Journal of New Music Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/nnmr20

Schenkerian Analysis by Computer: A Proof of Concept

Alan Marsden [a]
[a] Lancaster University, UK

Available online: 09 Nov 2010

To cite this article: Alan Marsden (2010): Schenkerian Analysis by Computer: A Proof of Concept, Journal of New Music Research, 39:3, 269-289

To link to this article: http://dx.doi.org/10.1080/09298215.2010.503898

Journal of New Music Research
2010, Vol. 39, No. 3, pp. 269–289

Schenkerian Analysis by Computer: A Proof of Concept

Alan Marsden

Lancaster University, UK

Abstract

A system for automatically deriving a Schenkerian reduction of an extract of tonal music is described. Schenkerian theory is formalized in a quasi-grammatical manner, expressing a reduction as a binary-tree structure. Computer software which operates in the manner of a chart parser using this grammar has been implemented, capable of deriving a matrix of reduction possibilities, in polynomial time, from a representation of the score. A full reduction of the extract can be discovered by selecting a tree from this matrix. The number of possible valid reductions for even short extracts is found to be extremely large, so criteria are required to distinguish good reductions from bad ones. To find such criteria, themes from five Mozart piano sonatas are analysed and samples of 'good' reductions (defined by reference to pre-existing analyses of these themes) are compared with randomly sampled reductions. Nine criteria are thereby derived, which can be applied in the process of parsing and selecting a reduction. The results are promising, but the process is still too computationally expensive—only extracts of a few bars in length can be reduced—and more extensive testing is required before the system can be properly claimed to perform automatic Schenkerian analysis.

1. Introduction

Since the first flurry of interest in formal grammars and their potential application to music, beginning in the 1960s and most famously manifested in Bernstein's Norton Lectures of 1973 (Bernstein, 1976), some have looked to Schenkerian theory as providing a kind of grammar of music and have sought to implement it in a computer program after the fashion of computational linguistics. A decade or so later, Lerdahl and Jackendoff (1983) presented a similarly reductional theory explicitly grounded in generative linguistics, which has also attracted attempts at computational implementation, most recently and most successfully in the work of Hamanaka, Hirata, and Tojo (2006, 2007).

This paper gives a proof of concept of a mechanism for deriving quasi-Schenkerian reductions automatically from the information in a score. The immediate objective is a computer program which takes as input a representation of information from a score and produces as output a representation of a reduction of that piece which matches a Schenkerian analysis of the same piece. There are several reasons for doing this. One is as a means of investigating the practice of Schenkerian analysis, and of exploring Schenkerian theory. Schenkerian analysis is taught to students, and presented by Schenker (1935) himself, as a set of principles applied in a large number of examples. Knowledge of the principles is clearly not sufficient to make analyses, but on the other hand it is not clear what additional knowledge is gleaned from the tutorial examples. Attempting to write a computer program to make analyses exposes the deficiencies of the principles alone and offers a means for explicit exploration of the additional knowledge required. As such, the project is a music-theoretic exercise, but there are those (I am among them) who believe that Schenkerian analysis also tells us something about the cognition of music. Lerdahl and Jackendoff (1983), for example, in presenting a theory which also expresses musical structure through reduction, explicitly claim to model musical cognition. Validating models of musical cognition, however, is difficult, and one generally has to resort to comparing the output of a model with the

Correspondence: Alan Marsden, Lancaster Institute for the Contemporary Arts, Lancaster University, Bailrigg, Lancaster, LA1 4YW, UK. E-mail: a.marsden@lancaster.ac.uk

DOI: 10.1080/09298215.2010.503898

product of a musical activity, whether it be listening experiments, music analyses or compositions. The cognition-modelling exercise therefore comes to resemble the music-analysis-modelling exercise. Thus while I explicitly do not aim here to model cognition, because I make no attempt to construct a model which plausibly reflects mental processes, I hope that the outcome can contribute to greater understanding of musical cognition.

A second reason for pursuing this project is for its potential contribution to music information retrieval (MIR) tools. A number of studies suggest that such tools can be improved by the incorporation of high-level musical knowledge. Orio and Rodà (2009), for example, observe that using a reductional structure allows for a better definition of similarity between two melodies. It will become clear that an MIR tool using Schenkerian theory is still far away, because the software described here is far too inefficient to make a useful tool, but this at least is the first step towards making Schenkerian theory, applied to more than just melodies, available to MIR.

The outline of the paper is as follows. Section 2 gives an essential summary of Schenkerian reduction and reviews related prior systematic work. Section 3 gives a formalization of quasi-Schenkerian tonal reduction with sufficient precision for computational implementation. Section 4 outlines the problem of the explosion of the solution space and describes a reduction algorithm to cope with this problem. Section 5 describes exploratory work, using short extracts of Mozart piano sonatas and pre-existing analyses of those extracts, to discover a 'goodness' metric by which to select a preferred reduction from among the many possibilities. Section 6 discusses the results with respect to this small sample, which show moderate success, and considers the prospects for further work.

As stated above, this is a proof of concept rather than a claim to have solved the problem of Schenkerian analysis by computer. The mechanism described here will not simply scale up to cover more than a few bars of music. Two substantial steps remain to be taken before the problem of Schenkerian analysis by computer can be claimed to have been properly solved: thorough testing, and development of a mechanism which can be applied to realistic spans of music.

2. Systematic reduction of tonal music

While reduction, in the broad sense of 'information reduction', is a part of any analysis, when applied to music the term generally has the quite specific meaning of deriving from a piece of music a structure of notes which contains the main outline of that music without its more 'ornamental' features. Thus the result of reducing a piece of music is itself a piece of music (at least in the sense that it is a structure of notes) which contains fewer notes than the original. The reduction process can then be applied recursively, producing ever simpler and simpler musical structures, until an irreducible basis is reached. Although the fundamental idea of such reduction is found in many places in music theory, it takes its most developed form in Schenkerian analysis, a technique of music analysis which has its origins in the publications and teaching of Heinrich Schenker (expressed most succinctly in *Der Freie Satz* (Schenker, 1935)).

As mentioned above, Lerdahl and Jackendoff (1983) have proposed a theory which also makes use of the same essential concept of reduction. Their theory is more systematic since it is explicitly expressed in a set of rules, but these are not directly computable on account of ambiguities, incompletenesses and circularity. I choose to implement Schenkerian analysis rather than that of Lerdahl and Jackendoff principally because there is a large quantity of pre-existing Schenkerian analyses which provide potential test material. However, I take the idea of representing reduction in binary trees from Lerdahl and Jackendoff (see below).

2.1 Recursive functions

Earlier work on implementation of Schenkerian theory in computer software has drawn on similarities between Schenkerian theory and the concept of recursion in mathematics and computer science. The earliest publication which relates Schenkerian theory to computing in this manner is by Kassler (1967), which expresses the middle-to-background reductions of Schenkerian theory as a set of operations on matrix representations of pitch structures. Smoliar and others instead represented music in a list structure (the classic vehicle for recursion) and implemented a set of Schenkerian elaboration functions in LISP (Frankel, Rosenschein, & Smoliar, 1976, 1978; Smoliar, 1980). The structure of a piece of music could thereby be expressed as a nested sequence of function calls which, when evaluated, would generate the list structure of notes corresponding to the score.

Others took as their starting point formal grammars rather than computing concepts, but the results are not significantly different. One of the most Schenkerian among these projects was by Baroni and colleagues (Baroni, 1983; Baroni, Dalmonte, & Jacobini, 1992), though it applied only to melodies. Once again, the software generated music rather than analysing it, but one attraction of using formal grammars is the possibility of using a parsing mechanism proven to be effective for a class of grammars as a means of analysis. Interest in musical grammars has therefore often focused on the application of different kinds of grammar to music (Steedman, 1984, 1996; Barbar, Desainte-Catherine, & Miniussi, 1993; Desainte-Catherine & Barbar, 1994; Terrat, 2005). However, such parsing has not often been shown to be effective in analysis of pieces. One notable exception is Pachet (2000). Another is the Schenkerian component of Ebcioglu's (1987, 1988, 1990) chorale-harmonization system which contains a grammar and heuristics for deriving Schenker-like analyses of the melody and bass lines of a chorale.

2.2 Automatic reduction

Frankel et al. (1976, p. 30) were hopeful of extending their LISP-based work to make a system capable of automatic analysis: 'we anticipate that on the basis of our results thus far, we should be able to formulate an experimental grammar to be used for automated analysis.' In a later publication, however, they are more cautious: 'it is questionable whether a program which produces Schenkerian analyses may be designed without a peripheral "world model" of musical perception' (Frankel et al., 1978, p. 134). Two years later Smoliar presented his software explicitly as an aid to the human analyst (Smoliar, 1980) with no implication of future automation.

Kassler (1975, 1977, 1988) has worked steadily on software to derive analyses according to his matrix-manipulation model of Schenkerian theory, resulting in software which is able to derive an analysis from a three-voice middleground, i.e. not directly from the score but from an already partially reduced structure. This is impressive work which deserves to be better known, though unfortunately Kassler himself has not been able to develop it further (personal communication, 5 May 2005).

Musical grammars have often been concerned with chords rather than notes, partly because chords form single sequences of symbols rather than the multiple parallel sequences of notes which constitute most pieces of music. Grammars which deal with notes have tended, like Baroni's, to represent only single lines. Ebcioglu's grammar, for example, handled the bass and melody of a chorale harmonization separately. Furthermore, its author made clear that he did not attempt to faithfully implement Schenker's theory, but rather only aspects of it, principally linear progressions (Ebcioglu, 1987, p. 88). The result was a system reported to be capable of deriving 'good hierarchical voice-leading analyses' of chorale melodies, but not of bass lines (Ebcioglu, 1987, p. 79).

Mavromatis and Brown (2004) also took a grammar-based approach, but their grammar dealt with multi-voice structures rather than one-dimensional sequences of either chord symbols or notes. They demonstrated the theoretical possibility of expressing Schenkerian theory in a context-free grammar, a kind of grammar which has a particularly simple and effective parsing mechanism. While this suggested that the grammar could form the basis for automatic derivation of Schenkerian analyses, this promise has not been fulfilled because the number of re-write rules required is preventatively large (Mavromatis, personal communication, 6 March 2007). Possibly using a context-free grammar merely moves complexity from the parsing process into the grammar itself.

As will become clear below, complexity is the real stumbling-block to progress towards Schenkerian analysis by computer. A simple analysis system leads to many, many possible analyses, of which many are either incorrect or not as good as others. Although it has not been reported, I suspect that it was this complexity which prevented Smoliar and Kassler from achieving their objectives in the 1980s. Ebcioglu overcame the problem to some degree through the use of explicit heuristics to select analysis steps which are more likely to produce good analyses. Gilbert and Conklin (2007) took as their starting point a grammar of melody with similarities to that of Baroni, but made it explicitly probabilistic so that analysis steps were more likely to lead to good analyses. A Hidden Markov Model learned probabilities based on a set of exemplars, resulting in software which was able to derive reductions from melodies with a moderate degree of success.

The software ATTA, by Hamanaka et al. (2006), derived a reduction in the style of Lerdahl and Jackendoff from a melody. The problem of multitudes of possible analyses was here handled through the user adjusting parameters for each melody to arrive at an acceptable reduction (a process reported to take an expert about 10 min for each melody (Hamanaka et al., 2006, p. 271)). A development of the system which uses a feedback loop from higher levels of reduction to adjust parameters automatically has been shown to be able to produce better results than no parameter adjustment, but the results are still quite distant from 'good' analyses (Hamanaka et al., 2007).

The most recent research explicitly directed at computational Schenkerian analysis has been by Kirlin and Utgoff. They describe a representation system and theoretical framework for the implementation of Schenkerian analysis, conceived as search through a space of directed acyclic graphs. Their response to the problem of the size of the search space is essentially to work on 'pre-processing' to guide search. They report software which is able to pick out a candidate *Ursatz* (fundamental structure) from a piece of piano music (Kirlin & Utgoff, 2008). Working from the other end of the problem, Kirlin has recently implemented software which derives a foreground from a score with a notable degree of success (Kirlin, 2009).

In summary, for more than three decades there have been attempts to implement Schenkerian reduction (or similar) in computer software. Kassler's work demonstrated that this was possible for restricted kinds of music (middlegrounds), but, as demonstrated in the work of Mavromatis and Brown, there are serious impediments to effective automatic reduction. The relative success of Ebcioglu, Gilbert and Conklin, and Hamanaka et al. in dealing with melodies, however, suggests that techniques for overcoming these impediments could be found for full musical textures.

3. Formalization of reduction

3.1 Musical primitives

A Schenkerian analysis generally consists of a set of 'graphs' presented in a kind of music notation, at several levels, often with an accompanying textual commentary and sometimes notes and comments written on the actual graphs. Here I will ignore these textual comments and deal only with the music notation of the graphs. Each graph spans the entire length of the piece, and each represents a level of reduction between the 'surface' of the piece (the notes of the score) and the *Ursatz*, which, according to the analysis, is the particular model used in this piece of one of the three forms of fundamental structure which Schenker believed to underlie every piece of proper tonal music.

The graphs most importantly consist of notes, slurs, barlines, clefs and key signatures. Barlines, if present at all, appear only in the lowest level graphs, but the use of bar numbers and the alignment of graphs with each other can imply where the time-points corresponding to barlines occur in higher-level graphs also. Other symbols on the graphs will not be considered here. While an analyst might make reference to factors such as articulation and dynamics in the course of making an analysis, these factors do not have any explicit role in the theory. One factor which is clearly of considerable importance is which instrument plays which note, but its importance seems to have been so self-evident for Schenker that even that factor deserves no explicit role in his theory. Here, I simply side-step the issue by considering only keyboard music where this factor is irrelevant. Thus, for the present purposes, the surface of a piece of music will be considered to be a collection of notes defined by their pitch and rhythm, within a given framework of key and metre.

Pitches are here represented by MIDI numbers. (A more complex representation which distinguishes between enharmonically equivalent pitches such as C♯ and D♭ is not necessary, since in the system described below it is the size of the interval between pitches which is important rather than their spelling.) Pitch classes (used in the representation of harmony) are represented by integers modulo 12. Durations are represented in relative terms only, since Schenkerian theory gives no significance to absolute duration. Music notation is only capable of showing durations with rational ratios, so durations are represented by integers, with 1 corresponding to the greatest common divisor of all the durations used in a piece. A key is represented by its tonic pitch class. (Only major keys are accommodated in the system at present.) A metre is a pattern of beats at two or more levels, represented by a set of durations corresponding to the recurring time interval between beats at the corresponding level. Thus the highest level (corresponding to the strongest beats) gives the duration of a single bar (measure), the next level gives the duration of a half or third of the bar, depending on the metre, and so on. The metre is indicated explicitly in the encoding of a piece. (If this were a model of music cognition, the key and metre would have to be derived from the pattern of pitches and durations rather than being given in the input.)

The surface of a piece of music could then be represented as a collection of notes, each represented by three integers: for the pitch, the duration, and the time at which the note starts relative to the start of the piece. However, for reasons which will become clear later, it is useful to be able to treat a piece as a succession of events each consisting of any number of sounding notes (i.e. a single note or a chord), or none (i.e. a rest), and at which each note might start (or start again) or might be a continuation of a note in the previous event (a tied note). These events will be called 'segments' (corresponding to the 'time spans' of Lerdahl and Jackendoff). (In some music-theoretic texts such a construct is called a 'simultaneity', but sometimes the word is also used for instants which have no duration or an indeterminate duration, so the word 'segment' is used here to avoid any potential confusion.) In this representation, individual notes do not need to have durations: the duration can be attached to the segment and all notes within the segment will have that duration. Furthermore, starting times do not need to be represented, because they are inherent in the sum of the durations of the preceding segments in the sequence. On the other hand, it is now necessary to add to each note a property to indicate whether it is tied to a preceding note or is a newly started note. Furthermore, a single note in a score might have to be represented by several notes across several segments, with a tie on the second and subsequent notes.

The surface of a piece is thus represented as a sequence of segments. A segment has a duration and a set of notes (possibly empty). Each note has a pitch and a state of being tied or not tied to a preceding note. (An illustration is given in Figures 1 and 2.) Note that there is no explicit representation of voices: the notes of one segment follow the notes of another with no implication that any note in the first is in the same voice as any note in the second. While the voices of a piece are generally evident to the listener and to someone reading the score, they are not generally unambiguously represented in the score in keyboard music. Furthermore, there are many cases in Schenkerian analysis where elaborations involve notes from more than one voice, or where one voice at the surface becomes more than one voice at some level of reduction. In general, the analysis system presented here employs no explicit concept of voice, except that restrictions are employed on the pairs of children from elaboration of simultaneous parents comparable to restrictions on crossing voices, and in some situations a particular role is given to the lowest or highest notes of segments.

3.2 Binary trees

Lerdahl and Jackendoff explicitly represent their reductions in tree structures. Schenker, working within a different and earlier intellectual tradition, did not, but the work of Kassler and Smoliar demonstrates that a conversion of his graphs to tree structures is possible (discussed in detail in Marsden, 2005). To simplify, a reductional structure will here be considered to be a *binary* tree. Each node of a reductional tree is a segment (as defined above) (again following Lerdahl and Jackendoff). The duration of a segment which is not a terminal (i.e. not a 'leaf' of the tree) is equal to the sum of the durations of the two segments below it in the tree. In this I follow Komar (1971) who adds treatment of rhythm to Schenkerian theory by regarding elaboration as introducing additional notes within a specific time span. Thus at each branching point a longer (more background) span is divided into shorter (more foreground) spans.

Two aspects of Schenkerian theory do not fit neatly into this simple structure: elaborations which produce sequences of more than two notes and elaborations which depend on context outside the time span being elaborated. Unaccented neighbour notes and anticipations, for example, depend on a following pitch to which they resolve, and can be unrelated in pitch to the preceding note. However, they are related to the preceding note in their timing. The use of a structure based on segments (collections of simultaneous notes), rather than one based on notes, requires that time be privileged over pitch, and so neighbour notes and anticipations are paradoxically represented as deriving from a note unrelated in pitch rather than from the note to which they resolve. An alternative would be to use planar directed acyclic graphs (i.e. DAGs which can be represented in two dimensions with no crossing arcs) instead of trees, allowing notes such as passing notes to be linked to two parents. This kind of representation has been used by Kirlin and Utgoff (2008) and developed in detail by Yust (2006). I used a similar representation myself in previous work (Marsden, 2001) but came to the conclusion that the additional complication of a graph structure over a tree structure did not outweigh the complication of handling context constraints within a tree structure. (See the discussion of 'context notes', Section 3.4, below.) I suspect that ultimately the best course will be to switch between the two kinds of representation (a topic given formal consideration by Yust) according to the task at hand.

As mentioned above, the highest level of a Schenkerian analysis is an *Ursatz*, one of a class of three prototype structures which Schenker believed to underlie every great piece of music. Each of these begins and ends with tonic harmony in root position, has root-position dominant harmony immediately preceding the final tonic, and has a top line (the *Urlinie*) which descends by step from the tonic, fifth or third of the scale to the tonic an octave, fifth or third below respectively. However, it makes for a simpler formal structure if reduction is considered to continue beyond the *Ursatz* until a single segment is reached, making a single tree. Indeed Schenker himself considered the three forms of the *Ursatz* to be derived from the static tonic triad, the 'chord in nature'. As formalized here, therefore, a legitimate analysis will be a binary tree of segments, of which the highest (the 'root') has root-position tonic harmony, and which contains a sequence of segments forming an *Ursatz*. This sequence must span the entire tree, i.e. every other segment in the tree must be either an ancestor or a descendent of a segment in the sequence.

3.3 Atomic elaborations

The notes of segments which are not the highest-level segment (the 'root') can be derived by applying specific 'atomic elaborations' to the notes of the segment above in the tree. Each atomic elaboration refers to a single note in the segment above, which will be called the 'parent', and produces two notes (or a note and a rest), called the 'children', one in each of the segments below. For this proof of concept the simple set of atomic elaborations defined in Table 1 will be used, derived largely pragmatically with the objective of a small set which allows successful reduction of the example materials described below. To establish a set of atomic elaborations with wide validity will require further research.

Every segment is considered to contain a notional rest, even if no rest is notated in the score. Thus there is always a rest available to be a child of a 'shortening' or 'delay' elaboration. This allows configurations such as 'unfoldings', where a single voice adumbrates more than one voice in 'pseudo-polyphony', to arise naturally by multiple 'shortening' and 'delay' elaborations.

In the formal definition given below, the term 'atomic elaboration' is used for elaborations such as those described here which have a single note as parent and a pair of notes (or a note and a rest) as children. The term 'elaboration' is used to refer to a ternary relationship between *segments* whereby a parent segment is elaborated to become a pair of child segments. Such an elaboration must be made up of a valid set of atomic elaborations.

Table 1. Definitions of 'atomic elaborations'. Letters in italics are variables which must take the same value in each column of a row. *Additional conditions also apply to the 'interruption' elaboration: the harmony of the second segment must be V or V7.

Name	parent		1st child		2nd child		pre-context	post-context	consonant pitches	inheritance
type	*pitch*	*tie*	*pitch*	*tie*	*pitch*	*tie*	*pitch*	*pitch*	*set of pitch class*	*inheritance*
consonant skip 1	x	t	x	t	y	no	$\emptyset$	$\emptyset$	$x+y$	either
consonant skip 2	y	no	x	t	y	no	$\emptyset$	$\emptyset$	$x+y$	either
appoggiatura	y	no	$x=y\pm 1$ or $y\pm 2$	no	y	no	$\emptyset$	$\emptyset$	y	2nd
anticipation	x	t	x	t	y	no	$\emptyset$	y	x	1st
neighbour note	x	t	x	t	$y=z\pm 1$ or $z\pm 2$	no	$\emptyset$	z	x	1st
suspension	y	no	$x=y\pm 1$ or $y\pm 2$	yes	y	no	x	$\emptyset$	y	2nd
repetition	x	t	x	t	x	no	$\emptyset$	$\emptyset$	x	either
hold	x	t	x	t	x	yes	$\emptyset$	$\emptyset$	x	either
shortening	x	t	x	t	rest	no	$\emptyset$	$\emptyset$	x	either
delay	x	no	rest	no	x	no	$\emptyset$	$\emptyset$	x	either
interruption*	x	t	x	t	y	no	$\emptyset$	$\emptyset$	I	IV

There seems no reason to insist that a parent note cannot be elaborated by more than one atomic elaboration. On the contrary, there appear to be instances in actual analyses where a single note is elaborated simultaneously in more than one way. Similarly, there seems no reason to insist that a single note cannot be the child of more than one atomic elaboration, but it cannot be the subsidiary note in one and the main note (i.e. having the same pitch as the parent) in the other.

3.4 Context notes

As noted above, the atomic elaborations 'anticipation' and 'neighbour note' can exist only in the presence of an appropriate following note. Suspensions require a particular preceding note. These will be referred to as 'post-context' and 'pre-context' notes respectively. The requirements, if any, for each atomic elaboration are specified in Table 1.

Note that the context notes can be found either in surface segments or in segments at a higher level, but making two segments children of an elaboration places constraints on where the elaborations of those segments may find their context notes: it must be possible to progressively elaborate from the root segment by applying one elaboration at a time to any currently non-elaborated segment in a sequence such that at every step the required pre-context notes of every atomic elaboration making up that elaboration are present in the immediately preceding unelaborated segment and the required post-context notes are present in the immediately following unelaborated segment. This constraint can be embodied in the concept of a 'valid sequence' of segments. This is a pair of segments which do not violate these context constraints, and therefore can validly become a pair of children of a parent segment. All pairs of surface segments make a valid sequence. It is possible to deduce from this and from the elaborations which relate parents to children whether or not any other pair of segments in a partially formed analysis form a valid sequence.

Often, in computational parsing or reduction, context-sensitivity is removed if possible because of the complications it brings. Both Mavromatis and Brown (2004) and Gilbert and Conklin (2007), for example, use a representation based on pitch intervals rather than notes to allow their grammars to be context-free. Such measures generally bring new complications, however. Removing an appoggiatura, for example, changes the pitch interval between one pair of notes but the time interval between another. If context dependencies are only local (as these ones are), then while it is generally possible to transform the representation of the material so that the system becomes formally context-free, it is not always simpler to do so than to use an efficient mechanism for checking the local context.

The serious issues over context-sensitivity come when a distant context becomes important. These issues are given some discussion in the conclusion.

3.5 Harmonic constraints and combinations of elaborations

Atomic elaborations generally require certain notes to be consonant. (The others need not be dissonant, but often are, or at least are dissonant with respect to the time span of the elaborated note.) The consonances implied by the atomic elaborations which make up the elaboration of a segment must be consistent: the pitch classes of the consonant notes must be members of an acceptable harmony. Furthermore, this harmony must be inherited by at least one of the child segments (i.e. no elaboration of that child can require a pitch class to be consonant which is not a member of the same set). (Schenker does not state this requirement for inheritance of harmony in his theory, but it is evident from his analyses. It is explicit in the theory of Lerdahl and Jackendoff (1983, pp. 152–155).)

In the formalization and implementation below, these constraints are accommodated through two devices. Firstly, every segment has associated with it a 'harmony', which is a set of the pitch classes which must be consonant at that segment. The set must be equal to or a subset of an acceptable triad or seventh chord. Thus the 'harmony' might actually be ambiguous (i.e. be a subset of more than one acceptable harmony). Initially every surface segment has the empty set as its harmony. Note that not all notes of a segment need to be consonant with the harmony, except in the case of the top-level segment and pairs of child segments which make a harmonic progression (i.e. the union of their harmonies does not form an acceptable harmony).

Secondly, each atomic elaboration has associated with it an 'inheritance' which specifies which child must inherit the parent harmony: '1st', '2nd' or 'either'. The inheritances of atomic elaborations are combined in the elaboration of a segment, which will have the inheritance 'both' if it consists of atomic elaborations with both '1st' and '2nd' inheritances.

Often the harmony of a segment can be inferred from the pitches of its notes, but it is not safe to always do so. For example, consider the sequence of the pitches F5, Bb4, Eb5, A4, with the pitch D4 sounding below both F5 and Bb4 and F4 sounding below Eb4 and A4. (Something like this actually occurs in the melody of the theme from Mozart's K.333 used below.) A possible reduction is to take the first and second pairs of upper notes as consonant skips, selecting F5 from the first pair and A4 from the second, resulting in the sequence F5, A4, once again with D4, F4 sounding below. If we have not recorded that Bb must be consonant in the segment containing F5, and Eb in the segment containing A4, it would be possible at this stage to treat the resulting sequence of notes F5, A4 as also forming a consonant skip within the erroneous harmony of D minor.

3.6 Summary

To summarize, the formal theory presented here uses the following concepts:

- **pitch**: integer
- **duration**: integer
- **pitch class**: integer modulo 12
- **tie**: boolean
- **note**: pitch × tie
- **rest**
- **harmony**: set of pitch class
- **inheritance**: '1st', '2nd', 'either', 'both' or 'IV'
- **segment**: (set of note + rest) × duration × harmony
- **surface**: sequence of segments
- **analysis**: binary tree of segments

and the following relations ('_' stands for any unspecified argument, ∅ for the empty set, I for the tonic triad, and V7 for the dominant seventh):

- **class(*p*, *pc*)**: pitch × pitch class
  - $pc = p$ modulo 12; the operator **class(*p*)** can be defined to yield the pitch class *pc* such that class (*p*, *pc*).
- **consonant(*p*, *h*)**: pitch × harmony
  - consonant(class(p) ∪ h).
- **consonant(*h*)**: harmony
  - The harmony must be equal to or a subset of one of a certain set of harmonies, defined by the tonal language. In the system described here this set is all major, minor, diminished and augmented triads, and all dominant, minor, diminished and half-diminished sevenths.
- **atomic_elaboration(*n*, *n1*, *n2*, *p1*, *p2*, *h*, *i*)**: note × (note ∪ rest) × (note ∪ rest) × (pitch ∪ ∅) × (pitch ∪ ∅) × harmony × inheritance
  - The arguments must conform to one of the elaboration patterns defined by the tonal language, such as those defined in Table 1. The arguments are, respectively, the parent note, the first child note, the second child note, the pre-context pitch (if any) required to be present in some preceding segment, the post-context pitch (if any) required to be present in some following segment, the set of pitch classes which must be consonant (the harmony), and the inheritance.
- **elaboration(*s*, *s1*, *s2*, *E*)**: segment × segment × segment × set of atomic_elaboration
  - Let *h*, *h1*, *h2* be the harmonies of *s*, *s*1, *s2* respectively.
  - All the following must be true:
    - valid_sequence(*s1*, *s2*).
    - consonant(*h*).
    - duration of s = duration of $s1$ + duration of $s2$.
    - $\{ n: (n, \_, \_, \_, \_, \_, \_) \in E \}$ = notes of *s*.
    - $\{ n1: (\_, n1, \_, \_, \_, \_, \_) \in E \}$ – rest = notes of *s1*.
    - $\{ n2: (\_, \_, n2, \_, \_, \_, \_) \in E \}$ – rest = notes of *s2*.
    - Let $P1 = \{ p1: (\_, \_, \_, p1, \_, \_, \_) \in E \}$.
      - If $P1 \neq \emptyset$, there must exist a segment *s3* such that valid_sequence(*s3*, *s*) and $\forall\, p1 \in P1$ ($\exists$ *n3* component of *s3* ($p1$ = pitch of *n3*)).
    - Let $P2 = \{ p2: (\_, \_, \_, \_, p2, \_, \_) \in E \}$.

- ■ If $P2 \neq \emptyset$, there must exist a segment *s4* such that valid_sequence(*s*, *s4*) and $\forall\, p2 \in P2$ ($\exists\, n4$ component of *s4* ($p2 =$ pitch of *n4* and *n4* is not tied)).
- □ Let $h4 = \cup \{\, h3$: (_, _, _, _, _, $h3$, _) $\in E \,\}$ and $I = \{\, i$: (_, _, _, _, _, _, i) $\in E \,\}$.
  - ■ Either $(h1 \cup h2 \cup h4) \subseteq h$
  - ■ or let $h1a = h1 \cup$ {class(pitch of *n1*): *n1* member of *s1*} and $h2a = h2 \cup$ {class(pitch of *n2*): *n2* member of *s2*}.
    - ♦ consonant(*h1a*) and consonant(*h2a*) and
    - ♦ either $I \subseteq$ {1st, either} and $(h1a \cup h4) \subseteq h$
    - ♦ or $I \subseteq$ {2nd, either} and $(h2a \cup h4) \subseteq h$
    - ♦ or $I =$ IV and $h1a \subseteq$ I and $h2a \subseteq$ V7 and $h =$ I.

- **valid_sequence(*s1*, *s2*)**: segment × segment
  - ○ Either *s2* follows *s1* in the surface
  - ○ or let *E1* be a set of atomic elaborations such that elaboration(*s1*, _, *s12*, *E1*), and
    - □ valid_sequence(*s12*, *s2*) and
    - □ let $P2 = \{\, p2$: (_, _, _, _, $p2$, _, _) $\in E1 \}$.
      - ■ Either $P2 = \emptyset$ or $\forall\, p2 \in P2$ ($\exists\, n2$ component of *s2* ($p2 =$ pitch of *n2*)).
  - ○ or let *E2* be a set of atomic elaborations such that elaboration(*s2*, *s21*, _, *E2*), and
    - □ valid_sequence(*s1*, *s21*) and
    - □ let $P1 = \{\, p1$: (_, _, _, $p1$, _, _, _) $\in E2 \}$.
      - ■ Either $P1 = \{\emptyset\}$ or enumerate $\forall\, p1 \in P1$ ($\exists\, n1$ component of *s1* ($p1 =$ pitch of n1)).

As noted above, formally every segment contains a rest, even if no rest is notated in the score at the corresponding point. (The only place where a rest must be notated is where a segment contains no notes.) The piece to be analysed must be represented as a sequence of segments whose harmonies are all $\emptyset$ (the empty set). To derive a reduction, one needs to find an analysis (a tree of segments) such that the sequence of leaves of the tree is equal to the surface to be analysed, and for every segment *s* of the analysis and its children *s1* and *s2*, a valid relation elaboration(*s*, *s1*, *s2*, _) holds.

3.7 Example of formal reduction

The analysis of an example taken from the end of the French folk song *Au claire de la lune* is illustrated in Figure 1. As described above, the piece must first be split into segments, as in the line labelled 'Surface', so the minim (half note) D3 is divided into two crotchets (quarter notes), the second tied to the first. Figure 2 gives the same information (but this time containing information on harmonies) in a numerical form.

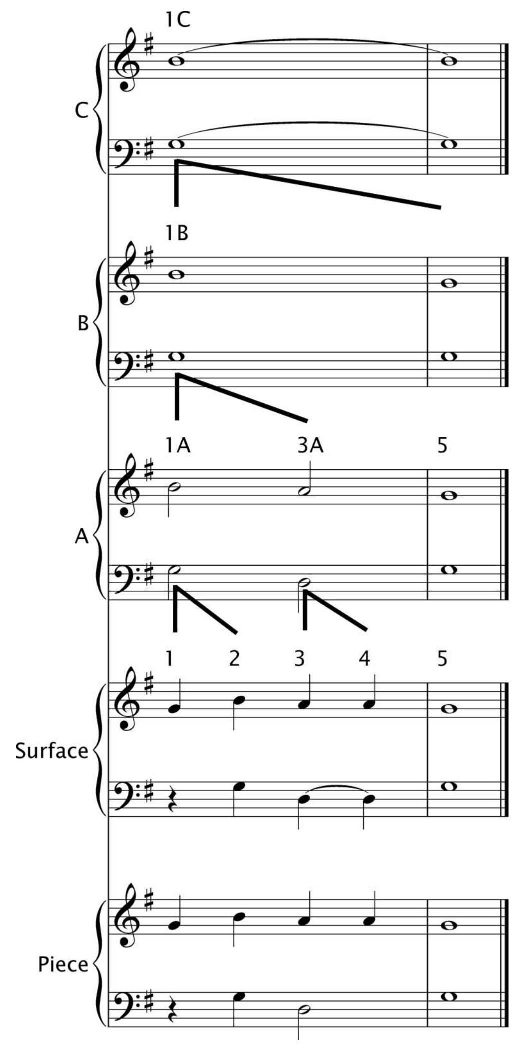

Fig. 1. Final bars of *Au claire de la lune*.

The first stage of reduction is to derive segment 1A from segments 1 and 2. These two segments form a valid sequence, since both are surface segments. There is just one note in segment 1, but every segment also includes a notional rest, so two atomic elaborations can form segment 1A from segments 1 and 2:

- consonant_skip_2(<71, no>, <67, no>, <71, no>, $\emptyset$, $\emptyset$, {7, 11}, either)
- delay(<55, no>, rest, <55, no>, $\emptyset$, $\emptyset$, {7}, either).

The combined inheritance of these two atomic elaborations is 'either', so the harmony of segment 1A must be a superset of the harmony of either segment 1 or segment 2, and of the union of the required harmony of the atomic elaborations, i.e. {7, 11}. The harmony of segment 1A is therefore {7, 11} (i.e. pitch classes G and B must be consonant), which is a valid harmony. The duration of segment 1A is the sum of the durations of segments 1 and 2, i.e. 2.

```
Surface:
[1, {<67, no>}, ∅],
[1, {<71, no>, <55, no>}, ∅],
[1, {<69, no>, <50, no>}, ∅],
[1, {<69, no>, <50, yes>}, ∅],
[4, {<67, no>, <55, no>}, ∅]

Reduction:
[8, {<71, no>, <55, no>}, {2, 7, 11}]
    [4, {<71, no>, <55, no>}, {2, 7, 11}]
        [2, {<71, no>, <55, no>}, {7, 11}]
            [1, {<67, no>}, ∅]
            [1, {<71, no>, <55, no>}, ∅]
            {consonant skip 2, delay}
        [2, {<69, no>, <50, no>}, {2, 9}]
            [1, {<69, no>, <50, no>}, ∅]
            [1, {<69, no>, <50, yes>}, ∅]
            {repetition, hold}
        {neighbour note, consonant skip 1}
    [4, {<67, no>, <55, no>}, ∅]
    {consonant skip 1, repetition}
```

Fig. 2. The surface and reduction of Figure 1 in numerical notation. Each segment is notated as [duration, notes, harmony], and each note as <pitch, tied>.

Segment 3A similarly derives from segments 3 and 4 by the following two atomic elaborations:

- repetition(<69, no>, <69, no>, <69, no>, ∅, ∅, 9, either)
- hold(<50, no>, <50, no>, <50, yes>, ∅, ∅, {2}, either).

The harmony of segment 3A is {2, 9} (i.e. pitch classes D and A must be consonant).

Segments 1A and 3A make a valid sequence, and can be reduced to segment 1B by the following pair of atomic elaborations:

- neighbour_note(<71, no>, <71, no>, <69, no>, ∅, 67, {11}, 1st)
- consonant_skip_1(<55, no>, <55, no>, <50, no>, ∅, ∅, {2, 7}, either).

The harmony of segment 1B must therefore be the union of the required harmony of these two atomic elaborations, and of the harmony of segment 1A {7, 11} by virtue of the '1st' inheritance of the first atomic elaboration. Its harmony is therefore {2, 7, 11} (i.e. pitch classes G, B and D must be consonant).

One of the atomic elaborations in the reduction which formed segment 1B has a required post-context—pitch 67 (G4)—but this does not prevent segments 1B and 5 from forming a valid sequence since there are no competing pre- or post-context requirements and that pitch is found in segment 5. Thus segment 1C can then be derived from 1B and 5 by the following two atomic elaborations:

- consonant_skip_1(<71, no>, <71, no>, <67, no>, ∅, ∅, {7, 11}, either)
- repetition(<55, no>, <55, no>, <55, no>, ∅, ∅, {7}, either).

The segments at level A form an *Ursatz*: the three segments have top pitches in the sequence 71, 69, 67 (B4, A4, G4) forming a step-wise descent to the tonic. The first and last segments have tonic harmony and the tonic pitch in the bass. (In fact the harmony of the last segment is the empty set, which means that strictly any pitch can be consonant, but this admits the possibility of tonic harmony, which is all that is required.) This reduction therefore forms a valid analysis of this tiny piece.

Note that other sets of atomic elaborations can produce the same reduction, and there are other possible valid reductions of this piece. Music theorists admit the possibility that there can be more than one valid Schenkerian analysis of a piece of music. A great deal of the literature discusses alternative analyses, with a strong implication that some analyses are better than other valid but weaker analyses. As will become clear below, a major difficulty in the implementation of Schenkerian analysis is to choose between many alternative analyses.

4. A reduction algorithm

4.1 Deriving a reduction

From inspection of Table 1, it is clear that, given any pair of notes *n1* and *n2*, it is possible to determine the possible parent notes and elaborations which could generate this pair of notes, i.e. to find all members of the relation atomic_elaboration(*n*, *n1*, *n2*, *p1*, *p2*, *h*, *i*) which have *n1* and *n2* as arguments. Using this information, for any pair of segments *s1* and *s2*, it is therefore possible to determine all possible parent segments *s* such that elaboration(*s*, *s1*, *s2*, _). A procedure for this is as follows. First we must ensure that *s1* and *s2* form a valid sequence. We can determine whether or not this is the case from a record of the intermediate segments and pre- and post-context requirements of elaborations by which *s1* and *s2* are derived from the surface. Then, taking each possible pair of notes and rests from *s1* and *s2*, and each combination of the atomic elaborations which can apply to these pairs, we can determine valid combinations of atomic elaborations by taking the union of their harmonies and their inheritances and ensuring that these are valid harmonies, and we can form the resulting segments from the set of resultant parent notes and the resultant harmonies.

A naive reduction procedure, then, might select a pair of as-yet unreduced segments and determine a set of atomic elaborations and a parent segment which could produce this pair as children. Then it would recursively select another pair of unreduced segments, possibly including the newly created parent segment, and determine its set of atomic elaborations and parent. If for any chosen pair a set of atomic elaborations and parent segment cannot be found, the procedure could try another pair instead. If all possible pairs have been tried, the procedure could backtrack, undoing the last reduction, and try a different reduction, either for the same pair or for a different pair. At each step, except one involving backtracking, the number of segments still to be reduced is one less. Provided the backtracking procedure tries every possibility and does not try a possibility which has been tried before, the procedure will eventually find a complete reduction if one exists, resulting in just a single segment at the highest level.

This is essentially the same argument as used by Kassler (1976) to demonstrate that automatic Schenkerian reduction is a logical possibility. However, there is no guarantee that an analysis will be found in a reasonable time, and the atomic elaborations defined here can produce analyses which most analysts would regard as unacceptable. If the 'rules' of Schenkerian analysis are anything like those set out above, they seem to be too loose to determine acceptable analyses by themselves alone. Temperley (2007, pp. 175–176) makes a similar point based on some initial experiments. Other factors must come into play to distinguish acceptable from unacceptable analyses or, better, good from bad.

4.2 Coping with complexity

As pointed out in Marsden (2007), the size of the 'search space' in which a Schenkerian analysis must be found (according to the definitions above) grows factorially with the length of music to be analysed. It is not an easy matter to design an algorithm which will find a good analysis in this space within a reasonable amount of time and using reasonable computing resources.

The technique used here is a kind of 'dynamic programming', and essentially uses the CKY (also called CYK) algorithm (Jurafsky & Martin, 2009, pp. 470–477) to act as a kind of chart parser (Jurafsky & Martin, 2009, pp. 482–484). This constructs a 'chart' of partial parses, like a dynamic-programming array, on the way to making a complete analysis. An empty triangular matrix of 'cells' is formed, with the segments of the surface in the lowest row. Counting upwards, the second row of cells will hold segments which will each span two surface segments, the third row segments spanning three surface segments, etc., until the highest cell contains segments which span the entire piece. This array is filled from the bottom right with all the possible segments which result from reduction of each of the pairs of segments which cover the span of each cell. Above the first row, single cells therefore generally contain more than one segment, each corresponding to different ways the segments below may be validly reduced. The advantage of the CKY algorithm is that it is of polynomial complexity (i.e. much less computationally demanding than the factorial complexity of a backtracking approach). In this case, the complexity is between $O(n^3)$ and $O(n^5)$ in time and $O(n^2)$ and $O(n^4)$ in space, depending on what is regarded as constant and what as dependent on the length of the extract to be analysed.

Once the matrix is filled, complete analyses can be extracted from the matrix by selecting one of the top-level segments and then recursively selecting pairs of possible child segments until the surface is reached. The context-note requirements of some elaborations (see Section 3.4 above) mean that unconstrained selection of child segments can result in an invalid tree, so backtracking can be required. In the worst case, therefore, the computational complexity of this part of the algorithm is of factorial order. (As is common for chart-parsing techniques, the complexity is shifted to a different part of the problem.) A best-first search procedure is used to reduce the likelihood of this worst case arising, but still for some longer extracts finding the putatively 'best' analysis takes a prohibitively large amount of time or space even though the chart can be derived in a reasonable length of time.

As the chart is filled, three additional pieces of information are attached to derived segments.

1. 'Goodness' score. This allows later selection from the chart of a best-scoring analysis. Section 5 describes a first attempt to establish a suitable metric for goodness.
2. *Ursatz* possibilities. This gives the parts of an *Ursatz* which a segment can supply. This allows segments which cannot be part of an analysis with an *Ursatz* to be deleted from the chart, and allows selection of top-level segments and subsequent children to be restricted to only those which yield an analysis containing a particular *Ursatz*.
3. The multiplicity of sub-trees. This is an upper bound on the number of different possible sub-trees there are below the segment concerned. This information is used in deriving a 'fair' sample of possible analyses and in estimating the total number of possible analyses. (See Section 5.3 below.)

4.3 Limits

Initial results showed that unrestricted filling of the matrix of cells was practical only for extracts of music just a handful of segments long. Thus certain limits have been imposed to rule out reductions believed to be almost always 'bad' and to keep the processing requirements of time and space within practical bounds, as follows:

1. Segments are limited to no more than four simultaneous notes on the grounds that Schenkerian graphs usually show no more than four simultaneous lines.
2. 'Complex' harmonies (i.e. seventh harmonies other than the dominant seventh) are permitted only in contexts where there is no possible reduction which results in a simple harmony.
3. Reduction of very short with very long segments is restricted, especially if the short segment comes first. If the short segment comes first, it must have no less than a third of the duration of the longer one. If the short segment comes second, it must have no less than an eighth of the duration of the longer one.
4. The complexity of division of a time span is limited. If the ratio between two child segments is expressed as a rational number in its simplest form, the sum of the numerator and denominator must not exceed nine.
5. There is a limit on the syncopation inherent in derived segments, i.e. in segments during which a beat occurs which is stronger than the beat at the beginning of the segment. This beat can be no more than one level stronger. In a 4/4 metre, for example, to have a segment continue into the next bar is acceptable if it starts on the third beat, but not if the segment starts on the fourth beat.
6. Pairings of notes between segments analogous to crossing voices are not permitted. If note *n1* in the first segment is paired in an atomic elaboration with note *n2* in the second, then no note below (above) *n1* can be paired with a note above (below) *n2*. A similar restriction prevents crossing over a note paired with a rest.
7. Pairings of notes analogous to splitting or joining voices are restricted. A note can have no more than two sibling notes in the other segment, or just one if the sibling is a rest.

4.4 Example of automatic reduction

The reduction procedure is illustrated in Figure 3. The music here is the same as in Figures 1 and 2. The bottom row **S** of the charts **a–f** represents the segments of the

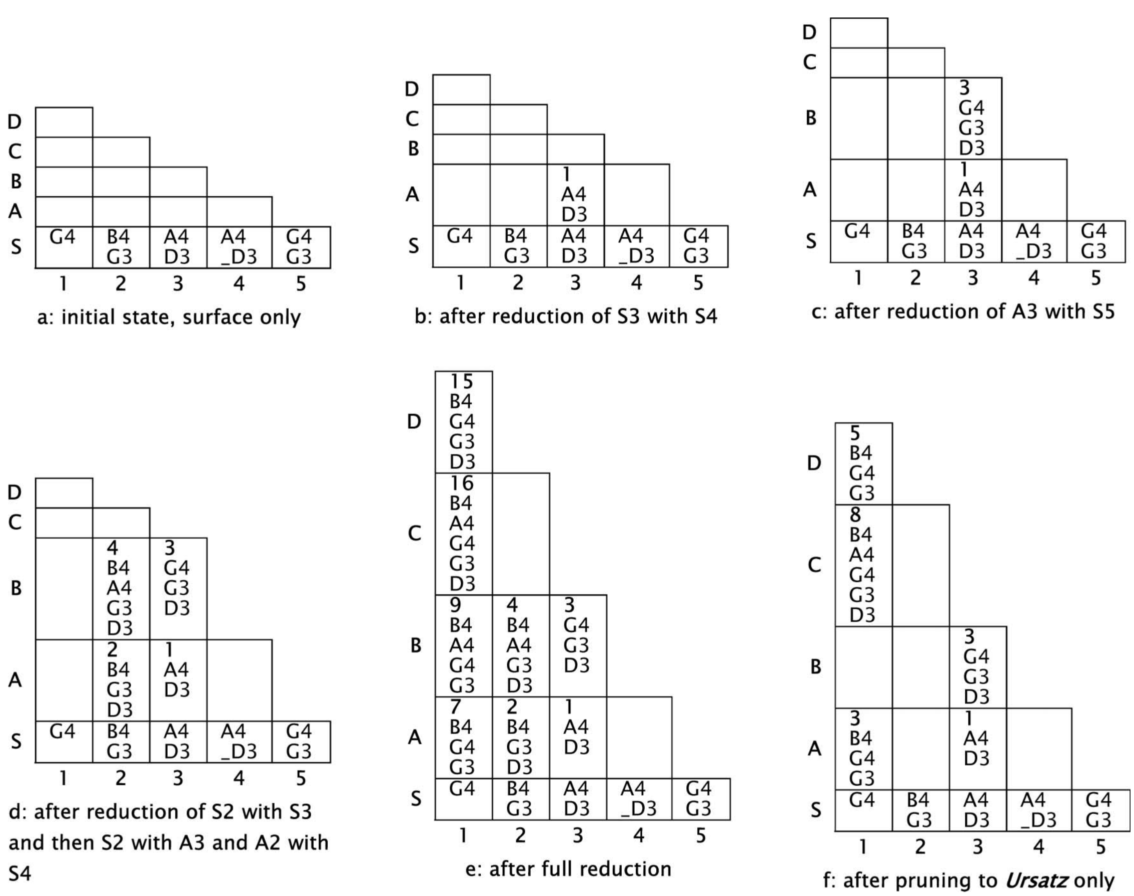

Fig. 3. Illustration of reduction procedure.

surface of the piece. Chart **a** represents the initial state, when only the surface segments exist. The first step is to reduce the rightmost pair of segments **S4** and **S5**, but the resulting segments would exceed the syncopation limit since they would start on the fourth beat and continue into the following bar. Cell **A4** is thus left empty.

At the next step, the possible segments resulting from reducing **S3** with **S4** are placed in cell **A3**. Only one segment is possible: one consisting of the notes A4 and D3, as shown in chart **b**. Then the segments for cell **B3**, which covers the last three segments of the surface, are derived. The segments in this cell could be derived from two pairs of cells: cell **S3** with **A4** and **A3** with **S5**. However, **A4** is empty, so just the one segment in **A3** is reduced with the segment in **S5**, producing three possible segments: G4/G3/D3, G4/G3 and G4/D3. There is not space in the figure to represent these segments individually, so the figure gives in cell B3 of chart **c** the count of number of segments (3) plus the constituent pitches of all segments.

In a similar manner, cells **A2** and **B2** are filled with two and four segments respectively, to produce the chart **d**. Cell **C2** remains empty because its segments would once again exceed the syncopation limit. Finally the cells in column **1** are filled to produce chart **e**. Chart **f** shows the result of pruning the resulting set of segments to leave only those which can be part of analyses which contain an *Ursatz*. From the resulting chart **f**, it is possible to select a number of valid analyses, including the one shown in Figures 1 and 2.

5. Tests and a 'goodness' metric

5.1 Software and hardware

The reduction procedure described above has been implemented in software, using the programming language Java, version 1.6. Example software is available for download at http://lancs.ac.uk/staff/marsdena/schenker. A demonstration applet is also provided. The results reported below arose from using this software on a computer with 4 GB of RAM and an Intel Core2 Duo processor running at 2.66 GHz with the Ubuntu operating system 8.04.

5.2 Materials

The materials used in this investigation are four themes from movements in rondo form from piano sonatas by Mozart (the last movements of K.545, K.333, K.570, and K.494) and two themes from variation movements (the first movement of K.331 and the last of K.284). All these themes form small but complete and self-contained musical units: in Schenkerian terms, they contain an *Ursatz*. For all of these themes, pre-existing analyses were available: in the case of the rondo themes in model answers I have used in teaching, plus a model answer for the K.333 theme from Forte and Gilbert (1982b, p. 42); in the case of the theme from K.331 the analysis by Forte and Gilbert (1982a, p. 134); and in the case of K.284 an analysis by my colleague Neil Boynton. These analyses form a 'ground truth' sufficiently sound for this initial investigation.

Short as they are, only parts of these six themes have been used. The extracts analysed are taken from, or based on, just the second phrase of each of the themes. (All the themes have a first phrase which ends on the dominant and a second which starts like the first phrase but ends on the tonic.) The resulting six phrases are given in Figure 4. The example from K.570 is not exactly the second phrase of the Mozart theme; it omits the anacrusis and a grace note and replaces the penultimate note of the melody line. In the original this note is D5, a note which music theory would call an 'échappée' because it is dissonant with the dominant harmony but does not resolve down to the consonant C5. (Other formulations would call this an 'incomplete neighbour note'.) This kind of elaboration was not included in the set of possible atomic elaborations (Table 1) because it would generate many additional possible reductions. As an expedient to allow some results to be obtained, this note has been replaced by the Bb4 found in example 3, which is instead an 'anticipation'. Similarly, the example from K.284 omits the anacrusis of two crotchet (quarter-note) A4s, and a middle-voice A3 is missed out in the last bar to ensure that no segment contains more than four notes.

Outline reductions were derived from the 'ground-truth' analyses of each of the themes. As an example, Figure 5 shows the 'ground truth' outline reduction derived from Forte and Gilbert's (1982b, p. 42) analysis of the K.333 theme.

5.3 Method

The research method, in outline, was as follows:

1. The reduction chart was derived for each example theme, following the procedure described in Section 4.
2. A sample of 1000 analyses was selected from the chart, without regard to whether or not they contained an *Ursatz*.
3. A second sample of 1000 analyses was selected, this time selecting only analyses which contain an *Ursatz*.
4. A third sample of 1000 analyses was selected, this time selecting only analyses which both contain an *Ursatz* and conform to a 'ground-truth' reduction. A sample was taken for each 'ground-truth' reduction. These analyses were taken to constitute 'good' analyses, since they conform to the ground truth.

Fig. 4. Six Mozart themes (some slightly modified).

5. A 'preferred' analysis was selected, containing an *Ursatz* and conforming to a 'ground-truth' reduction, which yielded the apparent best analysis, according to my experience of Schenkerian analysis and my musical judgement. Figure 6 shows the preferred analysis for the K.333 theme.
6. The results of steps 2–5 were measured according to a set of candidate goodness measures, and the means and standard deviations computed for each set.
7. A candidate 'goodness' metric was derived from those measures which distinguished the preferred analyses (step 5) and the good analyses (step 4) from the sample analyses with *Ursatz* (step 3).
8. To test the 'goodness' metric, the analysis with the highest score according to this metric was selected from the reduction chart for each theme. If the procedure and goodness metric were correct, each selected analysis should be similar to a 'ground-truth' reduction and appear reasonable on the basis of musical judgement.
9. As a further test of the 'goodness' metric, step 8 was repeated with pruning. The reduction charts were made anew, but at each reduction step, only the n best-scoring segments were kept, with n, the 'pruning limit', ranging from 1 to 100. If the goodness metric is effective, a good analysis should result from a low pruning index in each case.

At steps 2 to 4, samples were derived automatically by a random procedure designed to ensure, so far as possible,

Fig. 5. Outline reduction of the K.333 theme derived from Forte and Gilbert (1982b, p. 42).

Fig. 6. Preferred analysis of the K.333 theme.

that the selected analyses were a representative sample of the population of possible analyses in each case. The main aspect of this was to bias selection of children according to the estimated multiplicity of possible subtrees below those children.

5.4 Reduction results

Table 2 gives the estimated population sizes, the computation times and other data for each of the six example themes. The theme of K.545 is somewhat unusual. If no complete reduction of a theme is possible, the software produces an empty chart. Sometimes reduction is possible, but there is no *Ursatz*, in which case the chart is emptied when it is pruned to analyses with *Ursatz* only. In the case of K.545, the software does find reductions with an *Ursatz*, but this is a 3-2-1 *Ursatz* with its initial tonic based on the bass C in the third bar. Both my own analysis and that of Forte and Gilbert (1982b, p. 44) find a 5-4-3-2-1 *Ursatz*, with an implied bass C at the beginning. However, the software has no way of deriving implied notes. Thus none of the analyses for this theme conformed to a ground truth, so the sets at steps 4 and 5 were empty and this theme was excluded from further testing. (Something about this theme has also caused inaccurate estimates of size, since it is impossible for there to be more analyses with *Ursatz* than the total, as suggested by the results in Table 2.)

The striking thing about the figures in Table 2 is how many possible analyses there appear to be for even short and simple themes. As Temperley (2007, p. 176) suggests, the rules of Schenkerian analysis seem to be much too loose to specify an analysis with any degree of usefulness. Clearly other factors are at work in Schenkerian analysis, which are probed in the following sections.

5.5 Candidate goodness measures

Schenkerian analysts have long recognized that alternative analyses of the same piece are possible, and the literature does include some discussion of the factors used to decide which is best among alternative analyses. However, there is no accepted set of principles which distinguish a good analysis from a bad one. Candidate goodness measures could be found in the 'indices' proposed by Plum (1988), the preference rules of Lerdahl and Jackendoff (1983), Larson's 'musical forces' (Larson & Vanhandel, 2005), and Schenkerian instructional literature (e.g. Forte & Gilbert, 1982a; Cadwallader & Gagné 2007; Pankhurst, 2008). However, for this initial proof of concept, measures have been designed on the basis of what is simple to compute and is presumed to be potentially fruitful. In principle, the same procedures could be applied to test measures derived from these sources.

The fourteen measures used were as follows:

1. **duration ratio**: the duration of the longer child divided by the duration of the shorter.
2. **short-long**: 1 if the duration of the first child is less than the duration of the second; 0 otherwise.
3. **syncopations**: if a beat stronger than the beat at the start of the segment occurs before the end of the segment, the difference in level of strength between the strongest such beat and the beat at the start of the segment; otherwise 0.
4. **harmonic simplicity**: 1 if the harmony is (consonant with) I or V7; otherwise 0.
5. **root-position**: 1 if the lowest pitch of the segment can be the root of the harmony; otherwise 0.
6. **second-inversion**: 1 if the lowest pitch of the segment can only be the fifth of the harmony; otherwise 0.
7. **harmonic support**: proportion of the surface in the span of a segment which is consonant with the harmony of that segment (measured in time).

Table 2. Reduction results.

Theme	Number of segments	Number of notes	Computation time (s)	Computation space (MB)	Total number of possible analyses	Number of possible analyses with *Ursatz*	Conformance to 'ground truth' of best-scoring analysis
K.545	22	58	138	141	6.3e12	9.2e12	–
K.333	19	40	125	88	7.5e11	1.0e11	79%
K.570	31	65	543	277	4.5e22	5.8e20	83%
K.494	34	82	2512	880	1.0e25	1.0e24	90%
K.331	17	55	248	230	1.8e15	8.3e14	91%
K.284	26	67	1524	510	1.3e21	1.2e20	98%

8. **pitch support**: proportion of the surface in the span of a segment which contains the pitches of that segment, averaged for each pitch of the segment.
9. **interval** in semitones between the paired notes of the child segments (i.e. the pairs of notes which participate in atomic elaborations), averaged per pair.
10. **voice split/join**: the number of notes in child segments which are paired with more than one note in the other segment.
11. **delay**: the number of atomic elaborations which start with a rest.
12. **shortening**: the number of atomic elaborations which finish with a rest.
13. **post-context from ancestor**: the number of levels of reduction between the lowest common ancestor and the required context (if any).
14. **post-context from surface**: the number of levels of reduction between the surface and the required context (if any).

For each measure and each analysis, the sum total of the measure for each segment in the analysis was computed, and then divided by the number of segments in the analysis.

5.6 Measurement results

Table 3 gives the mean value for each of these measures for the sets of results from each of the themes. For each theme, the row 'All' gives the values for the sample of all possible reductions (step 2 of the procedure outlined above), the row 'With *Ursatz*' from the sample of reductions with an *Ursatz* (step 3), the row 'Conforming' from the samples of reductions conforming to a 'ground truth' (step 4), and the row 'Preferred' from the single most preferred reduction in each case (step 5). The row 'Std dev.' gives the standard deviations of the sample of measure from all possible reductions (step 2). (The standard deviations from the *Ursatz* samples (step 3) were not significantly different.)

The objective of the sampling and measuring exercise was to discover factors which distinguish the 'good' analyses (those which conform to a ground truth, and the preferred analysis) from other possible analyses with an *Ursatz*. Thus in Table 3, every case of a 'Conforming' or 'Preferred' measure which differs from the 'With *Ursatz*' measure by more than one standard deviation is underlined. Columns corresponding to measures for which the difference is always positive or always negative (if it is not negligible) are shaded. These are measures which distinguish 'good' analyses from others in a consistent manner (if at all). The bottom row of the table gives the harmonic mean of the differences between the measures for the preferred reduction and the *Ursatz* sample in these nine cases, expressed in standard deviations. (The harmonic mean is used because it results in lower values for distributions with greater variance but the same mean. Thus high values result from consistently large differences.) From this we can propose nine principles for making good analyses, in order of decreasing strength.

1. Select higher level pitches which are more often present in the surface.
2. Avoid splitting and joining of voices.
3. Select reductions with small intervals between notes reduced together.
4. Reduce segments of approximately equal duration together.
5. Avoid reductions which create syncopations at higher levels.
6. Avoid reducing a shorter segment with a following longer segment.
7. Prefer reductions with more tonic and dominant harmony.
8. Avoid reductions where a note is followed by a rest.
9. Prefer reductions where higher level harmonies are more often consonant with the surface.

Some of these reflect ideas in the possible sources of goodness metrics referred to in Section 5.5 above. The first principle, for example, follows from Plum's (1988, p. 149) index-form 'continuation 1a', 'when a structural

Table 3. Goodness measure results.

	duration-ratio	short-long	syncopation	harmonic-simplicity	root-position	second-inversion	harmonic-support	pitch-support	voice-split/join	interval	delay	shortening	post-context from parent	post-context from surface
K.333														
Std dev.	*0.39*	*0.07*	*0.04*	*0.10*	*0.16*	*0.08*	*0.04*	*0.03*	*0.07*	*0.42*	*0.14*	*0.15*	*0.31*	*0.28*
All	2.17	0.17	0.07	0.76	0.63	0.07	0.69	0.47	0.15	2.05	0.42	0.36	0.31	0.46
With *Ursatz*	2.23	0.17	0.06	0.82	0.75	0.01	0.70	0.46	0.16	1.99	0.42	0.38	0.30	0.54
Conforming	1.70	0.13	0.03	0.88	0.81	0.00	0.74	0.51	0.14	1.76	0.36	0.38	0.19	0.60
Preferred	1.56	0.11	0.03	0.89	0.91	0.00	0.74	0.54	0.06	1.16	0.28	0.28	0.33	0.50
K.570														
Std dev.	*0.24*	*0.08*	*0.04*	*0.03*	*0.12*	*0.07*	*0.03*	*0.02*	*0.07*	*0.44*	*0.10*	*0.10*	*0.26*	*0.21*
All	2.12	0.28	0.14	0.85	0.68	0.08	0.67	0.50	0.21	2.58	0.36	0.43	0.36	0.25
With *Ursatz*	2.21	0.26	0.14	0.85	0.74	0.04	0.67	0.48	0.17	2.40	0.33	0.40	0.30	0.27
Conforming	1.95	0.27	0.11	0.87	0.66	0.06	0.70	0.48	0.14	2.05	0.32	0.41	0.16	0.24
Preferred	1.40	0.03	0.00	0.90	0.53	0.13	0.71	0.52	0.00	1.17	0.27	0.37	0.14	0.29
K.494														
Std dev.	*0.24*	*0.07*	*0.04*	*0.07*	*0.13*	*0.06*	*0.03*	*0.02*	*0.06*	*0.32*	*0.10*	*0.09*	*0.24*	*0.21*
All	2.04	0.25	0.11	0.87	0.57	0.05	0.72	0.51	0.19	1.82	0.46	0.39	0.29	0.36
With *Ursatz*	1.99	0.24	0.10	0.89	0.65	0.02	0.73	0.52	0.19	1.77	0.47	0.40	0.22	0.47
Conforming	1.92	0.25	0.08	0.95	0.63	0.00	0.78	0.53	0.15	1.45	0.45	0.41	0.08	0.57
Preferred	1.30	0.03	0.00	0.94	0.55	0.00	0.74	0.58	0.00	1.13	0.21	0.18	0.13	0.63
K.331														
Std dev.	*0.32*	*0.09*	*0.04*	*0.06*	*0.20*	*0.10*	*0.05*	*0.04*	*0.12*	*0.50*	*0.13*	*0.10*	*0.18*	*0.17*
All	2.83	0.29	0.10	0.92	0.54	0.08	0.66	0.53	0.45	2.86	0.42	0.22	0.21	0.09
With *Ursatz*	2.83	0.28	0.10	0.92	0.66	0.02	0.67	0.53	0.44	2.85	0.42	0.24	0.19	0.08
Conforming	2.11	0.16	0.06	0.95	0.70	0.02	0.70	0.62	0.29	1.98	0.28	0.20	0.42	0.18
Preferred	2.44	0.00	0.00	1.00	0.71	0.00	0.70	0.65	0.06	0.99	0.13	0.19	0.17	0.67
K.284														
Std dev.	*0.28*	*0.07*	*0.04*	*0.05*	*0.14*	*0.07*	*0.03*	*0.02*	*0.07*	*0.44*	*0.11*	*0.11*	*0.25*	*0.28*
All	2.10	0.24	0.12	0.79	0.71	0.06	0.68	0.46	0.20	2.56	0.45	0.39	0.25	0.36
With *Ursatz*	2.19	0.25	0.12	0.78	0.74	0.04	0.68	0.46	0.20	2.39	0.43	0.36	0.25	0.38
Conforming	1.90	0.19	0.09	0.80	0.84	0.02	0.73	0.49	0.18	2.16	0.41	0.37	0.33	0.34
Preferred	1.08	0.04	0.02	0.80	0.93	0.00	0.72	0.51	0.00	1.48	0.44	0.32	0.75	0.50
Weight	**−2.16**	**−1.93**	**−2.05**	**0.75**			**0.37**	**2.72**	**−2.39**	**−2.35**		**−0.54**		

note appears to be 'prolonged' by repetition'. The third, fourth, fifth and seventh principles are reflected in the preference rules of Lerdahl and Jackendoff. For example, the third is similar to their PRPR 3 in its 'melodic condition (distance)' which finds more stable connections in smaller intervals (Lerdahl & Jackendoff, 1983, pp. 351–352), and is reflected also in their GPR 3 (Lerdahl & Jackendoff, 1983, p. 346), and indeed in commonly espoused Gestalt principles of grouping by pitch proximity. As mentioned above, however, a systematic attempt to test ideas from those writers is beyond the scope of this article.

From the product of the harmonic means and the standard deviations, a weight can be computed to be used in combining each measure into a composite 'goodness metric'. For simplicity, linear combination was used (i.e. each measure is multiplied by the appropriate weight and the results added together).

5.7 Selection of best-scoring analysis

The scoring metric could now be used in a procedure which selected the best-scoring analysis from a reduction chart in order to complete the process of automatically deriving an analysis from the notes of the surface (step 8 of the process outlined in Section 5.3 above). First, reduction charts were recomputed for each theme using the goodness metric proposed above. Most segments above the surface of the chart were parents of a number of possible elaborations, sometimes with the same pair of children and sometimes with different pairs. The score recorded for a segment was the best among the various possibilities. The best-scoring analysis could then be selected by selecting the highest-scoring top-level segment which had a complete *Ursatz* among its recorded *Ursatz* possibilities. Children were selected in a recursive procedure which, at each level, selected the highest-scoring pair of children which yielded this *Ursatz*. (It was here that it was vital to have recorded the *Ursatz* possibilities of each segment.) However, in many cases the recorded best score for a segment arose from an elaboration or pair of children which was no longer valid, either because a required context had been removed as a result of selection of segments in another part of the chart, or because they did not conform to the required *Ursatz*. Thus the complete analysis selected by this means might not have a final score as high as that recorded for the top-level segment. A best-first search procedure was therefore used to ensure that the highest-scoring valid analysis with an *Ursatz* was found.

In no case did a top-scoring analysis derived by this procedure exactly match a preferred analysis. On the other hand, the results did generally conform quite well to the ground truths derived from pre-existing analyses. It was to be expected that the ground truths would contain fewer notes than the computed analyses because they contained less detail. An appropriate test would check the inclusion of notes from the ground truths in the automatically derived analyses but not vice versa. Thus the measure used here was simply to count the proportion of notes in the ground truth reductions which occurred at the same time and with the same duration in the automatically computed analyses. This was nevertheless quite a severe test, since where there were deviations the automatically derived analysis often showed some similarity with the ground truth (for example by having the same pitches but in a different rhythm) but this similarity was not taken into account in the test. The match with the ground truth was therefore generally better than suggested by the figures given in Table 2, but these figures are nevertheless encouraging. As an example, Figure 7 shows the automatically derived analysis for the K.333 theme, which was the worst match with the ground truth among the themes tested. (The ground truth for this theme is shown in Figure 5.)

5.8 Pruning

As indicated above, the reduction procedure is extremely expensive in terms of computing time and space. Because the complexity arises from combination, reducing the number of segments to be combined at an early stage has a disproportionate impact on the overall computing cost. It would thus be beneficial to discard low-scoring segments in

Fig. 7. Analysis of the K.333 theme with the highest 'goodness' score, derived entirely automatically by the software.

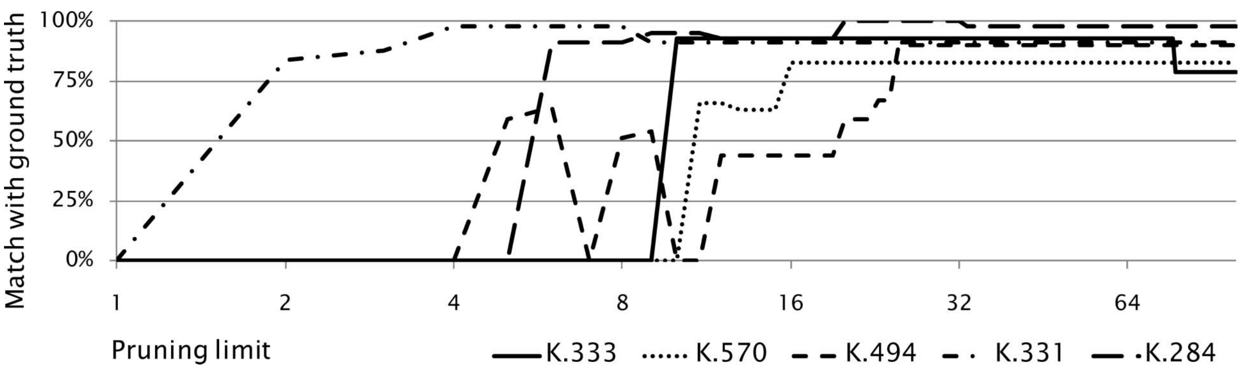

Fig. 8. Match with ground truth against pruning limit for each theme.

the course of the reduction procedure if this can be done with confidence that a good analysis is not thereby jeopardized. To test this, the reduction procedure was rerun on the five examples which produced analyses conforming to the ground truth, with one simple modification. This was to limit the number of segments retained in a cell after each reduction step, and discard lower-scoring segments beyond that limit. The limit ranged from 1 to 100, and the resulting best-scoring analyses were tested for conformity with the ground truths in the same manner as described in Section 5.7 above. The resulting matches with the ground truth reductions are shown in Figure 8. A figure of 0% indicates that the system failed to derive an analysis at all with the given pruning limit.

In no case was an analysis derived by keeping only the best-scoring segment at each reduction step (i.e. with a pruning limit of 1). In every case, the best-scoring analysis derived without any pruning (as in Section 5.7 above) was derived with a pruning limit of 79 or less. In four of the five cases, the best score was already reached at a pruning limit of 35. It was only for K.333 that the much larger pruning limit was necessary to achieve the maximum score, and in this case the match with the ground truth is seen to be markedly worse than with a much lower pruning limit of 10, even though the score is higher. A higher score (as measured by the metric derived in Section 5.6 above) does not therefore necessarily mean a better analysis. For comparison with Figure 7, Figure 9 shows the best-scoring analysis for the K.333 theme derived with a pruning limit of 10.

In two other cases we also see a reduction in the match with the ground truth with the last increases in the pruning limit. In fact, in the case of K.284, there is a perfect match with the ground truths for the best-scoring derived analysis with a pruning limit in the range 20 to 32. Although this decrease in the 'quality' of the analyses with less pruning is unexpected, it is actually an encouraging result. It suggests that the factors which determine a good analysis are, to some degree at least, local, since the pruning decision is taken only on the basis of local information. It should be possible, therefore, to find a better pruning mechanism which makes better use of this local information. This also

Fig. 9. Analysis of the K.333 theme with the highest 'goodness' score, derived entirely automatically by the software with a pruning limit of 10.

suggests that a reduction procedure based on search (see below) has some chance of success. For the present, it appears that a simple pruning limit of about 25 is likely to produce a good analysis, but cannot guarantee the best analysis.

6. Conclusion and discussion

6.1 Outcome

The essential outcome of the research described here is a computational procedure capable of deriving Schenkerian-like reductions from the pitch and time information in a score. The reductions do match, to some considerable degree, the analyses that expert analysts make. There are two very significant caveats, however.

Firstly, the procedure is extremely expensive in terms of computation time and space, and only short phrases can be processed with realistic resources. While the results from pruning are encouraging, I suspect that ultimately an entirely different procedure is required: one which has a lower order of complexity as a result of its essential process rather than as a result of incidental savings. Kirlin and Utgoff (2008) have advocated intelligent search as a solution to this problem, and Geraint Wiggins and I have already had some small-scale success in implementing a search procedure (Marsden & Wiggins, 2008). Such a procedure will almost certainly not be guaranteed to produce the 'best' result, but to do so is probably unrealistic for this particular problem.

The second caveat is that the quality of analyses produced has not been adequately tested, so the claim that the procedure produces Schenkerian-like reductions must be regarded as provisional. The same sources were used both in the derivation of goodness measures and in testing of those measures, which will tend to over-fitting. Tests on different material are required, but will have to await collection of suitable ground truths.

6.2 Music-theoretic lessons

What does this outcome tell us about Schenkerian analysis? In part it confirms the views expressed by others (e.g. Temperley, 2007, p. 176) that the explicit principles of Schenkerian analysis are too loose to determine an analysis, because the principles themselves, at least as formulated here, produce so many analyses, some of them very bad. On the other hand, the fact that quite acceptable analyses can be derived automatically when the 'goodness metric' is also applied appears to contradict Schachter's (1990, pp. 166–167) view that formally methodical Schenkerian analysis is impossible. If we take 'Schenkerian analysis' to be defined by the products of music analysts, however, I have to concede that Schachter is probably right. To fully mimic what music analysts produce would require the '"world model" of music perception' which Frankel et al. (1978, p. 134) suggested to be a stumbling block in the early days of such research: a complete 'world model' is an impossibility. The long-term value of this research to music theory is likely to be a system which, precisely because it does not depend on a 'world model' allows analysts to explore intelligently quite what it is about their own 'world models' which causes them to read a piece as having one structure rather than another.

In a similar vein, the results here suggest that there are aspects of Schenkerian reduction which arise in a bottom-up fashion from the music under analysis. As discussed in Section 3.4, the essential context-dependencies are only local, and the results from the experiments with pruning suggest also that local information is, to some degree, more important than global. On the other hand, there are other aspects where global context is crucial. To match the ground-truth analyses required imposition, top-down, of an *Ursatz*. Some of the preference rules of Lerdahl and Jackendoff similarly suggest top-down context-dependencies (e.g. rules to prefer an analysis in one domain which allows a more stable structure in another), as do some of Plum's indices (e.g. 'goal orientation' (Plum, 1988, pp. 148–149)). Indeed, many of Schachter's (1990) arguments to prefer one analytical interpretation over another are based on features of the overall structures which result from the different interpretations. An advantage of the approach taken here is that it allows the influence of global context to be isolated and identified, just as it allows the influence of a 'world model' to be isolated. In both cases, it becomes clear what in an analysis arises from the patterns of notes and a language of how they relate, and what comes from what the analyst brings to the piece from his or her previous knowledge.

Temperley (1999) has called for music analysts to be more clear about whether they aim to *describe* the structures in a piece of music or to *suggest* structures which a listener might profitably hear. As indicated above, I believe this research potentially allows description and suggestion to be distinguished on a systematic basis, but for this to become really solid requires further research on the degree to which Schenkerian analysis reflects actual listening. There have been attempts to test whether reduction does play some part in listening to or remembering music (e.g. Oura & Hatano, 1991) but such research is rare. Perhaps this is because there has been little to test with, for the same reason that material for this study has been hard to come by: complete analyses of short complete extracts are rare. I hope that the work reported here, in providing at least some mechanism for the systematic derivation of reductions, might facilitate such research in future.

Acknowledgements

This research was supported by a research-leave award from the UK Arts and Humanities Research Council in 2008. I also gratefully acknowledge the co-operation of Michael Kassler, Panayotis Mavromatis and Philip Kirlin in responding to requests for information about their work.

References

Barbar, K., Desainte-Catherine, M., & Miniussi, A. (1993). The semantics of musical hierarchies. *Computer Music Journal*, *17*, 30–37.

Baroni, M. (1983). The concept of musical grammar (translated by S. Maguire with the assistance of W. Drabkin). *Music Analysis*, *2*, 175–208.

Baroni, M., Dalmonte, R., & Jacobini, C. (1992). Theory and analysis of European melody. In A. Marsden & A. Pople (Eds.), *Computer Representations and Models in Music* (pp. 187–206). London: Academic Press.

Bernstein, L. (1976). *The Unanswered Question: Six Talks at Harvard*. Cambridge, MA: Harvard University Press.

Cadwallader, A., & Gagné, D. (2007). *Analysis of Tonal Music* (2nd ed.). Oxford: Oxford University Press.

Desainte-Catherine, M., & Barbar, K. (1994). Using attribute grammars to find solutions to musical equational programs. *ACM SIGPLAN Notices*, *29*, 56–63.

Ebcioglu, K. (1987). *Report on the Choral Project: An expert system for harmonizing four-part chorales* (Technical report RC 12628). IBM, Thomas J. Watson Research Centre, Yorktown Heights, New York.

Ebcioglu, K. (1988). An expert system for harmonizing four-part chorales. *Computer Music Journal*, *12*, 43–51.

Ebcioglu, K. (1990). An expert system for harmonizing chorales in the style of J.S. Bach. *Journal of Logic Programming*, *8*, 145–185.

Forte, A., & Gilbert, S.E. (1982a). *Introduction to Schenkerian Analysis*. New York: Norton.

Forte, A., & Gilbert, S.E. (1982b). *Instructor's Manual for 'Introduction to Schenkerian Analysis'*. New York: Norton.

Frankel, R.E., Rosenschein, S.J., & Smoliar, S.W. (1976). Schenker's theory of tonal music—its explication through computational processes. *International Journal of Man–Machine Studies*, *10*, 121–138.

Frankel, R.E., Rosenschein, S.J., & Smoliar, S.W. (1978). A LISP-based system for the study of Schenkerian analysis. *Computers and the Humanities*, *10*, 21–32.

Gilbert, E., & Conklin, D. (2007). A probabilistic context-free grammar for melodic reduction. In *Proceedings of the International Workshop on Artificial Intelligence and Music, 20th International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, pp. 83–94.

Hamanaka, M., Hirata, K., & Tojo, S. (2006). Implementing 'A generative theory of tonal music'. *Journal of New Music Research*, *35*, 249–277.

Hamanaka, M., Hirata, K., & Tojo, S. (2007, August). FATTA: Full automatic time-span tree analyzer. In *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, Denmark, pp. 153–156.

Jurafsky, D., & Martin, J.H. (2009). *Speech and Natural Language Processing* (2nd ed.). Upper Saddle River, NJ: Pearson.

Kassler, M. (1967). *A trinity of essays* (PhD dissertation). Princeton University, USA.

Kassler, M. (1975). *Proving musical theorems I: The middleground of Hienrich Schenker's theory of tonality* (Technical report no. 103). Sydney, Australia: University of Sydney, School of Physics, Basser Department of Computer Science.

Kassler, M. (1976). The decidability of languages that assert music. *Perspectives of New Music*, *14(2)–15(1)*, 249–251.

Kassler, M. (1977). Explication of the middleground of Schenker's theory of tonality. *Miscellanea Musicologica: Adelaide Studies in Musicology*, *9*, 72–81.

Kassler, M. (1988). APL applied in music theory. *APL Quote Quad*, *18*, 209–214.

Kirlin, P.B. (2009). Using harmonic and melodic analyses to automate the initial stages of Schenkerian analysis. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Kobe, Japan, pp. 423–428.

Kirlin, P.B., & Utgoff, P.E. (2008). A framework for automated Schenkerian analysis. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Philadelphia, PA, USA, pp. 363–368.

Komar, A.J. (1971). *Theory of Suspensions*. Princeton, NJ: Princeton University Press.

Larson, S., & Vanhandel, L. (2005). Measuring musical forces. *Music Perception*, *23*, 119–136.

Lerdahl, F., & Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. Cambridge, MA: MIT Press.

Marsden, A. (2001). Representing melodic patterns as networks of elaborations. *Computers and the Humanities*, *35*, 37–54.

Marsden, A. (2005). Generative structural representation of tonal music. *Journal of New Music Research*, *34*, 409–428.

Marsden, A. (2007). Automatic derivation of musical structure: A tool for research on Schenkerian analysis. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Vienna, Austria, pp. 55–58.

Marsden, A., & Wiggins, G.A. (2008). Schenkerian reduction as search. In *Proceedings of the Fourth Conference on Interdisciplinary Musicology (CIM08)*, Thessaloniki, Greece. Retrieved 8 December 2009 from http://cim08.web.auth.gr/cim08_papers/Marsden-Wiggins/Marsden-Wiggins.pdf

Mavromatis, P., & Brown, M. (2004). Parsing context-free grammars for music: A computational model of Schenkerian analysis. In *Proceedings of the 8th International Conference on Music Perception and Cognition*, Evanston, IL, USA, pp. 414–415.

Orio, N., & Rodà, N. (2009). A measure of melodic similarity based on a graph representation of the music structure. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Kobe, Japan, pp. 543–548.

Oura, Y., & Hatano, G. (1991). Identifying melodies from reduced pitch patterns. *Psychologica Belgica*, *31*, 217–237.

Pachet, F. (2000). Computer analysis of jazz chord sequences: Is *Solar* a blues? In E. Miranda (Ed.), *Readings in Artificial Intelligence and Music* (pp. 85–113). Amsterdam: Harwood Academic Publishers.

Pankhurst, T. (2008). *SchenkerGUIDE*. London: Routledge.

Plum, K-O. (1988). Towards a methodology for Schenkerian analysis (translated by William Drabkin). *Music Analysis*, *7*, 143–164.

Schachter, C. (1990). Either/Or. In H. Siegel (Ed.), *Schenker Studies* (pp. 165–179). Cambridge: Cambridge University Press. (Reprinted in C. Schachter, 1999, *Unfoldings*, pp. 121–133, Oxford: Oxford University Press).

Schenker, H. (1935). *Der frei Satz*. Vienna: Universal Edition. (Published in English as E. Oster (Trans. & Ed.), 1979, *Free Composition*, New York: Longman).

Smoliar, S.W. (1980). A computer aid for Schenkerian analysis. *Computer Music Journal*, *4*, 41–59.

Steedman, M.J. (1984). A generative grammar for jazz chord sequences. *Music Perception*, *2*, 52–77.

Steedman, M.J. (1996). The blues and the abstract truth: Music and mental models. In A. Garnham & J. Oakhill (Eds.), *Mental Models in Cognitive Science* (pp. 305–318). Hove: Psychology Press.

Temperley, D. (1999). The question of purpose in music theory: Description, suggestion, and explanation. *Current Musicology*, *66*, 66–85.

Temperley, D. (2007). *Music and Probability*. Cambridge, MA: MIT Press.

Terrat, R.G. (2005). A pregroup grammar for chord sequences. *Journal of New Music Research*, *34*, 355–360.

Yust, J.D. (2006). *Formal models of prolongation* (PhD dissertation). University of Washington, USA.