence of notes as the previous elaboration.

Some elaborations require information from a note immediately following (in the case of neighbour notes and passing notes) or preceding (in the case of suspensions). A link to the appropriate note is recorded with the elaboration (causing a deviation from a simple tree structure). Although the parent note is often copied as the first of the children, this is not the case for some elaborations, such as appoggiaturas and suspensions, as shown in Figure 1, where the parent note is displaced by a new note before it.

### 3.2. Rhythm

Elaborations produce notes at points of time which are, by default, spread within the time span of the parent note as evenly as possible within the given metrical context. For metrical divisions which are not divisible by two, such as the triple division of the second example in Figure 1, the minimally longer time interval(s) is/are, by default, placed at the start of the sequence. Deviations from these default temporal divisions are possible by explicitly specifying a time division with the elaboration, as in some cases in the following figures.

### 3.3. Polyphony

As originally developed, the representation system applied to melodies only. The simplest way to extend it to music with multiple voices is to allow more than one simultaneous tree of elaborations. In the simplest cases,

each voice of a piece of music will be represented by a separate elaboration tree. However, not all cases can be or should be represented so simply. The number of voices is not constant throughout a piece of music, and some melodies have an underlying polyphonic structure.

Two mechanisms handle cases where two voices can become one or one become two. (Both cause the structure of the representation to deviate from a simple tree.) Firstly, a particular kind of elaboration called an ‘unfolding’ (see Figure 1) has two (or more) simultaneous parent notes, and causes these to be ‘unfolded’ into a single sequence of notes in the children. Secondly, a single parent note can have more than one elaboration, and a single child note can be produced by more than one elaboration.

Brief analyses using this system are given in [9].

## 4. THE REDUCTIONAL MATRIX

### 4.1. Complexity of the Analysis Problem

To derive an analysis from a representation giving the pitch and duration of each note on the ‘surface’ of a piece of music would be quite possible. Essentially the problem is to identify, for small sequences of notes, what elaboration(s) could generate the sequence. Sequences will generally be pairs of notes, but in some cases longer. The elaboration might imply certain constraints of metre, harmony and key. It will also imply a certain parent note (or possible parent notes) at the next higher level. The same elaboration-identifying process can be applied recursively to this level, until the highest level consists of just a single chord.

At each step, all the information required to identify the possible applicable elaborations is available locally. The problem, however, is that there is no obvious way of determining locally which among the several possible candidate elaborations, or among the possible candidate segmentations of the surface, will lead to an acceptable final analysis. The constraints implied by simultaneous elaborations might, for example, turn out to be

inconsistent, or the parent notes might turn out not to form a sequence which could be produced by any elaboration. If a complete analysis of the piece does exist, it would eventually be found by exhaustive search through a backtracking process. However, the problem is essentially a combinatorial one, and the number of possible combinations of elaborations is related exponentially to the size of a piece of music.

An analysis method which relied on exhaustive search would therefore be practical only for very small segments of music. While there are some bases for a pruning strategy in work on method for Schenkerian analysis (e.g., [10, 11, 12]) and in the theory of Lerdahl & Jackendoff [6], there seems no guarantee that these will lead to a practical analytical procedure.

This project therefore takes a different route to reduction in complexity. The exponential ‘explosion’ of the original problem arises from two sources. The first is the segmentation of the surface and subsequent levels into sequences of notes to be children of elaborations. The actual number of total possible segments is related to the square of the size of the piece; it is the number of possible *combinations* of different segments which rises exponentially. The second is similarly the *combination* of different elaborations in the tree structure(s). However, the information required to identify possible elaborations is only the sequence of notes and any constraints of harmony and tonality attached to them. Information about the combination of elaborations which led to these notes is irrelevant. There is a limited number of possible pitches, and the number of possible constraints is also limited. Thus the number of possible simultaneous notes-plus-constraints in any possible segment is limited, and therefore the total number of possible elaborations in any analysis of a piece of music is some multiple of the total number of possible segments, i.e., related to the square of the size of that piece.

#### 4.2. Derivation of the Reductional Matrix

Automatic derivation of the set of all possible elaborations for all possible segmentations of a piece is therefore realistic, even for large pieces of music, and software to automate this derivation is currently under development. The procedure is best illustrated by an example (Figure 2), which shows the matrix derived from a miniature contrived piece. (Figure 2 is a simplification from the matrix as represented in software. Information concerning constraints, valid subsets of notes, and elaborations is not shown.)

The first step is to divide the piece into a sequence of segments, each of which consists of a single chord. This produces the bottom row of Figure 2, labelled ‘1’. Each pair of consecutive segments could be joined in an analysis to form a single segment at a higher level by finding elaborations which have children from each of the lower level segments, producing parent notes which make up the segment at the higher level. (Cases where

Figure 2. Illustration of a reductional matrix.

an elaboration has more than two children, such as the ‘passing’ elaboration of Figure 1 with three children, are decomposed to combinations of elaborations with two children, whose parents (except for the highest level) are special kinds of ‘notes’ which stand for a sequence of notes. This introduces additional complexity, but it is not common and always localised.) The next step of deriving the reductional matrix is to find all such pairs of consecutive segments, resulting in the ‘size-two’ segments in Figure 2 labelled ‘2’. The same process can now be applied recursively to find all ‘size-three’ segments, and so on.

The process continues, forming segments of increasing size, until a single segment covers the entire piece. The total number of segments is  $n(n+1)/2$ , where  $n$  is the number of segments at the lowest level. However, the number of ways of forming segments increases with their size (there are two pairs of lower-level segments which can form each size-three segment, and three pairs for each size-four segment, etc.), so the total number of pairs of segments which must be considered is  $n(n+1)(n+2)/6$ . Thus the space requirement for the reductional matrix is related to the square of the size of a piece, whereas the processing time required is related to the cube.

The crucial part of the reduction is to fill these segments with the notes plus elaborations and constraints which can be derived from the notes in the pairs of segments from which they are composed. This is achieved simply by considering what elaborations could apply to each pair of notes between each pair of segments. The possible parent notes are placed in the higher-level segment, together with references to the elaborations which could generate them plus any constraints. Valid subsets of parent notes are formed from subsets of the possible elaborations whose constraints are consistent and which are 'complete' in the sense that every note in the first child segment and every note in the second child segment participates in at least one elaboration. (It is these subsets which form the basis for the discovery of possible elaborations at the next higher level, rather than the full set of possible parent notes which is shown in Figure 2.) In some cases there will be subsets of elaborations with inconsistent constraints, corresponding to places in the analysis where more than one harmony or key is possible. Such cases are shown in Figure 2 with an oblique stroke between the inconsistent subsets. In the fourth segment of level 2, for example, the harmony could be either G major or E minor. In other cases, there will be no subsets of elaborations with consistent constraints, and a segment will remain empty and cannot participate in any complete analysis. This is the case for the second segment of level 2, where there is no combination of elaborations with consistent harmonic constraints which covers both the notes C4 and B4.

#### 4.3. Extraction of an Analysis from the Matrix

A complete analysis can be extracted from the reductional matrix in a top-down manner. First one of the subsets of notes in the top-level segment must be chosen. Associated with this subset are the sets of elaborations and pairs of lower-level segments which produce these parent notes. One set of elaborations and pair of lower level segments is then chosen, and the same process applied recursively to the subsets of notes in those lower-level segments which are the children of the elaborations. When elaborations are chosen which require particular notes in neighbour segments as context, subsets with those notes must be marked for

selection later in the process. It is this which causes the analysis-extraction procedure to have complexity related to  $n^2$ .

This process is not guaranteed to produce an acceptable analysis. It remains to be investigated whether information can be recorded in the matrix-derivation process which will allow an acceptable analysis to be extracted without significantly increasing complexity.

#### 5. REFERENCES

- [1] Frankel, R.E., Rosenschein, S.J. & Smoliar, S.W. A LISP-Based System for the Study of Schenkerian Analysis. *Computers and the Humanities*, 10, pp. 21-32, 1976.
- [2] Frankel, R. E., Rosenschein, S. J., & Smoliar, S. W. Schenker's theory of tonal music—Its explication through computational processes. *International Journal of Man-Machine Studies*, 10, pp. 121–128, 1978.
- [3] Hirata, K. & Aoyagi, T. Computational Music Representation Based on the Generative Theory of Tonal Music and the Deductive Object-Oriented Database. *Computer Music Journal*, 27, pp.73–89, 2003.
- [4] Kassler, M. A Trinity of Essays. PhD thesis, Princeton University, 1967.
- [5] Kassler, M. (1988). APL applied in music theory. *APL Quote Quad*, 18, 209–214.
- [6] Lerdahl, F. & Jackendoff, R. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, Mass., 1983.
- [7] Marsden, A. Representing Melodic Patterns as Networks of Elaborations. *Computers and the Humanities*, 35, pp. 37–54, 2001.
- [8] Marsden, A.. Novagen: A Combination of Eyesweb and an Elaboration-Network Representation for the Generation of Melodies under Gestural Control. *Proceedings of the COST287-ConGAS Symposium on Gesture Interfaces for Multimedia Systems (GIMS)*, Leeds University, pp. 52–57, 2004.
- [9] Marsden, A. Generative Structural Representation of Tonal Music. *Journal of New Music Research*, 34, forthcoming.
- [10] Plum, K.-O. Towards a Methodology for Schenkerian Analysis (translated by W. Drabkin). *Music Analysis*, 7, pp. 143-164, 1988.
- [11] Rothgeb, J. Design as a Key to Structure in Tonal Music. *Journal of Music Theory*, 15, pp. 230–53, 1971.
- [12] Schachter, C. Either/Or. In H. Siegel (ed.), *Schenker Studies*, Cambridge University Press, pp. 165–79, 1990.
- [13] Smoliar, S. W. A Computer Aid for Schenkerian Analysis. *Computer Music Journal*, 4, pp. 41–59, 1980.