

The Logical Structure of an Algorithmic Theory of Tonal Music

David Meredith

St. Anne's College, University of Oxford

Thesis submitted for the degree of D.Phil.

Abstract for

The Logical Structure of an Algorithmic Theory of Tonal Music

David Meredith

St. Anne's College, University of Oxford

Thesis submitted for degree of D.Phil. in Trinity term 1996

A study of the metatheoretical problems suffered by theories that aim to characterize musical styles leads to certain conclusions as to the logical structure that should be possessed by a style theory and the methodology that should be employed in a style theory research programme.

Some of the research that has led to formal style theories has been based upon Chomskyan generative linguistics. But there are serious flaws in Chomsky's methodology that have been transmitted to the musicological research influenced by it.

Chomsky believes that a grammar must not only weakly generate the sentences of a language, but also strongly generate correct structural descriptions of these sentences. I believe weak generation is a sufficient condition on the adequacy of a grammar because strong generation cannot be tested.

In Part 1, I propose that a theory for a musical style should be an *algorithmic style theory* which is a hypothesis that a *composing algorithm* generates all and only those pieces that are either members of a *corpus* or in the style of that corpus, as determined by an *acceptability algorithm* resembling a Turing test. An algorithmic style theory is always either true or false, and either refuted or unrefuted. It can never be verified but it should be tested for overgeneration and undergeneration until it is refuted.

Chomsky believes that a linguistic performance theory should consist of a competence theory that generates *grammatical* sentences, supplemented by 'performance filters' that take grammatical sentences as input and generate *acceptable* sentences as output. Because *acceptability* can be operationally defined, a performance theory can be refutable. But *grammaticalness* cannot be operationally defined, so a competence theory cannot be tested. A competence theory becomes partially testable when it forms part of a performance theory, but competence should not be studied in isolation from performance.

Some style theorists who study competence in isolation from performance are more concerned with the elegance of their theories than with weak generation. Others believe that avoiding overgeneration is more important than avoiding undergeneration. I believe style theorists should be more concerned with accounting for existing pieces in a style than with avoiding overgeneration or with keeping their theories as simple as possible.

Many different theories might successfully approximate a style, but the complexity of acceptable music severely limits the class of possibly correct algorithmic style theories for a style. Weak generation is therefore a sufficient condition on the adequacy of an algorithmic style theory.

In Part 2, I define a formal system for representing pitch in tonal music and describe how pitch relations can be represented using digraphs. I prove a theorem that relates circuits in the thirds relation digraphs with the traditional major and minor scales. Finally, I describe the progress I have made towards developing an algorithmic style theory for the style of a subset of Bach's chorale harmonizations.

(Approximate total number of words in thesis : 97000)

Part 1

1 Introduction

1.1 *Introduction and overview*

My research was originally motivated by a desire for a satisfactory theory for the style of J.S. Bach's chorale harmonizations. I was dissatisfied by the fact that the rules that one is taught when learning to harmonize melodies in the style of Bach were not obeyed by Bach himself and were neither necessary nor sufficient to characterize precisely the style of Bach's own compositions in the genre. This led me to try to determine what features a theory for a particular style would have to have before I would consider it to be satisfactory. Existing theories that seemed to be attempts to characterize styles had certain technical and metatheoretical features that made them unsatisfactory. From a consideration of these theories and the methodological principles of the research programmes that led to their development emerged certain conclusions as to the logical structure that should be possessed by a theory for a musical style and the methodological principles that should be adopted in a research programme aimed towards the development of such a theory.¹

Over the course of the first part of this thesis I shall attempt to justify the view that a theory for a particular style should take the form of a hypothesis that a well-defined set of scores generated by a 'grammar' is equal to all and only those pieces that are either members of a specified corpus or that are in the style of that corpus, where whether or not a given test piece is 'in the style of' the corpus is determined by means of a specified 'acceptability algorithm' that takes the form of an experiment along the lines of a Turing test (Turing 1950) in which subjects are required to identify the test piece among a sample of pieces taken from the corpus.

In Part 1 of this thesis I shall examine a number of research programmes that have led to the development of formal theories of Western tonal musical styles, that is, theories that can be construed to be attempts to account for at least one, more or less well-defined, Western tonal musical style in a more or less explicit and precise manner.

Formal theories have been developed for a variety of styles of Western tonal art music such as, for example, Bach's chorale harmonizations (Ebcioğlu 1987b), early French chansons (Baroni and Callegari 1984), keyboard works by C.P.E. Bach (Snell 1979) and Debussy's melodies (Wenck 1988). Such theories have also been developed for popular and folk styles such as modern jazz (Johnson-Laird 1991), 12-bar blues (Steedman 1984), ragtime (Ames and Domino 1992), rock songs (Moorer 1972) and nursery tunes (Sundberg and Lindblom 1993).

Formal theories for tonal styles range from those that are attempts to account for styles represented by enormous and heterogeneous repertoires to those that aim only to account for styles defined by very small and homogeneous corpora. For example, while the theories of Heinrich Schenker (1979) and Lerdahl and Jackendoff (1983) are intended to apply to most Western classical tonal music, Sundberg and Lindblom (1993) developed a grammar for a style represented by only a small corpus of Swedish nursery tunes by Alice Tégner, and Baroni, Dalmonte and Jacoboni (1989, 1992) have attempted to model a style represented by a corpus of just 17 arias by Giovanni Legrenzi.

¹ The term 'research programme' is used throughout this thesis in the sense of Lakatos 1970.

In this thesis I shall be considering only those theories that achieve some degree of precision and explicitness. However, such theories vary widely with respect to just how precise and explicit they are. Thus while some are sufficiently well-defined to be implemented directly as working computer programs (e.g. Ames' program, Cybernetic Composer (Ames and Domino 1992) and Ebcioğlu's CHORAL program (Ebcioğlu 1987b)), others are presented in a superficially 'formal' or 'technical' manner but are not sufficiently explicit to be used directly as specifications for computer programs (e.g. Schenker 1979; Lerdahl and Jackendoff 1983).

Some theories such as that described in Ebcioğlu 1987b are so complete that they have been embodied in programs that automatically compose entire, original pieces of music and produce output in the form of physical, printed scores in standard staff notation. On the other hand, some theories stop well short of generating complete pieces in the styles that they are intended to model. For example, Baroni and Jacoboni (1973, 1976) developed a grammar that was intended to account only for the soprano part of the first two phrases of major-mode chorales by Bach. Similarly, Kassler's (1975) automatic parser embodies an explication of only that part of Schenker's theory that deals with the middleground, and Steedman (1984) developed a grammar that aimed to account only for the chord sequences of 'grammatical' 12-bar blues.

Formal theories for Western tonal musical styles also vary with respect to their 'generalizability.' Thus the principal measure of success in some research programmes has been the extent to which the theory is capable of modelling a single, homogeneous style (e.g. Ebcioğlu 1987b). In other cases the goal has been to model two or more distinct styles using theories that have as much in common as possible. For example, Ames' computer program, Cybernetic Composer, automatically composes pieces in four distinct popular styles—'standard' jazz, Latin jazz, rock and ragtime—using four closely related algorithms (Ames and Domino 1992), and Baroni, in collaboration with a number of co-workers, has used three different realizations of essentially the same grammar to model three distinct melodic repertoires (Baroni and Jacoboni 1983; Baroni and Callegari 1984; Baroni, Dalmonte and Jacoboni 1989). An example of a theory that is intended to be extremely generalized is Cope's EMI program (Cope 1991) which, when given any set of pieces as input, automatically composes new pieces that are intended to be 'in the style of' these input pieces.

1.2 *Ill-defined goals, serial music and algorithmic composition*

James Snell has pointed out that 'in any attempt to formulate a generative theoretical model to account for the structure of some phenomenon (whether linguistic or not), one must decide on measures of adequacy by means of which to evaluate the model.'² Unfortunately, as Baroni has noted, 'the scope and purpose of ... studies [involving the construction of a computer program for composing tonal music] have not always been clear or firmly defined.'³ The logical structures and methodologies of a number of research programmes in the field are faulty either because researchers have not adequately defined their goals or the purpose that is to be served by the composing program, or because the goals themselves are not valid or not what the researchers claim them to be.

Indeed, as Baroni states,

² Snell 1979, 57.

³ Baroni 1983, 179.

scholars have sometimes had a more frivolous end in mind, and on occasion experiments in music seem to have been undertaken rather in the spirit of constructing computers to play chess; in fact, in many cases, the appearance of a ‘machine that sings’ has been of no greater value than a *musikalischer Spass*.⁴

Ebcioğlu similarly bemoans the fact that ‘computer enthusiasts with an interest in music (as opposed to formally trained musicians) are prominent in the MIDI software market’ with the result that trivial programs that use ‘real-time transitions between predetermined riffs, with small random variations’ to produce so-called “computer improvisations in the classical style” are enough to please both the developers and the audience.⁵

It is certainly true that many workers in the field of computational tonal theory have been less concerned than Snell, Baroni and Ebcioğlu with defining coherently and precisely the goals of their research. For example, in his published work Cope has stated the goals of his EMI project only in rather vague terms. He claims that the purpose of EMI is ‘the replication of musical styles’⁶ and that EMI is ‘a project for understanding musical style.’⁷ He claims that in the first instance, EMI was developed as a ‘composer’s aid’ able to ‘replicate’ his own style.⁸ The program is therefore described as a ‘friendly antagonist during composition,’ ‘an analysis tool for generating extensive lists of motivic patterns’ and ‘an imitative projector of possible next intervals of given phrases.’⁹ In the introduction to Cope 1991, John Strawn describes EMI as ‘a computer program that can accurately represent and freely manipulate musical styles.’¹⁰ None of these descriptions of the purpose of EMI constitutes a clear definition of the goals of the project or the task that the program is intended to be able to perform. The fact that Cope has not expressed his aims in anything other than rather nebulous terms suggests that he himself has only a vague idea of what he hopes to achieve.

It seems that, in general, composers who attempt to produce formal style theories suffer from a conflict of interests—their theories tend to be more or less thinly disguised attempts to justify their own methods of composition. For example, Brown and Dempster claim ‘that although [Boretz] claimed to produce scientific theories, [he] failed to achieve this goal because he was unable to reconcile the need for general laws with his desire ... to develop maxims for composing new pieces.’¹¹ Cross has similarly suggested that Lerdahl and Jackendoff’s decision to

bias their theory towards structural description rather than piece generation might be ideological, or rather, might be an attempt to conceal an ideology. In many respects Lerdahl’s more recent work (in particular, his work on the cognitive unfeasibility of atonal music) has tended to indicate that he views the theory as a vindication of *his* chosen compositional style, and that the inclusion of some of its constituent elements are motivated not solely by a desire to model ‘tonal musical intuitions’ but *in particular* his own musical preferences.¹²

⁴ Baroni 1983, 179.

⁵ Ebcioğlu 1996a.

⁶ Cope 1991, xi.

⁷ Cope 1989a, 117.

⁸ Cope 1989a, 117.

⁹ Cope 1987a, 30.

¹⁰ Cope 1991, ix.

¹¹ Brown and Dempster 1989, 66.

¹² Cross 1996a.

Peel and Slawson accuse Lerdahl and Jackendoff of taking the ‘position that if a piece of music cannot be demonstrated to possess the nested hierarchies in the four realms [of metric, grouping, time-span and prolongational structure]... then the piece is at best superficial.’¹³ Peel and Slawson compare this reactionary view to that expressed by Schenker in his *Das Meisterwerk in der Musik* (Volume 2) where ‘there can be found a musical ‘counterexample’ [to Schenker’s theory]—a demonstration that Stravinsky’s Concerto for Piano and Winds cannot rewardingly be heard according to the Schenkerian theory of tonality.’ Peel and Slawson point out that ‘Schenker interprets this not as a limitation of his theory but as evidence that the music is ‘bad, lacking in craft, and unmusical.’¹⁴

I think it is fair to say that much 20th century music differs intentionally from much preceding music in that the structural principles of many 20th century pieces are artificially constructed from scratch without any real concession to any auditory principles of organization. This applies not only in the realm of pitch structure but also that of orchestration. For example, Deliège (1987) showed that change in timbre is by far the strongest cue to local grouping structure and suggests that this may be because listeners ‘tend to track a given sound source in listening’ and thus ‘a change in source would easily introduce a discontinuity and thus a segmentation.’¹⁵

The strength of timbre change as a determinant of local grouping structure militates strongly against the view that an orchestration such as Webern’s of Bach’s six-part *Ricercar* from *The Musical Offering* was motivated by primarily aural considerations. Indeed, it suggests strongly that most listeners would find Webern’s ‘pointillistic’ orchestral technique a hindrance rather than a help to achieving an understanding of the music. Frequent and abrupt changes in timbre make it much more difficult for listeners to achieve a coherent and satisfying comprehension of the musical structure of a piece.¹⁶

Thus, whereas it can be argued that the success of much 20th century music must be judged on grounds other than how it sounds (for example, on the basis of whether or not its principles of construction adhere to certain artificial rules) the structural constraints on most pieces of *tonal* music are principally aural—that is, whether or not a tonal piece is judged by its composer to be successful depends primarily on how it *sounds*.

I therefore disagree with Peel and Slawson’s conclusion that because ‘tonal music is older than serial music,’ ‘our theories of tonal music are far better developed than our theories of atonal and serial music.’¹⁷ In my view, serial music can generally be explained in far greater detail than tonal music because its principles of construction are for the most part completely conscious and explicit. The principles of construction of tonal music are to a much larger extent subjective, unconscious and not readily formalizable. Consequently, theories of tonal music generally fall short of theories of serial music with respect to the completeness of the explanations that they generate for the pieces they aim to account for. I thus agree with Ebcioğlu when he notes that

¹³ Lerdahl and Jackendoff 1983, 296–301.

¹⁴ Peel and Slawson 1984, 291.

¹⁵ Deliège 1987, 357.

¹⁶ Lerdahl and Jackendoff (1983, 296–301) make essentially the same point by implying that such sudden changes of timbre prevent the listener from hearing a piece in terms of hierarchical time-span and prolongational structures.

¹⁷ Peel and Slawson 1984, 292.

‘traditional music, and most of modern music, which are usually composed without a computer, rarely permit economical characterizations.’¹⁸

However Ebcioğlu goes on to suggest that ‘in the traditional style, the basic training in harmony, strict counterpoint, fugue and orchestration that the composer has to go through before even beginning to compose, already imposes a certain minimal complexity on the amount of knowledge required to characterize the style.’¹⁹ While this is undoubtedly true, I do not think that it is this fact that makes it difficult to find an ‘economical characterization’ for tonal music. In fact, one would expect that the style of a ‘learned’ composer with an extensive academic training in traditional techniques would in general be more superficially consistent and rule-bound and thus *easier* to model algorithmically than the style of an ‘intuitive’ composer whose works are governed not by explicit, learned and conscious principles but by his or her own, unconsciously evolved musical taste and preferences.

Although a large number of automatic composition programs have been developed (see Ames, 1987 for a review) most have been designed to generate non-tonal music according to compositional systems developed by modern composers (for example, Xenakis, 1971; Bolognesi, 1983; Belfiore, 1985; Gill, 1963; Kendall, 1981; Holtzman 1980) and some are embodiments of grammars for non-Western music (e.g. Pelinski, 1984; Lippus, 1980; Kippen and Bel, 1992—see Hughes, 1991 for an overview). Only a small number of programs have been developed for the purpose of automatically generating *tonal* music.

Ebcioğlu has lamented the fact that ‘computer generated tonal music has somehow failed to be popular among computer musicians’ and has suggested that this might be ‘because of its reactionary overtones.’²⁰ He claims that ‘typically the musically trained avant-garde composers who use computers for algorithmic composition do not consider tonal music important, and those computer enthusiasts who do have an interest in algorithmic tonal music composition are not musically trained.’²¹

In my view, it is essential to make a strong distinction between the activity of ‘algorithmic composition’ in which a modern-day composer uses a computer program to compose or help to compose pieces in his or her own style and the activity of attempting to produce an explicit theory for a style of non-algorithmically composed music by writing a computer program that is intended to embody the necessary and sufficient knowledge required for composition of all and only possible pieces in that style.

I thus agree with Ames when he states that ‘composing programs such as CHORAL and Cybernetic Composer ultimately serve an analytic purpose, since their entire reason for being is as vehicles for determining necessary criteria for generating known musical styles.’ I also basically agree with his claim that ‘the difference between active style synthesis—composition—and empirical style modeling is simply the extent to which the generative criteria have been devised “before” or “after” the fact.’²² But I disagree strongly with the rhetorical implication that this difference is a small one and

¹⁸ Ebcioğlu 1987b, 3.

¹⁹ Ebcioğlu 1987b, 3.

²⁰ Ebcioğlu 1992, 327.

²¹ Ebcioğlu 1996a.

²² Ames 1992, 55.

that the activities of ‘active style synthesis’ and ‘empirical style modeling’ should therefore be considered essentially the same thing.

‘Empirical style modeling’ is a scientific enterprise aimed towards developing a theory for the style being simulated. The purpose of a composing program in such an enterprise is to produce random samples of pieces from the artificial language that it generatively specifies. The purpose of a composing program that generates new, ‘original’ music for a composer is totally different. In this case, the program is being used as a creative tool. As Brown and Dempster point out, ‘while it may be important to devise systems for composing new pieces, we should not confuse this pursuit with the explanation of those pieces.’²³ Ebcioğlu explains that attempting to write a computer program capable of composing all and only the pieces in some ‘real traditional style’ allows for ‘an objective evaluation of the results, and for probing the complexity involved in the mechanical generation of a non-computer style of music.’²⁴

Baroni claims that early experiments in writing computer programs for ‘pastiche’ composition ‘paved the way to clearer ‘generative’ theories based on a double approach to the analysis of synthesis.’ He claims that these programs established a sound methodology for the computer study of musical style in which one began ‘with the investigation of the source [i.e. pieces in the style being modelled,]... proceeded to the formation of hypotheses sufficient to account for [the] structure [of these pieces], and then from these hypotheses to the automated generation of artificial pieces.’ He goes on to state that ‘the use of the computer as a generative device transformed analytical studies into a sort of experimental science in which computer output could be used to verify the hypotheses underlying an analysis.’²⁵ In fact, although such programs can never actually be used to ‘verify’ these hypotheses, they can sometimes be used to refute them but only if the hypotheses are unambiguous. For example, if one’s theory is that the set of all and only pieces that can be composed by a particular computer program is equal to the set of all and only pieces in some particular style, then it is necessary for the style being modelled to have been precisely specified as a well-defined set before the hypothesis becomes strictly refutable. Baroni goes so far as to claim that the activity of writing a composing program becomes ‘musicologically significant’ only ‘when applied to pastiche composition’²⁶ where the intention is to embody as an explicit algorithm the necessary and sufficient knowledge required for composition of all and only the pieces in some well-defined style.

²³ Brown and Dempster 1989, 83.

²⁴ Ebcioğlu 1987b, 126.

²⁵ Baroni 1983, 179.

²⁶ Baroni 1983, 179.

2 Chomsky and Schenker

2.1 *Most efforts to produce formal tonal style theories have derived much from the work of Chomsky and Schenker*

In the first chapter of his *Aspects of the Theory of Syntax* (Chomsky 1965, 3-62), Chomsky presents certain methodological assumptions and principles for generative linguistics that have formed the metatheoretical basis of research programmes in this field ever since. Indeed, even in 1965 they were considered to be well-established principles. A number of musicological research programmes that have resulted in the development of formal theories for musical styles have been based in one way or another upon the methodological principles and assumptions that Chomsky describes in the first chapter of *Aspects*. Unfortunately, there are a number of serious flaws in Chomsky's methodology as presented in Chomsky 1965, and these flaws have been transferred to some of the musicological research programmes that have been influenced by this methodology.

It was not long after the publication of Chomsky's first exposition of his theory of generative grammar (Chomsky 1957) that musicologists noticed the similarity in structure between Schenker's theory and that of generative grammar (see, for example, Kassler 1963). Both theories are hierarchical in nature and both are concerned with the generative definition of sets of structures that can loosely be termed 'languages.' Moreover, both generative linguistics and Schenkerian theory are concerned with characterizing both the uniqueness of single utterances or pieces of music and the way in which such individual utterances or pieces relate to other utterances or pieces in the language by means of hierarchical descriptions that exhaustively characterize their structures.

This similarity between Chomskyan generative grammar and Schenker's theory, combined with the fact that Chomsky's methodology seemed to have the robustness that Schenker's theory lacked, led a number of researchers to advocate the application of the methodology of Chomskyan generative grammar to the study of tonal music. As Lerdahl and Jackendoff remark, 'if there are significant parallels between Schenkerian theory and generative linguistics, it seems logical to ask what is required to convert Schenkerian theory into a formal theory.'²⁷

Indeed, by 1979 James Snell felt justified in claiming that

even at this early stage of 'systematic music theory' certain elementary propositions have become fairly well established. Most of the literature cites them in some form, and they tend to be assumed in discussions.

- (1) Schenker's work constitutes at least the single greatest resource for anyone wishing to build a formal theory of music.
- (2) Certain parallels exist between musical structure and linguistic structure. ...
- (3) It is possible to reformulate at least the more elementary concepts of Schenker's analytical theory as formal generative rules and to construct a grammar ... that generates levels analogous to Schenker's *Schichten*.²⁸

²⁷ Lerdahl and Jackendoff 1983, 111.

²⁸ Snell 1979, 3-4.

However, Snell's claim that even in 1979 they had 'become fairly well established' was, I think, somewhat over-optimistic. As I shall show below, even amongst those who agree that a satisfactory theory of tonal music would result from explicating and implementing the theory of Heinrich Schenker, there is considerable disagreement as to exactly what form the explicated theory should take.

Similarly, although a number of researchers have employed Chomsky's methodology of generative grammar in attempts to develop formal theories for musical styles, these researchers have not all employed this methodology in the same way, and there is no consensus as to which aspects of Chomsky's metatheory can appropriately be applied to the formal study of Western tonal musical styles. As Roads (1985) has noted, there is a 'running controversy over the appropriateness of applying certain linguistic concepts to music.'²⁹ Nonetheless, many formal theories of tonal music have been based upon the ideas of Schenker and many have been based upon the methodological principles of linguistics, particularly as formulated by Chomsky in the late 1950s and early 1960s. Moreover, there is a significant overlap between these two categories of theory.

2.2 *Introduction to application of linguistics to music theory*

That the methodology and technical apparatus of generative grammar can be fruitfully employed in fields other than linguistics is well recognized. The generative linguist, John Lyons, acknowledges that

the syntax of a formal language is describable without reference to any interpretation that might be assigned to the elements or combinations of elements; and a formal language might serve as a model, in principle, for all sorts of systems that have nothing to do with communication and would never be described as languages in the everyday sense of the term.³⁰

Roads states that 'the idea of viewing music in terms of a musical grammar is not new' and cites in support of this Powers' (1980, 49) description of 'a ninth-century Latin passage that portrays song forms through a linguistic analogy,' 'a number of sixteenth-century composition manuals [that] treated composition as a form of rhetorical expression' and a book by Busby (1818).³¹ He goes on to claim that there was a resurgence of interest in the use of formal grammars for music description during the 1970s and cites Baroni 1981 (translated in Baroni 1983) as a main source for this. According to Roads, this renewed interest was also in part due to the apparent successes achieved in the field of generative linguistics particularly since the publication of Chomsky's *Syntactic Structures* (Chomsky 1957).

In contrast, even as late as 1984, Peel and Slawson felt justified in claiming that 'the revolution in linguistics brought about largely by the work of Noam Chomsky, his students, and his critics has had only a slight effect on music theory.'³² This view is hard to understand, however, given that by the time that Peel and Slawson were writing this, there already existed a considerable number of published reports of projects that

²⁹ Roads 1985, 404.

³⁰ Lyons 1977, 169.

³¹ Roads 1985, 403.

³² Peel and Slawson 1984, 271.

had explicitly employed notions derived from generative grammar in the study of music.³³

Nonetheless, Peel and Slawson correctly note that a number of these studies ‘exhibited only a superficial knowledge of linguistics.’³⁴ In particular, Bernstein (1976) has been criticized by Jackendoff (1977) and Keiler (1978) for making simplistic, and rather contrived ‘concept-for-concept correspondences between categories in generative grammar theory and music’ and for ‘using the overtone series as a basis for musical phonology.’ Keiler was ‘likewise unconvinced by Bernstein’s general conclusions concerning serialism versus tonality.’³⁵

One of the most ambitious attempts to apply the methodology of generative linguistics to the study of tonal music is Lerdahl and Jackendoff’s *A Generative Theory of Tonal Music* (henceforth *GTTM*) (Lerdahl and Jackendoff 1983). Lerdahl and Jackendoff explicitly state that their work ‘is based on the methodologies and outlook of Chomskian linguistics’³⁶ and that in writing *GTTM* they were attempting ‘to achieve a synthesis of the outlook and methodology of contemporary linguistics with the insights of recent music theory.’³⁷ Their theory is certainly more in the spirit of Chomsky’s metatheory than many other theories that employ notions from generative grammar (cf. Cope’s (1991) EMI system).³⁸

However, perhaps in order to distance themselves from earlier attempts such as that of Bernstein, they are careful to emphasize that, in their view, ‘one should not approach music with any preconceptions that the substance of music theory will look at all like linguistic theory.’³⁹ Baroni and his collaborators echo this warning, saying that ‘there is no evidence to suggest that musical grammars should possess structures and properties identical to those of verbal grammars.’⁴⁰

Lerdahl and Jackendoff claim that it is ‘the combination of psychological concerns and the formal nature of [linguistic] theory’ that should be transferred to a theory of music.⁴¹ Unfortunately, as I will discuss in more depth below, in my opinion Lerdahl and Jackendoff transfer over to music many of the worst aspects of Chomsky’s metatheory such as the view that the study of competence is more important than the study of performance and can usefully be carried out in isolation from it. Also, there are certain formal aspects of linguistic theory—such as the use of non-terminal symbols and the employment of different rewrite rules at different levels of a derivation—that Lerdahl and Jackendoff do not use, but that I think could profitably be transferred to

³³ For example, to name just some of the major works that already existed at the time that Peel and Slawson were writing: Baroni and Jacoboni 1973, 1975, 1976, 1978, 1983; Baroni 1981, 1983; Bernstein 1976; Boretz 1969, 1970; Deutsch and Feroe 1981; Feld 1974; Holtzman 1980, 1981; Jackendoff and Lerdahl 1980, 1981, 1982; Kassler 1963, 1967, 1975; Keiler 1981a, 1981b; Laske 1973, 1980; Lerdahl and Jackendoff 1977; Lidov and Gabura 1973; Moorer 1972; Narmour 1977; Nattiez 1975; Powers 1980; Rader 1974; Roads 1979; Rothgeb 1968; Smoliar 1976, 1980; Snell 1979; Sundberg and Lindblom 1976; Winograd 1968; etc.)

³⁴ Peel and Slawson 1984, 271.

³⁵ Roads 1985, 423–4.

³⁶ Lerdahl and Jackendoff 1983, back cover.

³⁷ Lerdahl and Jackendoff 1983, xii.

³⁸ See particularly Cope 1991, Chapter 2, 27–70, for Cope’s employment of natural language processing techniques in his EMI system.

³⁹ Lerdahl and Jackendoff 1983, 5.

⁴⁰ Baroni et al. 1984, 202.

⁴¹ Lerdahl and Jackendoff 1983, 5.

music theory. Moreover, Chomsky's grammars were explicit and formal to the extent of being almost directly implementable as generator and parser computer programs whereas *GTTM* is neither complete nor explicit enough to be directly implemented as a computer program.

2.3 *Introduction to basic Chomskyan linguistic concepts*

In *Syntactic Structures*, Chomsky defines a *language* to be 'a set (finite or infinite) of sentences, each finite in length and constructed out of a finite set of elements'⁴² and goes on to state that

the fundamental aim in the linguistic analysis of a language L is to separate the *grammatical* sequences which are the sentences of L from the *ungrammatical* sequences which are not sentences of L and to study the structure of the grammatical sequences. The grammar of L will thus be a device that generates all of the grammatical sequences of L and none of the ungrammatical ones.⁴³

If the set of sentences generated by a grammar contains strings that are not grammatical sentences in the natural language being modelled then it is said to *overgenerate*. If the set of strings generated by the grammar does not contain certain grammatical sentences in the natural language being modelled then the grammar is said to *undergenerate*.

To this extent, the concept of a generative grammar in Chomsky's methodology corresponds to a theory for a musical style that aims to generatively specify the set of all and only those pieces in the style. Just as a grammar generates a set of sentences, so such a style theory would provide a generative definition of a set of pieces. Just as the set of sentences generated by a grammar is intended to be equal to a set that contains all and only the grammatical sentences in some natural language, so the aim when constructing such a style theory is to generate a set of pieces that is equal to the set of all and only those pieces in the style being studied. There is thus a basic correspondence between a Chomskyan generative grammar and a theory for a musical style that models the style as a set of pieces and aims to provide a generative definition of this set.

However, a closer study of Chomsky's metatheory soon reveals that a generative grammar is intended to be much more than merely a precise definition of a set of sentences. Given that an *ideal speaker-listener* is someone who lives

in a completely homogeneous speech-community, who knows its language perfectly and is unaffected by such grammatically irrelevant conditions as memory limitations, distractions, shifts of attention and interest, and errors (random or characteristic) in applying his knowledge of the language in actual performance,⁴⁴

and given also that *competence* is 'the speaker-hearer's knowledge of his language'⁴⁵ whereas *performance* is 'the actual use of language in concrete situations'⁴⁶ then a

grammar of a language purports to be a description of the ideal speaker-hearer's intrinsic competence. If the grammar is, furthermore, perfectly explicit—in other words, if it does not rely on the intelligence of the understanding reader but rather

⁴² Chomsky 1957, 13.

⁴³ Chomsky 1957, 13.

⁴⁴ Chomsky 1965, 3.

⁴⁵ Chomsky 1965, 4.

⁴⁶ Chomsky 1965, 4.

provides an explicit analysis of his contribution—we may (somewhat redundantly) call it a *generative grammar*.⁴⁷

A generative grammar must ‘attempt to characterize in the most neutral possible terms the knowledge of the language that provides the basis for actual use of language by a speaker-hearer’⁴⁸ and it must do this by assigning ‘to each of an infinite range of sentences a structural description indicating how this sentence is understood by the ideal speaker-hearer.’⁴⁹ It must thus be ‘a system of rules that in some explicit and well-defined way assigns structural descriptions to sentences.’⁵⁰

If a grammar assigns ‘correct’ structural descriptions to the sentences in a language then it is considered to be *descriptively adequate*. Thus, a grammar

is descriptively adequate to the extent that it correctly describes the intrinsic competence of the idealized native speaker. The structural descriptions assigned to sentences by the grammar, the distinctions that it makes between well-formed and deviant, and so on, must, for descriptive adequacy, correspond to the linguistic intuition of the native speaker (whether or not he may be immediately aware of this) in a substantial and significant class of crucial cases.⁵¹

To say that a grammar must be a device for assigning ‘correct’ structural descriptions to the sentences of a language is a much stronger requirement than to specify merely that it must generate all the *sentences* in a language. Chomsky distinguishes between these two requirements by saying ‘that a grammar *weakly generates* a set of sentences and that it *strongly generates* a set of structural descriptions.’⁵² Thus a grammar weakly generates a natural language if and only if it does not overgenerate and does not undergenerate whereas, as Chomsky states, ‘a grammar is descriptively adequate if it strongly generates the correct set of structural descriptions.’⁵³

Unfortunately, music theorists have not always maintained a clear distinction between Chomsky’s concepts of descriptive adequacy and weak generation. For example, in discussing what he calls ‘Adequacy and the “all-or-only” issue,’ James Snell claims that there is a ‘pair of criteria of *descriptive* adequacy, concerning not the structure of the generative model, but only its output’ and that these criteria are, first, ‘does [the grammar] generate *all* of the instances of the corpus [i.e. style or natural language]’ and second, ‘does it generate *only* these?’ In fact, as should be clear from the definitions given above, whether or not a grammar generates all and only the sentences of some natural language determines only whether or not it *weakly generates* the language. In order to be descriptively adequate, the grammar must also generate for each sentence in the language a ‘correct structural description.’

Snell goes on to state that if a grammar generates all and only the sentences of some language ‘then the model can be said to “define” the phenomenon, and, in a certain weak sense, to “account for” it’ but that, ‘strictly speaking, these criteria together constitute a necessary, although not sufficient, condition for an adequate

⁴⁷ Chomsky 1965, 4.

⁴⁸ Chomsky 1965, 9.

⁴⁹ Chomsky 1965, 4–5.

⁵⁰ Chomsky 1965, 8.

⁵¹ Chomsky 1965, 24

⁵² Chomsky 1965, 60.

⁵³ Chomsky 1965, 60.

theory.’⁵⁴ I certainly agree that if a grammar weakly generates the natural language that it is intended to account for then it ‘defines’ and ‘accounts for’ that language. But in my view, a theory for a tonal musical style can and need aim to be *no more than* a hypothesis that some generatively specified set of scores is equal to the set of all and only those scores in the style of some specified corpus, because although there may be certain aspects of the mental processes involved in musical composition that are effectively ‘known,’ (such as, for example, that a composer knows nothing about pieces that were composed after he died), most of the details of the mental processes by which pieces of music come into existence are (and probably will remain) inaccessible to empirical observation. Thus whereas Snell (and Chomsky) hold the view that weak generation is only a *necessary* condition on the adequacy of a grammar, my own view is that it is a *sufficient* condition because, in practice, while it is certainly possible to test a grammar for weak generation, it is *not* possible to test a grammar for strong generation.

Just as Snell confuses Chomsky’s notions of descriptive adequacy and weak generation, so he confuses the notions of explanatory and descriptive adequacy, claiming that a grammar achieves explanatory adequacy to the extent that it ‘assigns correct structural descriptions for instances not used as model cases in its formulation’ that, first, ‘are similar for instances of the phenomenon that are believed to be underlyingly similar’ and second, ‘reflect known ambiguities in certain cases.’⁵⁵ In fact, as explained above, these are Chomsky’s criteria for *descriptive* adequacy. As will be explained below, ‘explanatory adequacy’ in Chomsky’s sense, is something that can be achieved only by a *linguistic theory*, not by a grammar for a particular language. A linguistic theory achieves explanatory adequacy if it generates exactly one descriptively adequate grammar for each humanly possible natural language.

Even the basic terms ‘generate’ and ‘generative’, as used, for example, in the expression ‘generative grammar,’ have been misconstrued by musicologists. When one states that an algorithm or grammar ‘generates’ a set of sentences, the term is being used in its mathematical sense to mean that the algorithm or grammar ‘provides a precise criterion or specification for membership in (a set).’⁵⁶

Baroni, however, claims that the difference between what he calls a ‘taxonomic’ analysis and a generative one is that ‘the former observes musical events and divides them into categories defined by particular concepts’ whereas ‘the latter observes some types of regularity present in the organisation of musical events and formulates hypotheses about the processes of thinking inherent in this organisation.’ In other words, Baroni claims that whereas a taxonomic analysis ‘has to do with observable phenomena,’ a generative one makes ‘hypotheses about events which are not directly observable.’⁵⁷

For Chomsky, however, the fact that a theory ‘formulates hypotheses about processes of thinking’ implies that it is a theory of *competence* not that it is a ‘generative’ theory. It is not this aspect of a ‘generative grammar’ that makes such a grammar ‘generative.’ What makes a generative grammar ‘generative’ is the fact that it precisely specifies a (possibly infinite) set of sentences by means of a finite set of rules. Indeed, the only way to define an infinite set is ‘generatively’ (or ‘recursively’)—that is, in terms of a finite number of sufficient conditions satisfied by all and only members

⁵⁴ Snell 1979, 57.

⁵⁵ Snell 1979, 57.

⁵⁶ Sinclair et al. 1991, entry for ‘generate’, sense 4.

⁵⁷ Baroni, Dalmonte and Jacoboni 1992, 188.

of the set. Thus Baroni is correct in claiming that ‘a generative analysis has the advantage of being economical, because a limited number of rules is able to describe an infinite number of events,’ which in the case of a theory for a melodic style consists of ‘the repertoire [i.e. corpus] and all other melodies composed or composable in the same style.’⁵⁸

What Baroni calls a ‘taxonomic’ analysis closely resembles the type of theory that Chomsky accused certain so-called ‘behaviourists’ of trying to formulate.⁵⁹ As I shall discuss below, Chomsky seems to have been of the opinion that because such theories tended to be ad hoc there was no point in trying to develop them. Just as Chomsky claims that this ‘taxonomic’ attitude ‘expresses itself in the proposal to limit the term “theory” to “summary of data”,’ so Baroni believes that theorists should not ‘limit themselves to the observation of regularities.’⁶⁰ Rather, if they wish to ‘arrive at a true grammar’ they must formulate ‘hypotheses about the structural grounds for regularities.’ This seems to imply that he believes that a grammar must ‘explain’ regularities by presenting them as logical consequences of some plausible ‘fact’ about mental processes. Thus he claims that

we could easily observe that the second and [fourth] phrases of Lutheran Chorales often have descending profiles, but we would arrive at a grammar only if we were to translate this observation into a rule such as: ‘Every chorale is divided into periods of at least two phrases; every period is concluded by descending movements’.⁶¹

Now, in fact, if ‘the second and [fourth] phrases of Lutheran Chorales’ only *often* ‘have descending profiles’ then the rule that he suggests is normative and is not actually true. On the other hand, the observation that they *often* have descending profiles may well be true. The rule is certainly more formal than the observation but it is not a very good explication of, or generalization from, the observation. The rule demands that one sees the second and fourth phrases of a chorale as being the answering halves of ‘periods,’ whereas the observation treats them in isolation. So the rule is more theory-laden and represents an attempt to present a more coherent and generalized view of the phenomenon.

From this example it would seem that in order for an ‘observation of a regularity’ to be raised to the status of a rule that is qualified to become part of a ‘true grammar,’ Baroni would demand either that it achieve some minimum degree of formality or that it be related to more general principles. I agree that a grammar should consist only of rules that are formal enough to be implemented algorithmically. However, I disagree that one needs to stipulate as an a priori constraint on the structure of a grammar that its rules need to be related to more general principles. This may well make the theory easier to understand and more coherent but I believe this should not be done at the expense of producing a normative theory that is incapable of accounting for some of the data that one has set out to account for. In other words, I believe that over-concern with avoiding ‘ad hocness’ in a theory, or equivalently, with ‘keep[ing] the rules as few and as simple as possible,’⁶² can lead to a normative theory that does not account for the facts that one is aiming to explain. Also, because of the structural complexity of

⁵⁸ Baroni, Dalmonte and Jacoboni 1992, 188.

⁵⁹ See Chomsky’s critique of Twaddell 1935 (Chomsky 1965, 193–4, note 1.)

⁶⁰ Baroni et al. 1984, 203.

⁶¹ Baroni et al. 1984, 203.

⁶² Steedman 1984, 73.

language and tonal music, I believe it is extremely unlikely that one could ever successfully characterize a *natural* language or the musical style of a *real* composer with a very ad hoc theory—in general, the more ad hoc a theory, the easier it is to refute with new positive evidence. I shall discuss this issue in more depth later.

2.4 *Explicating Schenker's theory*

The problem of implementing and explicating Schenker's theory as a working computer program is a challenging one that has been addressed by a number of researchers over the last thirty years or so including Kassler (1963, 1975), Smoliar (1980), Snell (1979), and Ebcioğlu (1987b). Each of these researchers has as a result of his attempt to explicate Schenker's theory succeeded in developing a new and more explicit theory of his own for the broad tonal style that is usually termed 'Western classical tonal music' and that is generally understood to include most of the pieces composed in Europe between about 1650 and 1900.

Kassler proposes that music theorists should concern themselves with the problem of developing 'intelligent music-processing machines' that would

be able to carry out such ... processes as language identification (is a presented composition an instance of tonality, the twelve-note system, or some other well-known musical language?), structural analysis (within an identified musical language, what syntactic relationships do the composition's notes, rests, chords, phrases, etc. possess?), and composition of coherent new utterances (within a particular musical language, and even within a particular musical 'style' that is a dialect of such a language).⁶³

Kassler suggests that

to endow a machine with such 'intelligence' it seems wise, as in other areas of artificial intelligence research, first to give the machine a program embodying the best of hitherto existing theories of the subject matter.⁶⁴

Kassler's decision to attempt to explicate the theory presented in Schenker's *Der freie Satz* was therefore motivated by the view that it was one of the 'best hitherto existing theories' of tonal music. I certainly agree that one should as far as possible build on existing knowledge and avoid inventing a completely new theory from scratch.

Unfortunately, in music, theories have been notoriously informal and inconsistent so producing explications of them that are precise enough to be computationally implementable is no small task in itself. Ebcioğlu goes so far as to claim that 'Schenker's rules do not [even] meet the level of precision typically found in a traditional treatise'⁶⁵ on harmony and counterpoint; and that although Schenker

was able to verbally describe the different ingredients that make up a series of legal analytic graphs that represent the deep voice leading structure of a musical piece, [he] was unable to provide any precise absolute rules that indicate which analytic graphs are unacceptable for a given piece, or heuristics that indicate which analytic graphs are preferred.⁶⁶

Therefore, as Kassler notes,

⁶³ Kassler 1975, 2–3.

⁶⁴ Kassler 1975, 3.

⁶⁵ Ebcioğlu 1987b, 82.

⁶⁶ Ebcioğlu 1987b, 88.

since the theory as published by Schenker is informal and in places unclear, ambiguous, or inconsistent ... [his] initial task [was] to explicate this theory, i.e., to replace Schenker's descriptions and examples of it by a structure which, though recognizably similar, has sufficient formalization that its constituent processes can be manipulated by computer.⁶⁷

The task of explicating Schenker's theory thus rests on the view, noted by Snell, that although Schenker's theory is 'informal and in many places inexplicit ... no theoretical barrier exists that would prevent one from rendering the great bulk of his theory fully explicit and formal.'⁶⁸

Snell complains, however that

if, as Kassler claims, his goal is to explicate the intuitive notion of tonality, he has chosen a strategy which is not likely to succeed, namely, restricting himself to testing formalized replicas of existing theories whose adequacy is determinable more directly,

and goes on to suggest that

a music theorist having the stated goal would do better to begin by developing a theory—perhaps a modified version of an earlier theory—whose adequacy he was unable to determine by direct inspection.⁶⁹

It is not clear how one is to distinguish between a 'formalized replica of an existing theory' and a 'modified version of an earlier theory,' but it is certainly true that Schenker's theory is not scientifically testable without a considerable degree of explication. In my view, implementability as a computer program is a good measure of whether or not a theory has been sufficiently explicated. Of course, an explication of Schenker's theory is a *different* theory from Schenker's own. A computer program could thus only ever be an implementation of a particular programmer's explication of Schenker's theory. It could never be an implementation of Schenker's theory itself.

Although Kassler's idea of what would constitute a satisfactory theory of tonal music coincides largely with my own, I think he attaches too much importance to characterizing the tonal style in general and too little to the need to be able to generate definable and recognizable particular tonal styles such as those of particular composers in particular genres. I think it is probably because of this that he adopts a research strategy that involves trying in the first instance to produce a computational implementation of a *general* theory of tonality such as Schenker's and Kollmann's, rather than inventing *particular* theories for the styles of music in single genres by single composers.

Snell similarly complains that Schenker's attitude of focusing on gaining a deeper understanding of *individual* pieces and composers 'is not the prevailing one among those who are presently trying to render tonal music theory more scientific' and that Kassler, Smoliar and Lerdahl and Jackendoff

fail to address very directly what [Snell conceives] to be the main task of music theory: extending concepts and techniques and finding new ones that will allow more penetrating musical understanding of particular pieces and composers.⁷⁰

⁶⁷ Kassler 1975, 4.

⁶⁸ Snell 1979, 16.

⁶⁹ Snell 1979, 13–14.

⁷⁰ Snell 1979, 6.

Snell therefore considers Kassler's goal of 'constructing systems for establishing by formal means whether [Schenker's theory is] adequate to generate the class of tonal compositions' to be 'limited'⁷¹ and claims that Kassler's explication is therefore a 'trivialization of Schenker' because Kassler interprets Schenker's theory as an attempt 'to account for tonality by specifying 'prolongation techniques' such that any composition that is an instance of tonality—but no other composition—can be derived by successive application of these techniques to an *Ursatz*.'⁷² However, while Schenker's aim may well not have been explicitly to account for tonality in this manner, it cannot be denied that his theory does serve as a (somewhat imprecise) generative specification of a set of pieces and it can readily be construed that an implicit hypothesis of Schenker's theory is that the set of pieces generated by it is equivalent to the universal set of tonal pieces—that is, a piece is a tonal piece if and only if it is explainable using his theory. Thus I agree with Lerdahl and Jackendoff that although 'the chief purpose of his theory was to illuminate structure in musical masterpieces,' Schenker 'can be construed (especially in *Der freie Satz*) as having developed a proto-generative theory of tonal music—that is, as having postulated a limited set of principles capable of recursively generating a potentially infinite set of tonal pieces.'⁷³

Thus while Snell is willing to admit that Kassler 'presents well-founded arguments for a more scientific approach to music theory,' he attacks him for being 'limited to constructing systems for establishing by formal means whether ... [Schenker's theory is] adequate to generate the class of tonal compositions' claiming that 'this aim is inappropriate' and that 'the adequacy of the elementary prolongation operations [of Schenker's theory] to generate any tonal piece is beyond controversy.'⁷⁴ This is tantamount to claiming that the set of pieces generatively specified by Schenker's theory contains without any doubt all tonal pieces. In my view, for this to be the case, Schenker's theory would have to allow *all possible pieces of music*. This might actually be the case, but I do not know of any proof of this. In any case, it cannot surely be 'beyond controversy' that Schenker's theory generates a set that contains all tonal pieces. Even if this were the case, there would still be a point in explicating and implementing it in order to test for overgeneration.

Lerdahl and Jackendoff also originally considered studying 'Schenkerian analyses of actual tonal pieces—pieces that are intrinsically interesting as well as sufficiently complex to be informative about cognitive processes' with a view to developing 'a rule system capable of generating these analyses.' 'This approach,' they claim, 'would reveal the lacunae in Schenkerian theory, and, generally, would put the theory on a solid intellectual foundation.' They state that they eventually decided against this approach, however, because 'it seemed a doubtful strategy to launch a theory of musical cognition by filling in the gaps in somebody else's "artistic" theory' and because it was not clear to them 'how such an approach would address cognitive issues.'⁷⁵

However, it seems to me that to adopt the strategy of attempting to build on the most successful existing theories in any particular field is perfectly justified. In science one generally builds on existing theories unless they are so poor that they have nothing at all to offer. In any case, Lerdahl and Jackendoff's (1983) theory (particularly the

⁷¹ Snell 1979, 9.

⁷² Snell 1979, 9–10.

⁷³ Lerdahl and Jackendoff 1983, 337–8, note 1.

⁷⁴ Snell 1979, 9.

⁷⁵ Lerdahl and Jackendoff 1983, 111–112.

prolongational reduction component) is in fact so heavily based upon Schenker's theory that it could justifiably be termed an 'explication' of it. In particular, Lerdahl and Jackendoff's 'basic form' is essentially identical to Schenker's *Ursatz* and serves an essentially identical purpose in their theory. At one point, for example, they present a diagram of the *Ursatz* in the form of a tree structure and claim that it 'represents the prolongational *basic form* for a typical tonal piece.'⁷⁶ Moreover, they claim that their 'basic form' 'is grammatically complete in the sense that it expresses both tonic prolongation and cadential resolution' and that 'this form appears not only for whole pieces but—when considered in isolation—also for subordinate grouping levels such as theme groups, periods, and phrases.' Finally, to complete the parallel with Schenker's *Ursatz*, they state that 'much of the unity of [pieces of] tonal music depends on' their being reducible to the basic form.⁷⁷ Later in this thesis I shall argue against the view that global structural constraints such as Schenker's *Ursatz* and Lerdahl and Jackendoff's 'basic form' and 'Strong Reduction Hypothesis' must be satisfied by tonal pieces if they are to be acceptable.

Lerdahl and Jackendoff's theory thus derives a great deal from Schenker's. Indeed, the only reason why one might not be justified in considering it an explication of Schenker's theory is that it is not explicit and formal enough to be considered an 'explication' at all. Lerdahl and Jackendoff, unlike Kessler, Smoliar, Snell and Ebcioğlu, were 'not concerned whether or not [their] theory can readily be converted into a computer program.'⁷⁸ Indeed, Peel and Slawson go so far as to claim that although 'Schenker did not use the language of linguistic theory ... nevertheless, his rules are clear and considerably stronger in their assertions than those of Lerdahl and Jackendoff.'⁷⁹ As I shall discuss below, I believe that Lerdahl and Jackendoff could profitably have devoted more care to making their theory more explicit and formal. However, I do not think that one can reasonably claim that their theory is *less* formal than that of Schenker.

In general, Peel and Slawson's criticisms of Lerdahl and Jackendoff's theory seem to be based upon a fundamental misinterpretation of its nature and purpose. Peel and Slawson claim that 'although [Lerdahl and Jackendoff's] well-formedness rules eliminate certain kinds of non-hierarchical readings, the admission of preference rules to the theory without some weighting method that would guide their application drastically weakens the theory.' In particular they compare Lerdahl and Jackendoff's analysis of Bach's chorale 'O Haupt voll Blut und Wunden' (BWV 244/44)⁸⁰ with an alternative and rather different reading that they claim is also allowed by Lerdahl and Jackendoff's rules and conclude that while they

realize that any method of analysis is likely to permit alternate readings of passages—as, for example, does the method of Heinrich Schenker ... here the difference between the two readings is so striking, ... that [they] cannot help but conclude that Lerdahl and Jackendoff's theory constrains analytic conceptions very little indeed.⁸¹

⁷⁶ Lerdahl and Jackendoff 1983, 189.

⁷⁷ Lerdahl and Jackendoff 1983, 189.

⁷⁸ Lerdahl and Jackendoff 1983, 332, note 6.

⁷⁹ Peel and Slawson 1984, 287.

⁸⁰ Lerdahl and Jackendoff 1983, 107 ff.

⁸¹ Peel and Slawson 1984, 282.

Similarly, in comparing Schenker's analysis of the first section of Mozart's Piano Sonata K.331⁸² with that of Lerdahl and Jackendoff, Peel and Slawson claim that 'Schenker's "alternative reading" actually represents a strong criticism of Lerdahl and Jackendoff's theory.'⁸³

All of these criticisms seem to be based upon the erroneous idea that Lerdahl and Jackendoff's theory was intended to be a 'method of analysis.' In fact, Lerdahl and Jackendoff's theory was intended to be a *theory of perception* that predicts how a certain (admittedly, poorly defined) class of listener will interpret a certain (poorly defined) class of compositions. As such, it is perfectly acceptable that their theory should generate more than one structural description for any given piece. The fact that it generates in general a number of possible structural descriptions and not a *single* structural description for any given piece merely reflects the fact—well confirmed by Peel and Slawson themselves—that different individuals can achieve widely divergent but equally valid interpretations of any given piece of music.

Peel and Slawson's claim that Schenker's theory generates for any given piece a much more constrained set of analyses than that generated by Lerdahl and Jackendoff's theory is probably wrong. But even if this were indeed the case, it would not imply that Schenker had come closer to achieving Lerdahl and Jackendoff's aims than did Lerdahl and Jackendoff themselves. Indeed, if Peel and Slawson's claim were correct, it would, if anything, militate *against* Schenker's theory being a plausible *theory of perception*. Consequently, the fact that Schenker's analyses are different from those of Lerdahl and Jackendoff does not constitute an indictment of Lerdahl and Jackendoff's theory. As Lerdahl and Jackendoff themselves say, 'if [their] results turn out like Schenkerian analyses, fine; if not, that too is interesting,'⁸⁴ but it does not constitute a refutation of their theory.

Ebcioğlu makes a parallel comment with respect to analysts' misinterpretation of the significance and epistemological status of Schenker's concept of linear progression:

The intrinsic importance of deep linear progressions is simply a new way of hearing, which should be learned, appreciated, and added to the existing intuitions of the educated musician. We feel that it is irrelevant to attack Schenker because his parsings do not follow existing down-to-earth intuitions, ... or to defend Schenker because his parsings do follow existing down-to-earth intuitions ... (incidentally, they often do).⁸⁵

In his review of *GTTM*, Harvey remarks with reference to Peel and Slawson's (1984) criticism that 'certain piqued Schenkerians, alarmed at some of the book's claims, have already reacted strongly' but goes on to write that he 'can agree with William Drabkin (1984) in reading the theory as ultimately supportive of and even compatible with Schenker, rather than as a more or less thinly disguised attack.'⁸⁶ My main point here, however, is that one cannot justifiably criticize Lerdahl and Jackendoff's theory for generating analyses that are not in accord with Schenkerian graphs. The purpose of their theory is to 'explicate the intuitions of the expert listener.' Now Schenker was without a doubt an 'expert listener,' but his expertise became more and more informed by his own theoretical convictions. To some extent, therefore, his

⁸² Lerdahl and Jackendoff 1983, 32 ff.

⁸³ Peel and Slawson 1984, 287.

⁸⁴ Lerdahl and Jackendoff 1983, 112.

⁸⁵ Ebcioğlu 1987b, 121.

⁸⁶ Harvey 1985, 296.

graphs become less and less a documentation of his intuitions as an expert listener and more and more evidence in support of his own theoretical convictions. That is, his interpretations became less and less the product of *unconscious intuition* (which is what Lerdahl and Jackendoff were trying to model) and more and more the deliberate constructions of an *explicit theoretical perspective*—which is certainly *not* what Lerdahl and Jackendoff were trying to model.

In any case, it seems very strange to me to decide to aim to describe how listeners *actually* interpret tonal music rather than to try to provide the means by which richer interpretations may be achieved. Indeed, one would have thought that part of the purpose of a theory of tonal music would be to provide a means for listeners to achieve interpretations that are as insightful, detailed and complete as possible of as many tonal pieces as possible. So it would hardly seem sensible to judge the ‘correctness’ of the structural descriptions generated by a theory in terms of how well they correspond to real listeners’ interpretations of pieces. This would be tantamount to making the a priori assumption that the more ‘intuitive’ the explanations provided by a theory for a class of phenomena, the more likely that theory is to be correct—an assumption that can hardly be made in the light of the past century of scientific history.

In fact, of course, Lerdahl and Jackendoff’s theory aims to do much more than merely *describe* the intuitions of listeners experienced in the idiom of Western tonal music. Their theory is in fact a system that takes a musical surface as input and aims to generate as output for it a partial description (or, more precisely, a *set* of partial descriptions) of the hierarchical aspects of the interpretation that an *idealized* listener would hypothetically achieve of the piece. In other words, their theory aims to generate for each tonal piece a set of *best possible* structural descriptions for it. That Lerdahl and Jackendoff are certainly not content with attempting merely to describe the intuitions of *real* listeners is clear from their need to introduce the concept of a ‘perfect listener.’

In dealing with especially complex artistic issues, we will sometimes elevate the experienced listener to the status of a ‘perfect’ listener—that privileged being whom the great composers and theorists presumably aspire to address.⁸⁷

⁸⁷ Lerdahl and Jackendoff 1983, 3.

3 The concept of an algorithmic style theory

3.1 Introduction to the concept of an algorithmic style theory

One of my main purposes in this thesis is to propose that if a theory for a musical style is not to suffer from the metatheoretical problems that can be identified in certain existing theories for musical styles then it should take the form of what I call an *algorithmic style theory*. An algorithmic style theory is a hypothesis that a well-defined set of scores called a *style* is equal to another well-defined set of scores called the *universal set of well-formed scores* of the theory. The *style* of the theory must be defined to contain all and only those scores that are either members of a well-defined *corpus* of existing scores or determined to be ‘in the style of this corpus’ according to an *acceptability algorithm*, which is an empirical, experimental test along the lines of a Turing test (Turing 1950). The *universal set of well-formed scores* of the theory must be defined to be the universal output set of an algorithm that automatically composes pieces of music from scratch with no run-time input other than perhaps a sequence of random or pseudo-random numbers. The universal output set of an algorithm is the set of all and only possible outputs that it can generate. The algorithm that generates the universal set of well-formed scores of an algorithmic style theory must output one member of its universal output set at random on each execution. In fact, this algorithm must consist of two algorithms, the output of the first being passed as input to the second which then produces a score as output. I call the first of these two algorithms the *composing algorithm*, and the second, the *score algorithm* of the algorithmic style theory. The composing algorithm does not actually generate scores as output, rather it generates *representations* of scores which must then be converted by the score algorithm into actual scores. The reason for this complication will be explained below.

In the remainder of this chapter, I shall flesh out this definition and discuss aspects of it in more detail. However, it should already be clear that in order to completely define an algorithmic style theory for a style, the theorist must define a number of objects associated with the theory—for example, the *style*, the *corpus*, the *acceptability algorithm*, the *composing algorithm* and the *score algorithm*. In fact, in order to fully specify and test an algorithmic style theory—that is, a hypothesis that a style (defined by means of a corpus and an acceptability algorithm) is equal to a universal set of well-formed scores (defined by means of a composing algorithm)—the theorist must define eight objects that together form what I call an *algorithmic style theory system*. An object T is an *algorithmic style theory system* if and only if it is an 8-tuple as follows:

$$\langle \underline{s}_u(T), \underline{s}_k(T), \alpha(T), \rho(T), \gamma(T), \sigma(T), \delta(T), \pi(T) \rangle$$

where:

- $\underline{s}_u(T)$ is the *universal set of scores* of T ;
- $\underline{s}_k(T)$ is the *corpus kernel* of T ;
- $\alpha(T)$ is the *acceptability algorithm* of T ;
- $\rho(T)$ is the *representation algorithm* of T ;
- $\gamma(T)$ is the *composing algorithm* of T ;
- $\sigma(T)$ is the *score algorithm* of T ;
- $\delta(T)$ is the *derivation algorithm* of T ;
- $\pi(T)$ is the *parsing algorithm* of T .

Each of these components will be defined and discussed in more detail below. For convenience, I may abbreviate the term ‘algorithmic style theory system’ to simply ‘style theory system’ or even ‘theory system’ on those occasions when no ambiguity would arise.

3.2 *Definition of the concept of an algorithm and input and output sets of an algorithm*

In this thesis I use the term *algorithm* in two senses. In the first sense, an *algorithm* is a precisely defined sequence of instructions forming a step-by-step procedure which can be carried out by a human in a finite length of time and which is sufficiently unambiguous for there never to be any question as to whether or not the procedure has been carried out correctly. In the second sense, an *algorithm* is a sequence of instructions that is sufficiently explicit to be implemented as an equivalent computer program. It will be clear from the context which sense is implied in any given case.

The *output set* of an algorithm for a specified *input set* is the set of all and only those outputs that can be produced when the input is a member of the specified input set. The *input set* of an algorithm for a specified *output set* is the set that contains all and only those inputs that generate outputs that are members of the specified output set. The *universal input set* of an algorithm is the set which contains all and only those objects that the algorithm can possibly accept as input. The *universal output set* of an algorithm is the set that contains all and only those objects that the algorithm can possibly generate as output. The universal output set of an algorithm is therefore the output set of the algorithm for the universal input set.

3.3 *Definition of the concept of a universal set of scores*

The *universal set of scores* of an algorithmic style theory system must be a set that is defined to contain all and only those representations of music that satisfy certain specified criteria. For example, it could be defined to be the set of all and only digital recordings, or the set of all and only German lute tablatures and so on. The universal set of scores of an algorithmic style theory system T will be denoted $\underline{s}_u(T)$.

Unless otherwise stated the reader should assume from this point on that the *universal set of scores* of any algorithmic style theory system is the set that contains all and only physical or possible representations of music in standard, modern, Western, staff notation (henceforth SN or ‘Standard Notation’). The logical structure of an algorithmic style theory system whose universal set of scores is defined in this way is the same as that of any other algorithmic style theory system, so nothing is lost by making this assumption. From this point on the reader may assume therefore that an object is a *score* if and only if it is a member of this universal set of scores. If it is a score that actually exists then it is a *physical score*; if it is a score that could exist in principle, but does not actually exist then it is an *abstract score*.

Figure 3-1 shows four physical scores. Two scores are *physically distinct* if and only if they are different physical scores or if one is a physical score and one is an abstract score. All physical scores are physically distinct and all abstract scores are physically distinct from all physical scores but two abstract scores cannot be physically distinct because they do not physically exist. For example, all four scores in Figure 3-1 are physically distinct.



Figure 3-1

Although all physical scores are physically distinct from each other, some physical scores are *notationally equivalent*. Two scores are *notationally equivalent* if and only if they consist of exactly similar symbols in exactly the same logical organization. If two scores are not notationally equivalent then they are *notationally distinct*. For example, in Figure 3-1, score *A* is notationally equivalent to score *B* and notationally distinct from scores *C* and *D*. Each member of the universal set of scores is defined to be notationally equivalent to itself.

Although two abstract scores cannot be physically distinct, *all* abstract scores are notationally distinct. The set of all possible correct performances of a score is exactly equal to the set of all possible correct performances of any other score that is notationally equivalent to it. However, it is also possible for the set of all possible correct performances of a score to be the same as that of another score that is notationally *distinct* from it. For example, if scores *A* and *D* in Figure 3-1 are assumed to be for the same class of keyboard instruments then the set of all possible correct performances of *A* would be equal to the set of all possible correct performances of *D*.

To provide a less artificial example, the reader is asked to imagine that he or she has in front of him or her two copies of Klaus Schubert's edition of the 371 Bach chorales (Bach 1990) and one copy of Riemenschneider's edition (Bach 1941). On page

7 of each copy of Schubert's edition and on page 1 of Riemenschneider's the reader would find a physical score of the chorale 'Aus meines Herzens Grunde' (chorale no.1, BWV 269). These three physical scores are physically distinct but notationally equivalent. On page 13 of each copy of Schubert's edition and on page 5 of the copy of Riemenschneider's there is a physical score of the chorale 'Christ lag in Todesbanden' (chorale no.15, BWV 277). Again, all three of these scores are physically distinct. But while the two physical scores of this chorale in the copies of Schubert's edition are notationally equivalent, they are both notationally *distinct* from the score of this chorale in Riemenschneider's edition which has a crotchet rest missing from its fourth complete bar.

The universal set of scores is thus partitioned exhaustively and exclusively into *notational equivalence classes* of scores where the *notational equivalence class* to which a given score belongs is the set that contains all and only those scores that are notationally equivalent to it.

3.4 *The need to define precisely the class of objects to which the predictions of a theory apply*

For any theory it is crucial to be absolutely clear as to the class of phenomena to which the theory is intended to apply before one attempts to use or test the theory. For example, consider the hypothesis:

All apples are green

One could not use this hypothesis to predict the colour of an orange, nor would the discovery of an orange-coloured fruit that was not an apple refute or corroborate the theory because the theory is manifestly not intended to apply to anything other than apples. In this example, the class of phenomena to which the theory is intended to apply appears to be clearly defined: the set of all and only apples. This set is clearly defined only because, given any phenomenon, there will almost invariably be a universal consensus as to whether or not it is an apple. This green apple theory can be in one of only two states: refuted and unrefuted. It could never be verified because one could never be sure that there did not exist an undiscovered apple that was not green. If such a non-green apple were discovered, the theory would become refuted. Until such a time, the theory would remain unrefuted. Because of the universal consensus as to the set of phenomena to which the theory applies, there would always be a universal consensus as to the state of the theory at any particular time. I think this is a desirable feature of this theory.

Unfortunately, this feature is not present in some recent theories of tonal music. For example, Lerdahl and Jackendoff state that their theory of interpretation is applicable to 'classical Western tonal music'⁸⁸ but at no point do they provide a precise definition of what they mean by 'classical Western tonal music.' In other words, they never provide a precisely defined criterion for deciding for any given piece of music whether or not it is one that their theory is intended to be able to account for and, unfortunately, there is no universal consensus among musicologists as to the precise meaning of the expression 'classical Western tonal music.' There is therefore a strong possibility of two musicologists disagreeing as to whether or not a given piece is one that Lerdahl and Jackendoff's theory should be able to account for, and consequently, a

⁸⁸ Lerdahl and Jackendoff 1983, 4.

strong possibility that there will not in general be a universal consensus as to whether or not Lerdahl and Jackendoff's theory has been refuted at any given time. I consider this to be a very undesirable feature of their theory and I think it arises primarily from the fact that Lerdahl and Jackendoff have not adequately defined the set of pieces that their theory is intended to be able to account for. As Popper has pointed out, 'we can demand that anyone who advocates the empirical-scientific character of a theory must be able to specify under what conditions he would be prepared to regard it as falsified.'⁸⁹

Snell complains when discussing the manner in which Kassler's explication of Kollmann's theory accounts for a particular passage of Bach that 'there are several objections to be raised' but confines himself to one: '*the passage is plainly not an instance of tonal music*'⁹⁰ (my italics). Such a claim is clearly untenable. There are no unanimously agreed criteria for deciding whether or not a piece is 'tonal,' therefore whether or not a passage 'is tonal' depends entirely upon what one means by 'tonal' in any given instance and because there is no universally accepted meaning attributed to the term, whatever one actually means on any given occasion must be defined. Thus it may well be that the passage is 'plainly not tonal' according to the tacit definition of 'tonal' that Snell happens to be unconsciously employing in this particular case. But for his criticism to be valid, he would need at least to explicitly state this definition. The fact that he can so categorically state that the passage in question is 'plainly not tonal' suggests that perhaps his explication of the distinction between a 'tonal' and a 'non-tonal' passage of music would in fact be quite sharp. But what is important in this context is not whether or not the piece is tonal *according to Snell* but whether or not it is tonal according to Kassler or Kollmann, since it is their theory that is under discussion. Unfortunately, neither Kassler nor Kollmann give a clear definition of what they mean by 'tonal,' consequently it is impossible to decide whether or not the passage in question is one that Kollmann's theory should be able to account for. This again highlights the fact that when attempting to develop a theory for a musical style, it is necessary to provide a totally objective and precise definition of that style so that for any given piece of music there can be no question as to whether or not the theory should be able to account for it.

Lidov and Gabura claim that their grammar

is designed to capture some extremely general features of style which determine possible ground rules for melody in both the so-called 'period of common practice' for art music (say Haydn's lifetime) and of folk tunes and popular music of some time earlier and since.⁹¹

They go on to admit that their aims are 'too general to be stylistically specific.' But how is one to decide whether or not a grammar is overgenerating or undergenerating if one hasn't specified the criteria by which one is to decide for any given sequence whether or not it is a member of the natural language that one is attempting to characterize?

Steedman correctly remarks that a grammar of jazz chord sequences 'must do more than generate all the sequences of some "language." In particular, it should generate *only* those sequences.' But he goes on to complain that 'it is not easy to show that the rules never generate anything that is not a potential 12-bar, *especially since it is*

⁸⁹ Popper 1983, p.xxi.

⁹⁰ Snell 1979, 13.

⁹¹ Lidov and Gabura 1973, 138.

not quite clear what that class includes' (my italics).⁹² Of course, the class of chord sequences that would need to be generated by any feasible grammar for the style of the jazz 12-bar would be far too large for one ever to be able to 'show that the rules never generate anything that is not a potential 12-bar.' But if one is 'not clear what that class includes' then one cannot categorically decide for *any* sequence whether or not it is one that the grammar should be able to generate and one is left uttering meaningless value judgements such as that although certain chord sequences 'may not be very *good* 12-bars ... they do seem to *be* 12-bars, albeit of a rather fringe variety.'⁹³ This highlights the fact that Steedman should have specified a precise criterion for deciding whether or not any given 12-bar was in the style that he was attempting to characterize—to say that some chord sequence 'does seem to be a 12-bar' is meaningless.

Ebcioğlu points out that any 'style is of course very subjective' and that 'musical beauty and degree of conformance with a style are not quantitatively measurable with an automated process.' He agrees that 'Turing tests are one way of testing the output of a program for style conformance' but claims that 'those too have imperfections similar to the jury system in law.' He 'nevertheless feel[s] that, despite the possible controversy, one should go ahead with this approach, and do one's best in forming a cognitive model for a style.'⁹⁴

It is clear then, that if one wishes to produce a precise and testable theory for a musical style, one must first explicate the style that one is attempting to characterize as a precisely defined object. I believe that if one is prepared to explicate the concept of the style of a corpus of pieces as being the set of all and only those pieces that are either in that corpus or *defined* to be in the style of that corpus by an acceptability algorithm along the lines of a Turing test then it becomes possible to produce satisfactory, objective and empirically testable theories for musical styles.

3.5 *Chomsky's unwillingness to define precisely the natural language that the artificial language generated by a grammar is intended to be equal to*

A generative grammar is, at least on the simplest level, a hypothesis that the artificial language generatively defined by it is equal to the set of all and only 'grammatical' sentences in the natural language that one is attempting to characterize. To test such a grammar for overgeneration, one takes random sentences generated by the grammar and determines whether or not they are grammatical sentences in the natural language being modelled. To test for undergeneration, one takes examples of grammatical sentences in the natural language and determines whether or not they can be generated using the grammar. Clearly, one can test satisfactorily neither for overgeneration nor undergeneration unless one is able to decide in all cases whether or not a given sentence is a 'grammatical' sentence in the natural language that one is attempting to model. Chomsky, however, while admitting that one can only test the adequacy of a grammar if one provides 'a behavioral criterion for grammaticality,' nonetheless is perfectly content to 'assume intuitive knowledge of the grammatical sentences of English.'⁹⁵ As Moore and Carling point out,

⁹² Steedman 1984, 71.

⁹³ Steedman 1984, 66.

⁹⁴ Ebcioğlu 1996a.

⁹⁵ Chomsky 1957, 13.

Chomsky rejected as inadequate all the tests of grammaticality he considered. Instead of seeking more adequate tests, however, he chose to rely on his belief that we can assume: ‘...intuitive knowledge of the grammatical sentences of English.’

They go on to assert—in my opinion, correctly—that

the absence of adequate tests for grammaticality has meant that even when [Chomsky’s] theory was apparently at its most rigorous, there was no way of empirically confirming its findings.⁹⁶

Moore and Carling explain that the task of devising ‘a grammar that will generate the sentoids: aa, bb, abba, baab, aaaa, bbbb, aabbaa, abbbba ... and in general all sentences consisting of a string X followed by the reverse of X and only these’ differs significantly from that of devising a grammar that will generate all and only grammatical sentences in English because ‘the test of the [former] grammar is unambiguously set by the pre-specified conditions.’⁹⁷ As they explain,

were the [first] grammar to permit the generation of the sentoid: abab we would know that it had failed. Were it to generate: abba that it was succeeding. In the case of natural language, however, unless a limit is arbitrarily imposed, there can be no question of stating in advance what the properties of the set of grammatical sentences of English are.⁹⁸

They conclude that

the absence of pre-specified conditions as to what is to count as a grammatical sentence leaves the theoretical linguist using formal language theory in a curious position. Unlike the formal language theorist, he has no clear test for the adequacy of his grammar, no way of telling whether his grammar is performing well or badly. One obvious and, for an empirical science, natural conclusion would be that there was little point in elaborating the grammar until some adequate and theory-independent tests of grammaticality had been devised.⁹⁹

3.6 *Definition of the concept of a corpus and the concept of a corpus kernel*

There are a number of occasions on which Snell seems to confuse the concept of a corpus with that of a language or a musical style. The theory presented in Snell 1979 is intended to be a preliminary attempt to produce a theory for the style of C.P.E. Bach’s *Kurze und Leichte Klavierstücke mit veränderten Reprisen* (1766). I construe one of his long-term goals to be to develop a theory that is essentially a grammar that generates an ‘artificial language’ that is equal to the set of all and only those possible pieces in the style of these 22 *Reprisenstücke* by C.P.E. Bach. C.P.E. Bach’s *Reprisenstücke* therefore form the *corpus* of the style that Snell wishes to characterize with his theory. So when Snell states that after analysing all the members of this corpus he expects his system to approach ‘more closely the goal of generating not only all, but only the pieces in the *corpus*’ (my italics),¹⁰⁰ clearly what he really means is that he expects his system to become a more accurate characterization of the *style represented by his corpus* of 22 *Reprisenstücke*. The aim of a grammar is to generate all and only the pieces in a *language* not a *corpus*. A corpus is an arbitrary set of utterances in a language that one

⁹⁶ Moore and Carling 1982, 82.

⁹⁷ Moore and Carling 1982, 69.

⁹⁸ Moore and Carling 1982, 69.

⁹⁹ Moore and Carling 1982, 69.

¹⁰⁰ Snell 1979, 59–60.

uses to produce a grammar that is intended to account for the language. In Snell's case, the language (i.e. musical style) that he is attempting to characterize contains not only the 22 pieces that C.P.E. Bach himself wrote but also all and only those *possible* pieces that are 'in the style of' this corpus, where 'in the style of' must be explicated in terms of some empirical criterion of stylistic acceptability. I shall discuss below possible ways of explicating the concept of 'being in a particular style.'

Given the foregoing definitions of the concepts of an algorithmic style theory system and the universal set of scores, I shall define that an object may serve as the *corpus kernel* of an algorithmic style theory system if and only if it is a well-defined set of physical scores that contains at least two notationally distinct members.¹⁰¹ The corpus kernel of an algorithmic style theory system T will be denoted $\underline{s}_k(T)$. Also, I shall define that every corpus kernel uniquely defines a *corpus* such that a score s is defined to be a member of the corpus defined by a corpus kernel $\underline{s}_k(T)$ if and only if it is notationally equivalent to at least one member of $\underline{s}_k(T)$. Alternatively, one could say that an object may serve as the corpus of an algorithmic style theory if and only if it is the union of all and only those notational equivalence classes that contain members that are also members of a specified corpus kernel. Every corpus is therefore the union of two or more notational equivalence classes. An object is a *corpus score* in an algorithmic style theory system if and only if it is a member of the corpus of the theory system. The corpus of an algorithmic style theory system T will be denoted $\underline{s}_c(T)$.

In general, theorists are interested in characterizing the styles of particular composers, the styles of music in particular genres by particular composers, the styles of music from particular periods and so on. For example, a theorist may be interested in attempting to characterize the style of the Baroque *concerto grosso*, Mozart's piano sonatas, Bach's chorale harmonizations, 15th century French chansons and so on. The above definition of the corpus of an algorithmic style theory system allows one to define an appropriate corpus for any style of this type. For example, the corpus kernel of an algorithmic style theory designed to account for the style of Bach's chorale harmonizations might be defined to contain all and only those scores in Klaus Schubert's edition of the *371 Four-Part Chorales* (Bach 1990). The corpus of the style theory would then, by definition, contain all and only those scores notationally equivalent to the scores in the Schubert edition. Similarly, if one wished to develop an algorithmic style theory for the style of Joplin's rags, one could define one's corpus kernel to contain, say, all and only those scores in Vera Brodsky Lawrence's edition of the complete piano works of Scott Joplin (Joplin 1981). The corpus of this theory would then by definition contain all and only those scores notationally equivalent to the scores in the Lawrence edition.

It is important to note, however, that the above definition of the concept of a corpus does not require that a corpus be defined to contain all and only scores of pieces that satisfy some specified set of conditions. In particular, it does not specify that the corpus kernel of an algorithmic style theory must contain all and only the scores of a particular composer, or all and only the scores of pieces by a particular composer in a particular genre. Thus one could, for example, quite legitimately decide to attempt to develop an algorithmic style theory for the style of a corpus whose kernel was defined

¹⁰¹ The corpus kernel must contain at least *two* notationally distinct members because the subjects in an acceptability algorithm must know *some* but not *all* of the pieces in the corpus. This will become clear later.

to contain a single score of a rondeau by Dufay together with a single score of a symphony by Mahler! The corpus of the style would then by definition contain all and only those scores that were notationally equivalent either to the Mahler symphony or the Dufay chanson and the style would contain all and only those scores that are either members of this corpus or that are determined to be in the style of this corpus by a specified acceptability algorithm. (The concept of an acceptability algorithm will be defined and discussed below.) It is very difficult to imagine why one should ever want to develop such a theory but I feel it would be too restrictive and over-complicated to demand that the corpus of an algorithmic style theory must be defined to contain all and only those scores by a particular composer or all and only those scores by a particular composer in a particular genre, etc. In any case, my purpose here is to characterize the *logical structure* that a theory for a musical style needs to have if it is not to suffer from certain metatheoretical problems such as untestability. Although the Mahler/Dufay theory might be somewhat bizarre, I hold that if such a theory actually conformed to the structure of an algorithmic style theory (as this structure will be defined over the course of the remainder of this chapter) then it would be *logically* sound.

My main claim in this section is therefore that to be *logically* sound, the corpus of a style theory should be defined as above. Note that I demand that the corpus kernel of an algorithmic style theory system be a *well-defined* set of physical scores. Most style theorists have not, in my opinion, defined their corpora with a satisfactory degree of precision. For example, Baroni states that the first grammar developed by his group was intended to account for the style of ‘the Lutheran chorale melodies later arranged by J.S.Bach’ and that ‘the repertory comprised complete chorales ... and covered the whole corpus of chorales that Bach selected for his collection.’¹⁰² But this ‘definition’ would not allow one to decide categorically for any given score whether or not it was a member of the corpus of the style that Baroni was trying to model. For example, if one discovered a previously unknown manuscript of a chorale harmonization by Bach that was a harmonization of a melody for which no other harmonization by Bach existed, would this melody need to be admitted to Baroni’s corpus? In other words, is Baroni’s corpus defined to contain all and only those melodies for which Bach produced harmonizations, or is it defined to contain only all those melodies for which Bach is *known* to have produced harmonizations, or all those melodies harmonized in some unspecified edition of Bach’s chorales? It is metatheoretical weaknesses like this that I believe can be avoided by constructing style theories that conform to the structure of an algorithmic style theory.

3.7 *The need for an acceptability algorithm that employs ‘blind’ judgements of stylistic acceptability*

Baroni and his collaborators have studied three melodic styles. The first project was an attempt to characterize the style of a corpus consisting of about 100 of the Lutheran chorale melodies harmonized by Bach (see Baroni and Jacoboni 1978; Baroni 1983; Baroni and Jacoboni 1983). The second project was an attempt to model the style of a corpus consisting of the scores in a book of French chansons from the eighteenth century (see Baroni, Brunetti, Callegari and Jacoboni 1984; Baroni and Callegari 1984). The third project was a study of the style of a corpus of 17 arias by Giovanni Legrenzi (Baroni, Dalmonte and Jacoboni 1989; Baroni, Dalmonte and Jacoboni 1992). In the first two cases, a grammar was developed and implemented as a corresponding

¹⁰² Baroni 1983, 185.

computer program that generated melodies intended to be ‘in the style of’ the corpus. In the case of the Legrenzi arias, the program (LEGRE 2) takes a verbal text as input and generates as output a setting for the text that includes a melody and an underlying chord structure, but not a complete accompaniment. In each of the papers cited, only one or two examples of the output of the programs are given so it is difficult to know whether or not these are of exceptionally high quality relative to the rest of the output.

Baroni correctly asserts that ‘according to Chomsky, only the competence of the speakers of a language can give a judgement about the grammatical correctness of a sentence’ but adds that in the case of his own work he and his collaborators ‘cannot say if [their] melodies are grammatically correct but only if they are stylistically correct.’ He claims that in making such a judgement, they are not making

an aesthetic valuation of beauty or ugliness, but a style-historical judgement concerning the possibility that a melody generated automatically could or could not belong to the given repertoire.¹⁰³

I agree with Baroni that a piece which is not a member of the corpus of the style one is attempting to model should be defined to be ‘in the style of’ that corpus if and only if it ‘could belong to’ the corpus. I also agree that the judgements of individuals who are familiar with some (but not all) of the pieces in the corpus should be used to determine whether or not any piece that is not a member of the corpus is ‘in the style of it.’

However, it is hard to imagine how one could make a ‘style-historical judgement concerning the possibility that a melody generated automatically could or could not belong to the given repertoire’ whilst being certain that one was not making ‘an aesthetic valuation of beauty or ugliness.’ Also, unlike Baroni, I claim that such judgements should only be relied upon if they are ‘blind’ judgements—that is, if they are made by individuals who *do not know for sure* whether or not the piece under consideration is in the corpus.

To make this assertion more concrete, imagine that one had developed a grammar that was intended to weakly generate the set of all and only possible melodies in the style of those that Bach used in the chorale harmonizations that appear in Bach 1990. Imagine further that to aid in testing the grammar for overgeneration, one had implemented it (as Baroni’s group has done) as a computer program whose universal output set is equal to the ‘artificial language’ defined by this grammar and that generates on each execution one member of this ‘artificial language’ at random. One could then test this grammar for overgeneration, by running the program a number of times and determining for each of the melodies produced as output whether or not it was ‘in the style of’ the melodies that Bach used in the chorale harmonizations that appear in Bach 1990.

Now Baroni seems to hold the view that one can justifiably decide that a melody is ‘stylistically distinguishable’ from the melodies in the corpus if and only if it is deemed to be so by someone who is familiar with some of the melodies in the corpus *and also knows that the given melody is not a member of this corpus because it was composed automatically by the computer program that implements the grammar*. In my view, however, one could never be sure that the judgements of individuals who are aware of the artificial origin of the automatically generated piece were not biased. For

¹⁰³ Baroni et al. 1992, 188.

example, Baroni himself might be more ready to judge a melody automatically composed by one of his programs to be ‘in the style’ that the program is intended to model than some other music theorist who had invented a competing theory for the same style.

Such problems of bias disappear, however, if one is content to explicate the concept of a piece ‘being in the style of’ a particular corpus by *defining* it to be if and only if individuals who are familiar with some but not all of the members of this corpus, when presented with a set of scores containing a sample from the corpus and the test score, are unable to identify which of the scores is not in the corpus. In my view, it is much more informative and interesting to say that a certain proportion of subjects correctly identified the automatically-generated piece in such an experiment than to say merely that the piece seemed to some particular individual to be in the style of the corpus.

3.8 *Definition of the concept of an acceptability algorithm*

The acceptability algorithm in conjunction with the corpus of an algorithmic style theory system must precisely define the *style* that the algorithmic style theory is intended to account for. The acceptability algorithm must be an algorithm that, when given any score as input, generates as output either a judgement that the score is in the style of the corpus or a judgement that the score is *not* in the style of the corpus. In other words, the acceptability algorithm of an algorithmic style theory system must be a decision procedure for determining for any given score whether or not it is in the style of the corpus of the theory system.

An algorithmic style theory, like a generative grammar, is a hypothesis that a particular generatively defined set of scores called the universal set of well-formed scores is equal to a particular *style* which must therefore be defined to be a set containing all and only those scores that are either in a specified corpus or ‘stylistically indistinguishable from’ or ‘in the style of’ the scores in this corpus. The acceptability algorithm must therefore provide a satisfactory *explication* of the quality of ‘being in the style of the scores in a corpus.’ The acceptability algorithm is a necessary component of a style theory because it provides a precise definition of the set of scores that the universal set of well-formed scores defined by the composing algorithm is intended to equal. Without the acceptability algorithm, it would be impossible to determine for any score whether or not it was in the style being modelled. Consequently, it would be impossible to test the theory.

In my view, the acceptability algorithm of an algorithmic style theory system should take the form of an experiment in which each of a number of subjects is given a single trial called a *test trial*. Each subject must be familiar with some but not all of the scores in the corpus.¹⁰⁴ In a test trial, the subject is presented with a set of two or more notationally distinct scores printed or displayed in exactly the same format so that it is not possible for the subject to derive any information about the circumstances of composition of the pieces represented in the scores other than from the logical organization of the Standard Notation symbols in the scores. The set of scores presented to a given subject in a given test trial is called the *test set* for that trial.

¹⁰⁴ It is this requirement that implies that the corpus kernel must contain at least two notationally distinct scores.

The acceptability algorithm is executed in order to determine whether or not a particular score called the *test score* is in the style of the corpus. This test score must be present in the test set for each test trial. The test score must not be a member of the corpus but all members of the test set in each test trial apart from the test score must be members of the corpus. The set containing all and only those scores in a test set that are also members of the corpus is called the *corpus sample*. The corpus sample must be as representative as possible of the corpus as a whole and therefore must be selected randomly from those scores in the corpus that are known to the experimenter. Also, the number of scores in the corpus sample must be greater than some minimum number calculated using standard techniques of statistical analysis.

In each trial the subject must not know the test score and must not know at least one of the scores in the corpus sample. Each subject is given an answer sheet. On this answer sheet, the subject must indicate for each member of the test set whether or not he or she knows the piece. The subject is told the definition of the corpus and is also told that all of the pieces in the test set are in the corpus except for one. The subject is then asked to identify the score in the test set that is not in the corpus. The scores must be presented in such a way that it is impossible for the subject to be able to decide whether or not any score in the test set is a member of the corpus on the basis of any information other than the logical organization of the symbols in the scores, his or her previous knowledge of pieces in the corpus and the definition of the corpus provided by the experimenter. (As will become clear below, if all the scores in the test set are generated by the score algorithm of the theory from representations derived in the case of the corpus sample from scores in the corpus then this will be guaranteed.)

In general, executing an acceptability algorithm for a particular score consists of carrying out as many test trials as possible with the score as the test score in each trial. The result of the acceptability algorithm is an answer sheet from each subject indicating which of the scores in the test set were known by the subject and which score he or she thought was the test score. In any given test trial, if the subject knows the test score or knows all of the corpus sample, then the trial is ignored. Otherwise, in general, given a trial in which the test set contains n scores and the subject knows m scores in the corpus sample then the subject will either correctly identify the test score, incorrectly identify the test score or claim to be unable to decide which of the unknown scores is the test score. The probability of the subject correctly identifying the test score after making a random decision is

$$\frac{1}{n - m}$$

Let us say that one has executed an acceptability algorithm as described above in order to determine whether or not a given test score is in the style of a corpus. Let us say that one obtained s valid test trials (i.e. ones in which, before carrying out the test trial, the subject did not know the test score and did not know at least one member of the corpus sample). Clearly, if a given subject makes an incorrect identification or is unable to make an identification in a given trial then, for that subject at that time and for that corpus sample, the test score is to all intents and purposes ‘stylistically indistinguishable from’ or ‘in the style of’ the scores in the corpus. If the subject makes a correct identification then either he or she is making a random judgement or the test score is ‘stylistically distinguishable from’ or ‘not in the style of’ the scores in the corpus sample. Let us further assume that the test score was correctly identified in c test trials out of the total of s trials obtained in the experiment and incorrectly identified in i

test trials, so that the number of trials in which the subject was unable to decide which score was the test score was

$$s - (c + i)$$

One could then define that a score is in the style of the corpus if and only if the probability of c random, correct identifications in s trials is below some specified threshold determined by standard criteria of statistical significance. It would seem reasonable also to specify that if a score s is determined by the acceptability algorithm to be in the style of a specified corpus, then all scores notationally equivalent to s should also be defined to be in the style of that corpus. In my view, such an experimental procedure would constitute a satisfactory acceptability algorithm for an algorithmic style theory system.

3.9 *Definition of the universal set of acceptable scores, the universal set of unacceptable scores and the style of an algorithmic style theory*

Given the above definitions of the universal set of scores, the acceptability algorithm and the corpus of an algorithmic style theory system, it is now possible to define a number of other concepts associated with such a theory system. The *universal set of acceptable scores* of an algorithmic style theory system is defined to be the set that contains all and only those scores that are not members of the corpus but that when given to the acceptability algorithm as input are determined by that algorithm to be in the style of the corpus. An object is an *acceptable score* if and only if it is a member of the universal set of acceptable scores. The universal set of acceptable scores of an algorithmic style theory system will be denoted $\underline{s}_a(T)$.

The *style* of an algorithmic style theory system is defined to be the union of the universal set of acceptable scores and the corpus of the style theory system. An object is a *style score* in an algorithmic style theory system if and only if it is a member of the style of the theory system. The style of an algorithmic style theory system T is denoted $\underline{s}_s(T)$, therefore

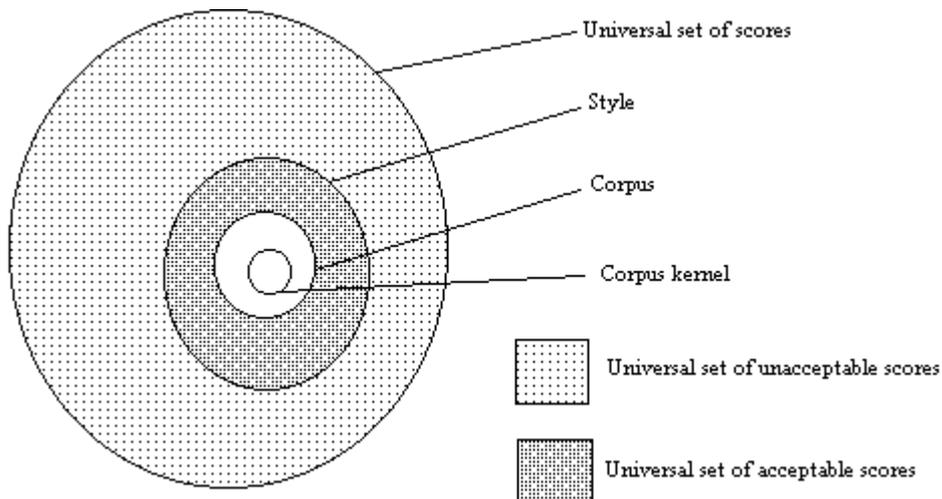


Figure 3-2

$$\underline{s}_s(T) =_{\text{df}} \underline{s}_a(T) \cup \underline{s}_c(T)$$

The *universal set of unacceptable scores* of a theory system is the relative complement of the style in the universal set of scores. The relative complement of a set A in another set B , denoted $B \setminus A$, is the set that contains all and only those objects that are members of B and not members of A . The universal set of unacceptable scores of an algorithmic style theory system T is denoted $\underline{s}_n(T)$, therefore

$$\underline{s}_n(T) =_{\text{df}} \underline{s}_u(T) \setminus \underline{s}_s(T)$$

An object is an *unacceptable score* if and only if it is a member of the universal set of unacceptable scores. The set-theoretical relationship between the corpus kernel, the corpus, the style, the universal set of acceptable scores, the universal set of unacceptable scores and the universal set of scores is shown in Figure 3-2.

3.10 The concept of an algorithmic style theory

As already stated above, an object T is an *algorithmic style theory system* if and only if it is an 8-tuple as follows:

$$\langle \underline{s}_u(T), \underline{s}_k(T), \alpha(T), \rho(T), \gamma(T), \sigma(T), \delta(T), \pi(T) \rangle$$

where

$\underline{s}_u(T)$ is the *universal set of scores* of T ;

$\underline{s}_k(T)$ is the *corpus kernel* of T ;

$\alpha(T)$ is the *acceptability algorithm* of T ;

$\rho(T)$ is the *representation algorithm* of T ;

$\gamma(T)$ is the *composing algorithm* of T ;

$\sigma(T)$ is the *score algorithm* of T ;

$\delta(T)$ is the *derivation algorithm* of T ;

$\pi(T)$ is the *parsing algorithm* of T .

The concepts of the universal set of scores, the corpus kernel and the acceptability algorithm of an algorithmic style theory system have been introduced above. The concepts of the corpus, the universal set of acceptable scores, the universal set of unacceptable scores and the style of an algorithmic style theory system have also been defined. The rest of this section will be devoted to introducing and defining the remaining components of an algorithmic style theory system.

Figure 3-3 represents schematically the structure of an algorithmic style theory system and shows the relationships between the different components. The large square region on the left hand side of the diagram bounded by a thick solid line represents the universal set of scores. This large square is divided up into 36 small square regions bounded by thin solid lines. Each of these small square regions represents a notational equivalence class. The number of small squares is not significant—in practice, the universal set of scores will be partitioned into a vast number of notational equivalence classes. I shall refer to individual notational equivalence classes in Figure 3-3 using a co-ordinate system. Each column of six notational equivalence classes in Figure 3-3 is denoted by a letter from A to F and each row by a number from 1 to 6. Thus, for example, the notational equivalence class in the top left hand corner of the universal set of scores containing the two scores, s_3 and s_4 , will be referred to as notational equivalence class A1, the score s_5 is in notational equivalence class D2 and so on. I shall denote any rectangular or square region within the universal set of scores equal to a union of notational equivalence classes by specifying the top left and bottom right

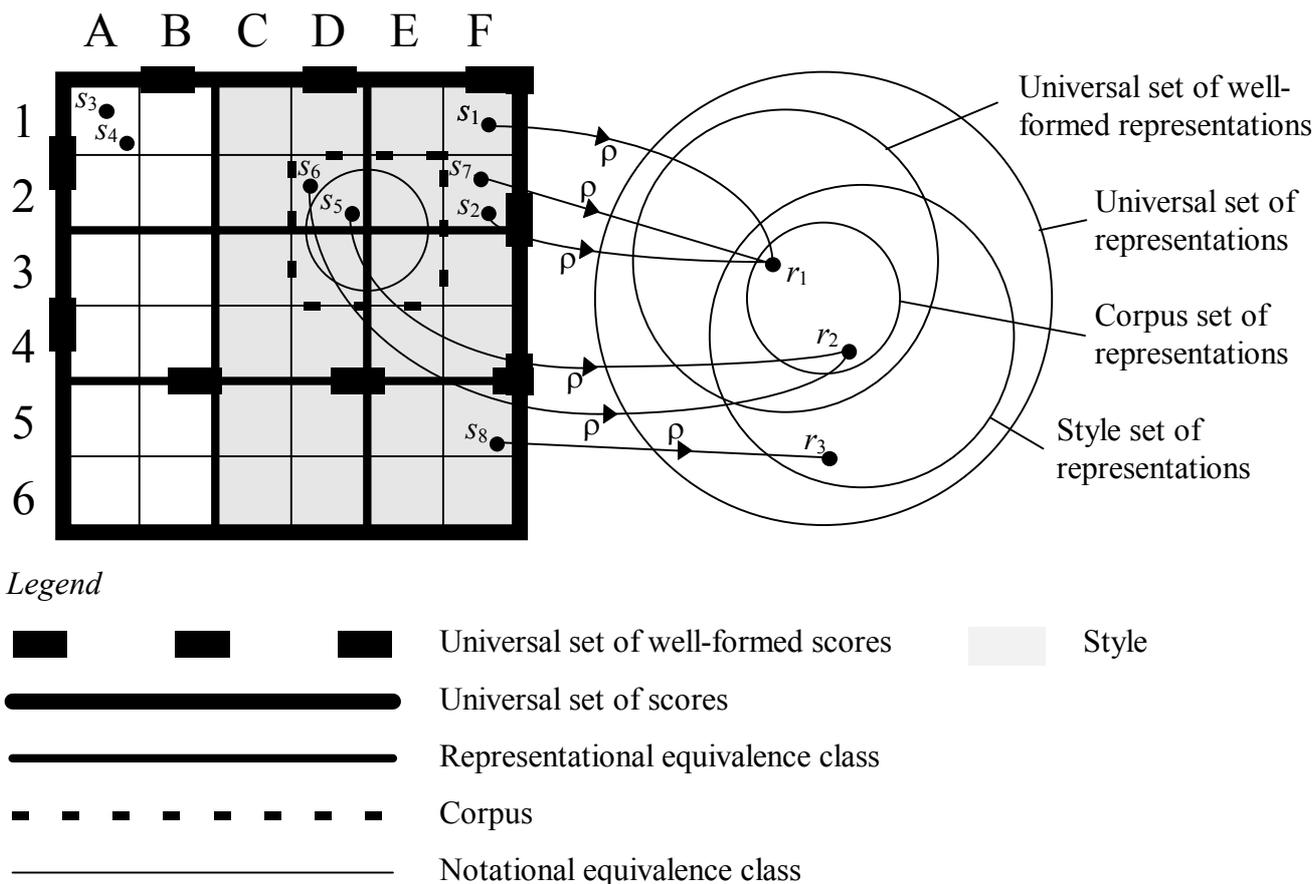


Figure 3-3

notational equivalence classes contained within the region. For example, the shaded rectangle that has square C1 in its top left hand corner and F6 in its bottom right hand corner would be referred to as ‘rectangle C1-F6.’

The circular region contained within square D2-E3 represents the corpus kernel. Note that, as shown in the diagram, the corpus kernel is not in general a union of notational equivalence classes. s_5 represents a physical score in the corpus kernel that is notationally equivalent to another score, s_6 , which is not a member of the corpus kernel. The square D2-E3 bounded by the medium-width dotted line represents the corpus. Note that the corpus is equal to the union of all notational equivalence classes that intersect the corpus kernel.

The shaded rectangle C1-F6 represents the style. Note that the style contains the corpus by definition. As stated above, if a score s is determined by the acceptability algorithm of an algorithmic style theory to be in the style of a given corpus, then all scores notationally equivalent to s are also defined to be in the style of the corpus. The universal set of acceptable scores is therefore a union of notational equivalence classes. Since the corpus of a style theory system is also a union of notational equivalence classes by definition, this implies that the style itself is a union of notational equivalence classes. The universal set of acceptable scores is equal to that portion of the shaded region outside of the corpus, and the universal set of unacceptable scores is equal to the unshaded region of the universal set of scores (i.e. rectangle A1-B6).

The *representation algorithm* of an algorithmic style theory system must be an algorithm that, when given a score from the universal set of scores as input, generates

for that score a *representation* as output. In other words, the universal input set of the representation algorithm must be the universal set of scores and the representation algorithm must map each score onto one and only one member of its universal output set which is called the *universal set of representations*. The representation algorithm of an algorithmic style theory system T will be denoted $\rho(T)$ and the universal set of representations of T will be denoted $\underline{r}_u(T)$. In Figure 3-3, the universal set of representations is represented by the large circle on the right hand side of the diagram. An object is a *representation* in a specified style theory system if and only if it is a member of the universal set of representations of the style theory system. A representation r is the representation of a score s if and only if the representation algorithm maps s onto r . In Figure 3-3, the fact that a score is mapped onto a particular representation by the representation algorithm is indicated by a directed line segment labelled with the symbol ρ drawn from the score to its corresponding representation. For example, when the representation algorithm of the theory system represented in Figure 3-3 is given score s_1 as input, it generates representation r_1 as output.

Two scores are *representationally equivalent* if and only if the representation algorithm maps them onto the same representation. The universal set of scores is therefore exhaustively and exclusively partitioned into *representational equivalence classes* where the representational equivalence class to which any given score belongs is the set of all and only those scores that are representationally equivalent to it. If two scores are notationally equivalent then they will also be representationally equivalent. This is illustrated in Figure 3-3 by the notationally equivalent scores s_2 and s_7 that each map onto representation r_1 . However, two representationally equivalent scores will not necessarily be notationally equivalent because the representation algorithm does not necessarily represent every symbol in a score. Thus in Figure 3-3, scores s_1 and s_2 are notationally distinct but representationally equivalent because they both map onto representation r_1 . Therefore every representational equivalence class is equal to a union of notational equivalence classes. The representational equivalence classes in Figure 3-3 are bounded by medium-width solid lines. Each representational equivalence class in Figure 3-3 is arbitrarily assumed to be equal to the union of four notational equivalence classes. For example, the representational equivalence class E1-F2 contains the notational equivalence classes E1, F1, E2 and F2. All and only the scores in this representational equivalence class will be mapped by the representation algorithm onto representation r_1 .

The *corpus set of representations* of an algorithmic style theory system is the set that contains all and only those representations that are representations of scores in the corpus. In other words, the corpus set of representations is the output set of the representation algorithm for an input set equal to the corpus. The corpus set of representations is also, of course, the output set of the representation algorithm for an input set equal to the corpus kernel. The corpus set of representations of a style theory system T will be denoted $\underline{r}_c(T)$. A representation is a *corpus representation* in a specified algorithmic style theory system if and only if it is a member of the corpus set of representations of the theory system. Thus, in Figure 3-3, representations r_1 and r_2 are corpus representations. r_1 is the representation of scores s_1 , s_2 and s_7 . r_1 is a corpus representation because s_1 , s_2 and s_7 are members of the representational equivalence class E1-F2 which intersects the corpus. r_1 is therefore the representation of the corpus scores in notational equivalence class E2. r_2 is the representation of corpus kernel score s_5 and the notationally equivalent corpus score s_6 .

The *style set of representations* of an algorithmic style theory system is the set that contains all and only those representations that are representations of scores in the style defined by the corpus and acceptability algorithm of the theory system. In other words, the style set of representations is the output set of the representation algorithm for an input set equal to the style. The style set of representations of an algorithmic style theory system T will be denoted $\underline{r}_s(T)$. A representation is a *style representation* in an algorithmic style theory system if and only if it is a member of the style set of representations of the theory system. In Figure 3-3, representations r_1 , r_2 and r_3 are all style representations.

The *score algorithm* of an algorithmic style theory system must be an algorithm that when given a representation r as input generates as output a score s such that s is a member of the representational equivalence class of scores that are mapped by the representation algorithm onto r . Thus the universal input set of the score algorithm must be equal to the universal set of representations. Also, the output set of the score algorithm for a given input representation r must be a subset of a notational equivalence class that is in turn a subset of the representational equivalence class of scores that are mapped by the representation algorithm onto r . The score algorithm must be implementable as a computer program that requires no runtime input during execution other than the representation for which it is generating a score.

Figure 3-4 shows four representationally equivalent scores, s_1 - s_4 that are mapped onto corpus representation r_1 . The fact that a representation is mapped by the score algorithm onto a particular score is depicted in Figure 3-4 by means of a directed line drawn from the representation to the score and labelled with the letter σ . For example, the score algorithm of the theory system illustrated in Figure 3-4 maps representation r_1 onto score s_2 .

As stated above, when carrying out the acceptability algorithm, a subject must be presented in each test trial with a set of two or more notationally distinct scores printed or displayed in exactly the same format so that it is not possible for the subject to derive any information about the circumstances of composition of the pieces represented in the scores other than from the logical organization of the Standard Notation symbols in the scores. This can be achieved by using the score algorithm to generate all of the scores in the test set of each test trial. The scores in the corpus sample in each test trial must be notationally equivalent to scores in the corpus kernel—that is, the scores in the corpus sample must be members of the corpus. But as the score algorithm must be a computationally implementable algorithm, the score generated by the score algorithm for a given representation can contain no information that cannot be derived algorithmically from the data in the representation. This implies that the representations generated by the representation algorithm must contain sufficient information so that when the score algorithm is given corpus representations as input, it outputs scores that are notationally equivalent to scores in the corpus kernel. In general, this sets a rather high lower limit on the amount of information in a score that the representation algorithm must preserve in the representations that it generates as output.

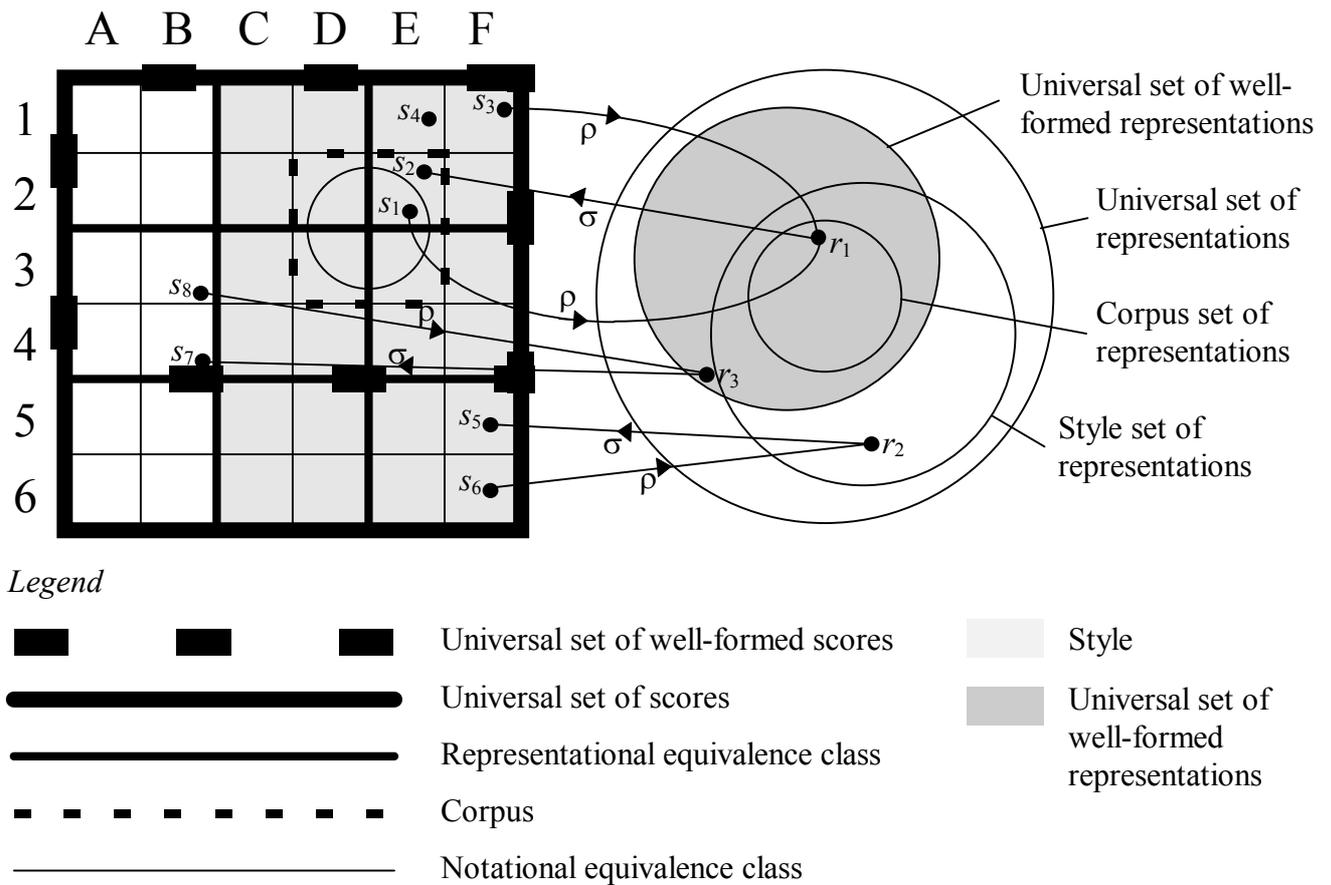


Figure 3-4

In Figure 3-4, s_1 is a score in the corpus kernel. This score is mapped onto corpus representation r_1 by the representation algorithm. To use a score that was notationally equivalent to s_1 in the corpus sample of a test trial in an execution of the acceptability algorithm, the experimenter would have to first derive r_1 from s_1 using the representation algorithm and then use the score algorithm to generate a score notationally equivalent to s_1 containing no information that a subject could use to determine the provenance of s_1 other than the logical organization of the SN symbols in the score. In other words, the score algorithm must generate a corpus score such as s_2 from r_1 . If the score algorithm generated score s_4 when given representation r_1 as input, then this would show that either it or the representation algorithm was not satisfactory. Note that although the corpus is by definition a union of notational equivalence classes it is not necessarily a union of representational equivalence classes because there may in general exist scores that are representationally equivalent to corpus scores but that are not notationally equivalent, perhaps, for example, because they have dynamic or expression markings that are not present in the corpus scores.

Therefore, in addition to the constraints given above that must be satisfied by the score algorithm and representation algorithm of an algorithmic style theory, it must also be true that whenever a corpus representation r is given as input to the score algorithm, the score algorithm generates a score that is notationally equivalent to all corpus scores that are mapped by the representation algorithm onto r .

However, because the representation algorithm may not preserve *all* the symbols in a score, it is in general possible for two scores to be representationally equivalent and

notationally distinct. And in particular, the situation might arise as shown in Figure 3-4 where a non-corpus score s_6 , whose representation is r_2 , is notationally distinct from the score s_5 that the score algorithm generates from r_2 . The style of an algorithmic style theory system is defined to be equal to the union of the corpus and the universal set of acceptable scores. The universal set of acceptable scores contains all and only those scores that are determined to be in the style of the corpus by the acceptability algorithm. Therefore, to determine whether or not a non-corpus score such as s_6 is in the style, it would be necessary to carry out the acceptability algorithm with score s_6 as input. But to ensure that subjects use only their prior knowledge of the corpus and the logical organization of the symbols in the score in their attempts to identify the test score in each test trial, all the scores in the test set of each trial need to be presented in an identical format and therefore must be generated by the score algorithm. It would not therefore be possible to use s_6 itself as the test score in a test trial. The best that one could do would be to first generate the representation of s_6 (i.e. r_2) using the representation algorithm and then use the score algorithm to generate a score from r_2 . But the score that is generated by the score algorithm from r_2 is s_5 which is notationally distinct from s_6 . Therefore it would be impossible to determine whether or not any score notationally equivalent to s_6 was in the style. This problem can be solved simply by specifying that if a score s generated by the score algorithm is determined by the acceptability algorithm to be in the style, then all other scores representationally equivalent to s are defined to be also in the style. This implies that the style of an algorithmic style theory system will always be a union of representational equivalence classes as shown in Figure 3-4.

The *composing algorithm* of an algorithmic style theory system T will be denoted $\gamma(T)$. The composing algorithm must be an algorithm whose universal output set is a subset of the universal set of representations. The universal output set of the composing algorithm of a style theory system is called the *universal set of well-formed representations* of the theory system. In Figure 3-4 the circular region of the universal set of representations that represents the universal set of well-formed representations is shaded. The *universal set of ill-formed representations* of a style theory system is defined to be the relative complement of the universal set of well-formed representations in the universal set of representations. In Figure 3-4 the universal set of ill-formed representations is represented by the unshaded portion of the universal set of representations. A representation is *well-formed* in the context of a particular algorithmic style theory system if and only if it is a member of the universal set of well-formed representations, otherwise it is *ill-formed*. For example, in Figure 3-4, r_1 is a well-formed representation and r_2 is an ill-formed representation. The universal set of well-formed representations of an algorithmic style theory system T will be denoted $\underline{r}_w(T)$ and the universal set of ill-formed representations of T will be denoted $\underline{r}_i(T)$. Therefore

$$\underline{r}_i(T) =_{df} \underline{r}_u(T) \setminus \underline{r}_w(T)$$

In addition to being an algorithm whose universal output set is a subset of the universal set of representations, the composing algorithm must be implementable as a working computer program called the *composing program* of the algorithmic style theory system. On each execution, the composing program may take as input only one or more random or pseudo-random numbers and must generate as output a single member of the universal set of well-formed representations. The composing program must not require any external input from the user. It must also be possible to use the

composing program to generate a random or pseudo-random sample of representations from the universal set of well-formed representations. The reason for the latter constraint will be given shortly.

The *universal set of well-formed scores* of an algorithmic style theory system is the set that contains all and only those scores that are mapped by the representation algorithm onto members of the universal set of well-formed representations. In other words, the universal set of well-formed scores is the input set of the representation algorithm for an output set equal to the universal set of well-formed representations. The universal set of well-formed scores of a theory system T will be denoted $\underline{s}_w(T)$. This implies that the universal set of well-formed scores is always a union of representational equivalence classes. In Figure 3-4, the universal set of well-formed scores is represented by rectangle A1-F4 and is bounded by a very thick dotted line. A score is a *well-formed score* in the context of a particular theory system if and only if it is a member of the universal set of well-formed scores. In Figure 3-4, scores s_1, s_2, s_3, s_4, s_7 and s_8 are all well-formed scores. The *universal set of ill-formed scores* of a style theory system is defined to be the relative complement of the universal set of well-formed scores in the universal set of scores. The universal set of ill-formed scores of a theory system T will be denoted $\underline{s}_i(T)$. Therefore,

$$\underline{s}_i(T) =_{\text{df}} \underline{s}_u(T) \setminus \underline{s}_w(T)$$

In Figure 3-4, rectangle A5-F6 represents the universal set of ill-formed scores and s_5 and s_6 are examples of ill-formed scores.

Every algorithmic style theory system has a single *algorithmic style theory* associated with it. The *algorithmic style theory* associated with an algorithmic style theory system T is the hypothesis that the universal set of well-formed scores of T is equal to the style of T . That is, the algorithmic style theory of a theory system T is the hypothesis that

$$\underline{s}_w(T) = \underline{s}_s(T)$$

The algorithmic style theory associated with a theory system T is *true* or *correct* if and only if $\underline{s}_w(T) = \underline{s}_s(T)$, otherwise it is *false* or *incorrect*. The algorithmic style theory associated with a theory system T is *verified* if and only if it has been *proved* that $\underline{s}_w(T) = \underline{s}_s(T)$ and *falsified* or *refuted* if and only if it has been proved that $\underline{s}_w(T) \neq \underline{s}_s(T)$. Also, the algorithmic style theory associated with a theory system T is said to *overgenerate* if and only if $\underline{s}_w(T)$ contains scores that are not members of $\underline{s}_s(T)$ and *undergenerate* if and only if $\underline{s}_s(T)$ contains scores that are not members of $\underline{s}_w(T)$. An algorithmic style theory is therefore correct if and only if it does not overgenerate and it does not undergenerate.

Figure 3-5 shows an algorithmic style theory system whose associated algorithmic style theory is correct. Figure 3-6 shows an algorithmic style theory system whose associated style theory is incorrect because it overgenerates. Figure 3-7 shows an algorithmic style theory system whose associated style theory is incorrect because it undergenerates. The algorithmic style theory associated with the theory system in Figure 3-4 both undergenerates and overgenerates.

It is important to note that at any given instant in time, any given algorithmic style theory is either *true* (i.e. *correct*) or *false* (i.e. *incorrect*). Also, by virtue of the constraints that must be satisfied by the members of an algorithmic style theory system,

it is always possible to determine by logical deduction and empirical observation alone whether or not any given score refutes any given algorithmic style theory.

3.11 *The nature of a research programme for the development and testing of an algorithmic style theory*

The first step in a research programme whose goal is to develop an algorithmic style theory should be to define the style that the theory is going to be an attempt to characterize by specifying a corpus kernel and an acceptability algorithm. The second step is to attempt to find a representation algorithm, a composing algorithm and a score algorithm that together define a universal set of well-formed scores that is equal to the style that has already been fully specified by the corpus kernel and the acceptability algorithm. When a representation algorithm, a composing algorithm and a score algorithm have been devised, one is then in a position to hypothesize that the universal set of well-formed scores defined by these three algorithms is equal to the style defined by the acceptability algorithm and the corpus kernel. The third step in an algorithmic style theory research programme is therefore to test this hypothesis—that is, to attempt to determine whether or not the universal set of well-formed scores is equal to the style. In my opinion, one should continue to test an algorithmic style theory until one has either verified or refuted it. As stated above, an algorithmic style theory is correct if and only if it does not overgenerate and it does not undergenerate. Testing an algorithmic style theory therefore reduces to testing it for overgeneration and testing it for undergeneration.

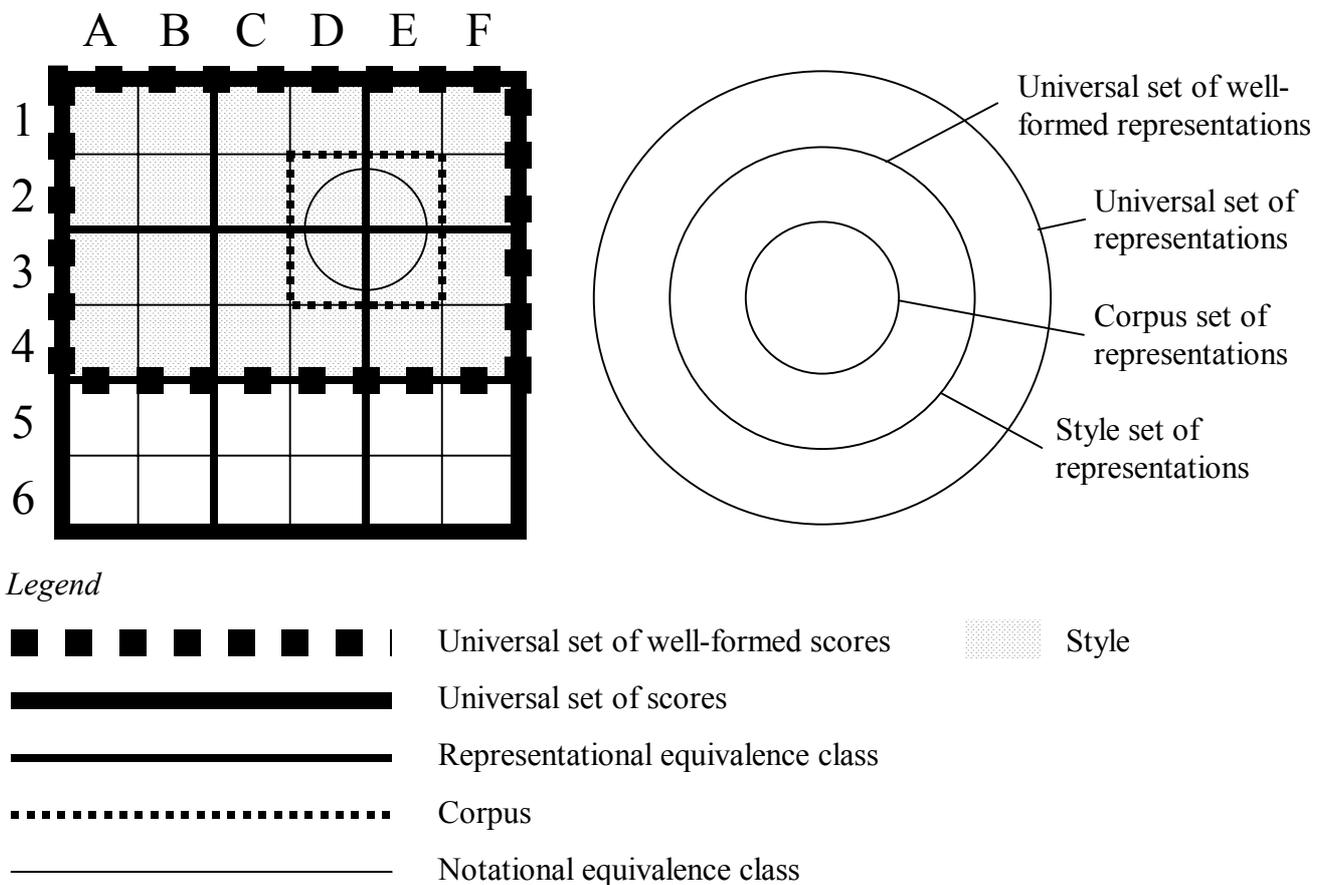


Figure 3-5

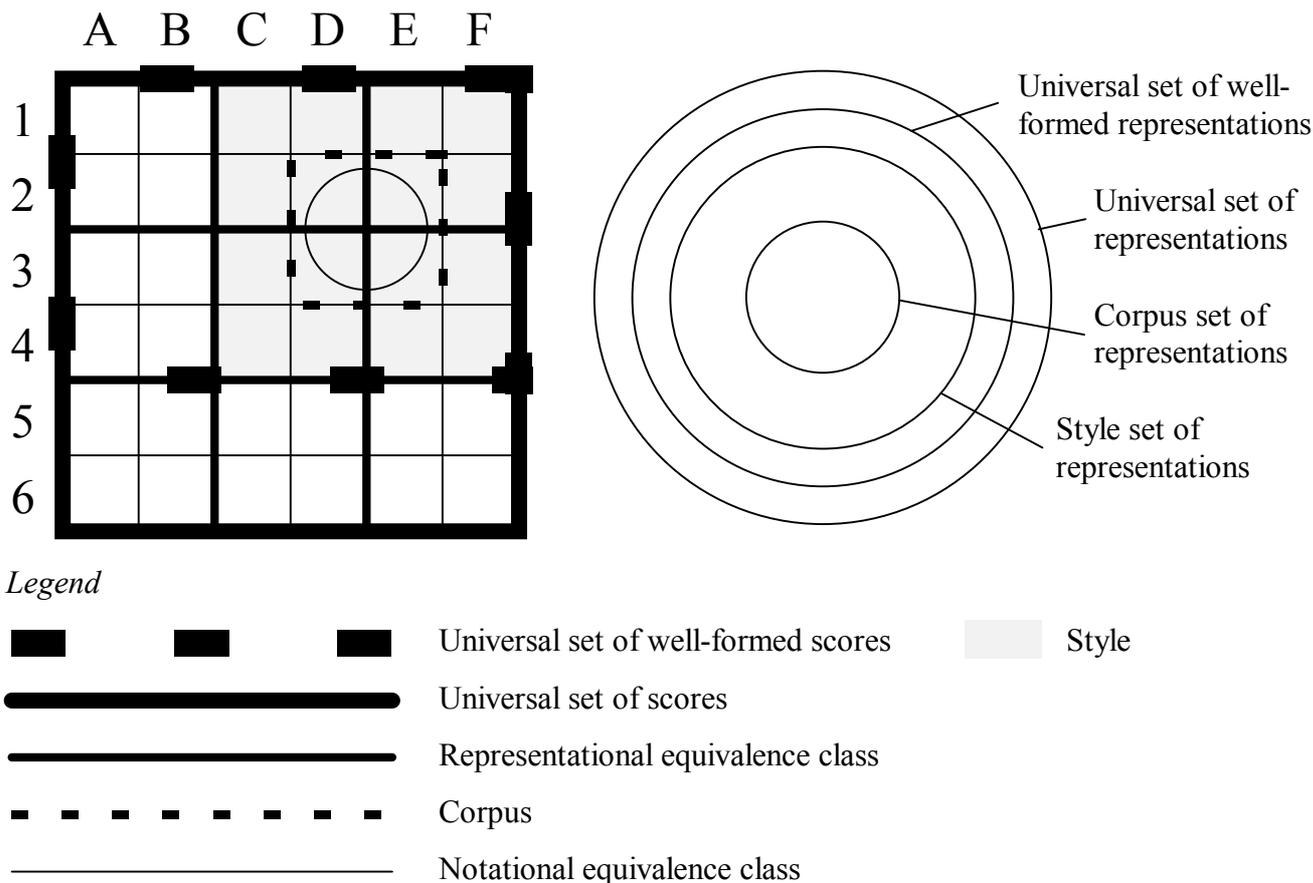


Figure 3-6

To test an algorithmic style theory for undergeneration, one needs to define a derivation algorithm and a parsing algorithm. The *derivation algorithm* of an algorithmic style theory system must be an algorithm whose universal output set is provably equal to the universal set of well-formed representations. The derivation algorithm should be essentially identical to the composing algorithm except that, whereas the composing algorithm must generate a random member of the universal set of well-formed representations on each execution, the derivation algorithm takes as input a *derivation* that specifies exactly which option the algorithm is to choose from the available options at each decision point in the algorithm. A *well-formed derivation* is a precise description of one particular possible execution of the composing algorithm. The derivation algorithm must be such that if and only if it is given a well-formed derivation d as input it generates as output the well-formed representation that would be generated by the composing algorithm if the sequence of choices made during an execution of the composing algorithm were the one that is described by d . Any given derivation must be either a *well-formed derivation* or an *ill-formed derivation*. If the derivation algorithm is given an *ill-formed derivation* as input then it must terminate without generating a representation.

The *parsing algorithm* of an algorithmic style theory system must be an algorithm that satisfies the following conditions:

1. the universal input set of the parsing algorithm must be the universal set of representations;

2. whenever the parsing algorithm is given a well-formed representation r as input it generates as output a well-formed derivation d such that the derivation algorithm generates r as output when it is given d as input;
3. whenever the parsing algorithm is given an ill-formed representation as input it terminates but does not generate a well-formed derivation.

The parsing algorithm of an algorithmic style theory system must be implementable as a computer program called the *parsing program* of the theory system. On each execution this program takes as input a representation and no other external input from the user.

The process of testing an algorithmic style theory for undergeneration is essentially that of attempting to find an ill-formed score that is a member of the style. The first step is to find a score s that is in the style. s can either be a corpus score or an acceptable score. The second step is to derive r , the representation of s , using the representation algorithm. The third step is to execute the parsing algorithm with r as input. If r is well-formed, the parsing algorithm will generate a well-formed derivation d that when given to the derivation algorithm as input will re-generate r as output. If r is ill-formed then the parsing algorithm will not generate a well-formed derivation for it. One should carry out this sequence of three steps repeatedly on different style scores until one finds an ill-formed score that is in the style. As soon as one has found an ill-formed score that is a member of the style, one has proved that the theory undergenerates and has therefore refuted the theory. The style of the music in a

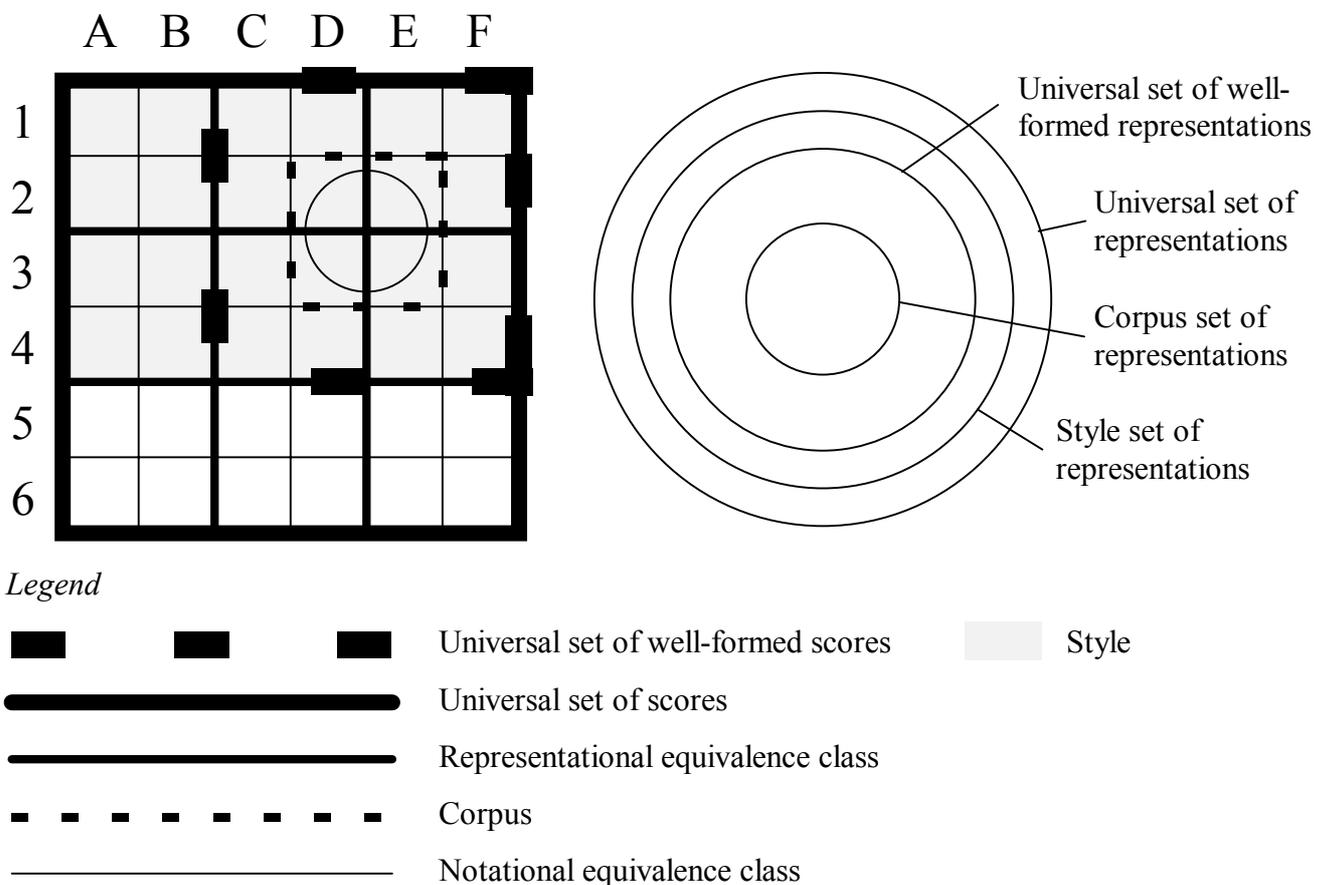


Figure 3-7

particular genre by a particular composer will typically be far too large for one ever to be able to show that every score in the style is well-formed. Therefore, in practice, one would never be able to verify that a style theory did not undergenerate. However, it might well be possible to show that all known members of the corpus were well-formed and thus verify at least that the style theory did not undergenerate with respect to the known members of the corpus.

An algorithmic style theory can be tested for overgeneration by using the composing algorithm and the score algorithm to generate members of the universal set of well-formed scores at random. To generate a random score from the universal set of well-formed scores of an algorithmic style theory, one first uses the composing algorithm to generate at random a well-formed representation and then gives this representation to the score algorithm as input which in turn produces a well-formed score. Each well-formed score produced in this manner must then be given as input to the acceptability algorithm in order to determine whether or not the score is a member of the style. As soon as one succeeds in generating a well-formed but unacceptable score in this manner, the theory has been shown to overgenerate and has therefore been refuted. The universal set of well-formed representations generated by the composing algorithm of any algorithmic style theory that was a feasible theory for the music in a particular genre by a particular composer would typically be far too large for one ever to be able to show that every well-formed score was in the style. Therefore, in practice, one would never be able to verify that an algorithmic style theory did not overgenerate.

When one has developed, tested and refuted one's first algorithmic style theory for a particular style as just described, one must then reformulate the theory by making appropriate changes to the composing algorithm, score algorithm and representation algorithm. In a research programme directed towards the goal of developing an algorithmic style theory, the process described above of constructing a style theory followed by testing the theory until it is refuted is repeated indefinitely. In a successful style theory research programme, the testing phase in each successive reformulation/testing cycle should become longer and longer as it becomes increasingly more difficult to refute the theory. Similarly, the theory system reformulation phase should get shorter and shorter as one approaches a correct theory.

3.12 *An algorithmic style theory is an unverifiable hypothesis*

As mentioned above, a grammar is said to *weakly generate* the natural language that it is intended to model if and only if it does not undergenerate and does not overgenerate. Some authors describe a grammar as being *complete* if and only if it weakly generates the language that it is intended to characterize. For example, Baroni states that a 'fundamental property that a grammar must possess is completeness. That is it must be able to generate all the appropriate features, excluding all others, which can refer to a given repertory.'¹⁰⁵

However, Baroni claims that 'it is possible to verify the completeness of a grammar testing the results of its application by means of one's competence in the language of that repertory.'¹⁰⁶ Now to *verify* the completeness of a grammar it would be necessary to prove that it weakly generated the natural language that it was intended to model. In other words, it would be necessary to *prove* both that it did not undergenerate

¹⁰⁵ Baroni et al. 1984, 203–4.

¹⁰⁶ Baroni et al. 1984, 203–4.

and that it did not overgenerate. Baroni admits that it would be ‘difficult’ to do this ‘since the calculation of all the possible consequences of a rule [in a grammar] is a tremendous task inasmuch as any rule can interfere with the effect of all the others.’ However, he claims that he has been able to ‘solve [this] problem with the aid of a computer’ and that ‘examining melodies produced automatically on the basis of [his] grammar enables [him] to evaluate its completeness.’¹⁰⁷ Similarly, elsewhere he states that ‘a computer program verifies the completeness and self-consistency of the ... generative rules used in the analysis.’¹⁰⁸ That is, Baroni is essentially claiming that it would be possible to verify an algorithmic style theory. To do this, one would need to prove that the theory did not overgenerate and did not undergenerate.

To prove that an algorithmic style theory did not overgenerate one would need to prove that every well-formed score was in the style being modelled. If a score *s* is determined by the acceptability algorithm to be in the style, then all scores representationally equivalent to *s* are also defined to be in the style. Therefore to verify that an algorithmic style theory did *not* overgenerate, one would need to show for every well-formed *representation* that it was the representation of a score in the style being modelled. But, as mentioned above, the universal set of well-formed representations generated by the composing algorithm of any algorithmic style theory that was a feasible theory for the music in a particular genre by a particular composer would typically be far too large for one ever to be able to show for every well-formed representation that it was the representation of a score in the style being modelled. Therefore, in practice, one would never be able to verify that an algorithmic style theory did not overgenerate.

Similarly, to prove that an algorithmic style theory did not undergenerate one would need to prove that every score in the style being modelled was well-formed. But no matter how many acceptable scores and corpus scores one showed to be well-formed, one could never be sure that there did not exist *any* ill-formed scores in the style until one had tested every score in the universal set of scores. This would be an impossible task because the universal set of scores is an infinite set. Thus Baroni’s claim that ‘it is possible to verify the completeness of a grammar’ is *not* true: an algorithmic style theory is an unverifiable hypothesis.

Baroni’s claim that ‘the quality of the musical examples produced by the computer is a test of the completeness and correctness of [a] musical grammar’¹⁰⁹ is, strictly speaking, also incorrect because a composing program implementation of a musical grammar can only be used to test for overgeneration. It can never be used to test for undergeneration and it can never be used to *verify* that a grammar does not overgenerate. It can be used, however, to show that a grammar *does* overgenerate. As Snell has pointed out,

establishing in an absolute sense whether a formal system generates ‘only’ tonal music [or ‘only’ music in the style being modelled] would entail, in principle, generating every piece it possibly could, and evaluating each in some way using human judgement—say, using a panel of non-communicating referees.¹¹⁰

¹⁰⁷ Baroni et al. 1984, 203–4.

¹⁰⁸ Baroni et al. 1989, 23.

¹⁰⁹ Baroni, Dalmonte and Jacoboni 1992, 201.

¹¹⁰ Snell 1979, 58.

Baroni suggests that ‘when, eventually, the musical production of the program is considered satisfactory, the computer has performed the role of the instrument used for the verification of the completeness and correctness of the whole grammar’¹¹¹ and asserts that ‘work stops when all the phrases produced or producible are judged to be sufficiently correct.’¹¹² But in order to be sure that a composing algorithm *only* generates scores in the style being modelled, one would need to have generated every member of its universal output set, which would be practically impossible. In my view, one should continue to test a style theory for overgeneration and undergeneration until one has either verified or refuted it. Given that it would be impossible to verify a style theory, the most one can hope for is that eventually one develops a theory that one is unable to refute.

¹¹¹ Baroni et al. 1992b, 600.

¹¹² Baroni and Jacoboni 1983, 1.

4 Mistaken notion of what would be required of an acceptability algorithm

A number of musicologists have, in my opinion, misunderstood what would be required of a satisfactory acceptability algorithm. These musicologists seem to believe that an acceptability algorithm should be a method for deriving algorithmically from *only* the information in a score, a decision as to whether or not the score is in a particular style. However, as I have pointed out, a satisfactory acceptability algorithm would have to take the form of an experimental procedure along the lines of a Turing test which defines a piece to be within a particular style if and only if *subjects* are unable, in general, to pick it out from a set of pieces known to be in the style.

For example, Snell suggests that a ‘purpose of a theory of tonality’ might be ‘to define tonal music theoretically, so as to provide a formal procedure for deciding whether a piece is tonal’ but adds that this application would be ‘fairly minor, being limited mostly to analyses of borderline non-tonal works.’¹¹³ I think this reflects a basic misunderstanding of the purpose that should be served by a theory for a musical style and an acceptability algorithm. A theory of tonality could not possibly serve as a formal procedure for *deciding* whether or not a given piece was tonal. Similarly, a theory for a musical style could never serve as a procedure for *deciding* whether or not a given piece was in the style that the theory was intended to model. This would render the theory circular and irrefutable. A theory of tonality or a theory of a particular musical style could only ever serve as a tool for *predicting* whether or not a piece would be considered tonal or would be perceived to be in the style being modelled. If a piece is not a member of the corpus of the style being modelled, then one can only *decide* whether or not this piece is in the style by using an acceptability algorithm along the lines of a Turing test as described above. If this algorithm is defined following the guidelines given above then it would not be possible for any piece to be a ‘borderline case’ since for any given piece, such an acceptability algorithm would always generate a decision as to whether or not the piece was in the style being modelled.

Snell’s erroneous idea that a theory of tonality might serve as a tool for deciding whether or not a given piece is tonal may derive from a similar mistake made by Chomsky. Chomsky incorrectly suggests that an operational definition of grammaticalness ‘must be tested for adequacy ... by measuring it against the standard provided by the tacit knowledge that it attempts to specify and describe’ and that ‘a proposed operational test for, say, segmentation into words, must meet the empirical condition of conforming, in a mass of crucial and clear cases, to the linguistic intuition of the native speaker concerning such elements.’¹¹⁴

This reveals a basic misunderstanding of what would qualify as a suitable operational definition of grammaticalness or segmentation into words. Chomsky appears to be implying that an operational test for grammaticalness would consist of an algorithm that takes an utterance as input and generates as output a judgement as to whether or not the utterance is grammatical such that the judgement is a direct function of the *structure of the utterance alone* and not of any empirically observable *behaviour* exhibited by humans that is defined to be indicative of whether or not they consider the

¹¹³ Snell 1979, 58.

¹¹⁴ Chomsky 1965, 19.

utterance to be grammatical. But such an algorithm would *not* qualify as a definition of grammaticalness equivalent to an acceptability algorithm in an algorithmic style theory.

A suitable operational test would be a specification that a given utterance is *defined* to be ‘grammatical’ if and only if native speakers exhibit certain specified behaviour in some well-defined class of situations in which they are *defined* to be exercising their intrinsic competence or intuitive capacity to decide whether or not the utterance is ‘grammatical.’

In my opinion, an algorithm that takes an utterance as input and generates a judgement of grammaticalness as output where the judgement is merely a function of the structure of the utterance itself might constitute a *model* of the human capacity to judge whether or not a sentence is grammatical, but would not be a satisfactory *explication* of the concept of grammaticalness. That is, it could not be used to provide an appropriate, independent, operational definition of the set of all and only those sentences that a grammar is intended to be able to weakly generate.

5 Parsers

The concept of a parsing algorithm is essentially identical to Kippen and Bel's (1992) notion of a 'membership algorithm' which is a device that they use 'to check whether or not a sentence entered into the editor is consistent with [their] grammar.'¹¹⁵

Another example of a parsing algorithm is Michael Kassler's (1975) program which he describes as a 'decision procedure' that not only determines 'whether or not a presented musical composition is an assertion' in the formalized language defined by his explication of Schenker's theory, but also produces

for every given assertable composition of the final formalized language a step-by-step derivation of the composition from an *Ursatz* that constitutes a structural analysis (literally, a structural synthesis) of the composition.¹¹⁶

Kassler's program takes as input a three row 'matrix' representing the pitch classes of the notes of a putative middleground structure and generates as output either a possible derivation of the matrix in terms of Kassler's formalized explication of Schenker's middleground theory or a statement such as 'NOT A THEOREM. COMPOSITION LACKS ACCEPTABLE HEADNOTE', if the matrix cannot be derived in terms of the theory.¹¹⁷ The program 'provides one proof for any given theorem, rather than all possible proofs.'¹¹⁸

On the flyleaf of his IBM research report on the CHORAL project, Ebcioğlu (1987) acknowledges his debt to his 'former advisor John Myhill for getting [him] interested in the mechanization of Schenkerian analysis.'¹¹⁹ This firmly places Ebcioğlu in the tradition of Rothgeb, Smoliar, Kassler and Snell. However, Ebcioğlu's motivations for attempting to explicate Schenker's theory seem to have been very different from his precursors. Rothgeb and Kassler explicated traditional theories in order to test them. Smoliar and Snell were motivated by a desire to achieve a deeper understanding of the structural principles of tonal music. Ebcioğlu, on the other hand, seems to have been primarily motivated by a fascination with the possibility of writing a computer program that automatically composes tonal music of a consistently high quality.

Ebcioğlu states that he 'had originally hoped to take the ... approach of top-down Schenkerian synthesis of a musical surface.' However, in practice,

this approach was later deemed to be impractical because it involves making commitments at an early program stage without knowing what these commitments will exactly lead to, which can cause unnecessary backtracking when attempting to meet local constraints later on.¹²⁰

He therefore decided to concentrate on 'the analysis rather than synthesis of the surface structure of a musical piece'¹²¹ and the Schenkerian component of his CHORAL program consists of a bottom-up parser that, like Kassler's program, automatically

¹¹⁵ Kippen and Bel 1992, 211.

¹¹⁶ Kassler 1975, 6–7.

¹¹⁷ Kassler 1975, 20–21.

¹¹⁸ Kassler 1975, 24.

¹¹⁹ Ebcioğlu 1987b, acknowledgements.

¹²⁰ Ebcioğlu 1987b, 92.

¹²¹ Ebcioğlu 1987b, 92.

generates pseudo-Schenkerian analyses of the descant and bass lines of a chorale directly from an encoding of the score.¹²²

Ebcioğlu's program therefore achieves considerably more than those of Smoliar and Snell which are intended merely to allow a user to construct all and only legal Schenkerian analyses but do not automatically generate such analyses from representations of scores.

Snell, however, shows considerable interest in 'building the equivalent analytical system' to his derivational program, that would embody 'a process whose input is a score and whose output is its preferred structural description' in terms of his explication of Schenker's theory.¹²³ As he points out, 'there would be no difference between the two systems in the musical knowledge they embodied.'¹²⁴

As explained above, in order to test an algorithmic style theory for undergeneration, one needs a derivation algorithm as well as a parsing algorithm. This is because the only sure way of checking the correctness of a derivation d generated by the parsing algorithm for a representation r , is to give d to the derivation algorithm as input and verify that the derivation algorithm generates r as output. It is clear that Snell also recognizes the need for both parsing and derivation algorithms to test a style theory objectively for undergeneration. For example, he remarks that,

with both the analytical and derivational systems available in the form of programs, it would become possible to input a score and have it analyzed and then re-derived without human intervention. This possibility is attractive on scientific-philosophical grounds because it would allow completely impartial theory-testing.¹²⁵

However, it is important to note that only certain restricted classes of grammar can be implemented as equivalent parsers. For example, it is not clear from what he writes whether or not Steedman (1984) implemented his grammar as a composing program, but Mouton and Pachet (1995) suggest that it would actually have been theoretically impossible for Steedman to have implemented his grammar as a *parsing* program, claiming that 'the mere presence of context-dependent rules makes his model not suitable for implementation, and therefore can only be useful in a 'contemplative mode.'¹²⁶ This seems to contradict Steedman's concern that the rules of his grammar should satisfy certain 'necessary conditions for [them] to be reversible and for a processing algorithm to exist for the grammar.'¹²⁷

¹²² Ebcioğlu 1987b, 88.

¹²³ Snell 1979, 66.

¹²⁴ Snell 1979, 66.

¹²⁵ Snell 1979, 66.

¹²⁶ Mouton and Pachet 1995, 33.

¹²⁷ Steedman 1984, 63.

6 The concept of a representation algorithm

6.1 *A representation algorithm does not need to be computationally implementable*

The process of testing an algorithmic style theory for undergeneration is essentially that of trying to find an ill-formed score that is a member of the style. To prove that a given score s is ill-formed, one first uses the representation algorithm to generate the representation of s and then one gives this representation to the parsing algorithm as input. The parsing algorithm will be unable to generate a well-formed derivation if the representation is ill-formed. But clearly, the fact that the representation of s is ill-formed only proves that s is ill-formed if the representation is definitely the *correct* representation of s . This shows that the definition of the representation algorithm must be sufficiently explicit for there never to be any doubt as to whether or not any given representation is the correct representation of any given score.

It might be suggested that in order to achieve this level of certainty it would be necessary to implement the representation algorithm as a computer system able to parse a visual image of a Standard Notation score into the symbols (e.g. notes, staves, accidentals, key signatures, time signatures, bar-lines etc.) that a musician who is familiar with this class of scores automatically identifies when he or she reads such a score. Unfortunately, such a program would have to include procedures for automatic optical recognition of scores and there are still some difficult research problems that need to be solved before such procedures could be incorporated into a representation algorithm that was required to define a unique and correct representation for any given input score.¹²⁸ I therefore do not think that the representation algorithm needs to be completely implementable as a working computer program. I think it is only necessary to specify that the representation algorithm must be defined sufficiently precisely for there to be no reasonable doubt that any two humans correctly following the instructions comprising the representation algorithm will produce an identical, correct representation for any score. It follows from this that the representation of a score should contain no information that cannot be inferred algorithmically from the score.

6.2 *A representation algorithm needs to preserve correct diatonic spelling of pitches*

As explained above, because the score algorithm of an algorithmic style theory is used to generate the scores presented to subjects in the test trials of the acceptability algorithm, it must generate corpus scores from corpus representations and this sets a lower limit on the amount of information in a score that the representation algorithm must preserve in the representations that it generates. In particular, it implies that representations must contain information indicating *explicitly* how pitches are spelt diatonically in the input score. In other words, enharmonically equivalent pitches that are spelt differently in a score (e.g. B-flat and A-sharp) should *not* be represented in the same way in the representations generated by the representation algorithm. If all the enharmonically equivalent pitches in a corpus score s were represented identically in its representation r then the score algorithm would not necessarily generate a score

¹²⁸ See Alphonse *et al.* 1988, Carter and Bacon 1990, Clarke, Brown and Thorne 1990.



Figure 6-1

notationally equivalent to s from representation r , because to do so it would have to incorporate a correct and complete *theory* of how to derive the correct diatonic spelling and tonal function of a note within in its key from a representation of the note that indicates only its ‘pitch class’¹²⁹ or ‘chroma.’¹³⁰ For example, if the pitches of enharmonically equivalent notes were represented identically in the representations generated by a representation algorithm then the scores in Figure 6-1 and Figure 6-2 would be representationally equivalent.

Figure 6-1 shows part of a score of Bach’s chorale, ‘Christus, der ist mein Leben’ (BWV 281, no.6 in Bach 1990). The set of all and only correct keyboard performances of the score in Figure 6-2 is identical to that of the score in Figure 6-1. However, most musicians familiar with Bach’s music would consider the score in Figure 6-2 to be an incorrect way of expressing the music represented by the score in Figure 6-1. In fact, a performer who was familiar with Western tonal music would be very surprised if Bach had chosen to notate this music in any way other than that shown in Figure 6-1. This demonstrates that, in general, although there are many distinct possible ways of representing in a Standard Notation score the sequence of actions that must be taken to perform a piece of Western tonal music, only a very small number of these possible scores would be considered correct by a musician that was familiar with tonal music. A Standard Notation score therefore represents much more than a sequence of actions. It also partially represents how the composer intends the piece to be *interpreted*.

As a further example, imagine that one had two scores of Chopin’s Ballade in G minor that were identical except that all the notes spelt as B flat in one were changed in



Figure 6-2

¹²⁹ See Forte 1973 for a definition of the concept of ‘pitch class’.

¹³⁰ See Deutsch 1982 for a definition of the concept of ‘chroma’.

the second to A sharp. On a piano, a performance from the second score would sound identical to one from the first. But when a note is written as a B flat in a passage in G minor, this implies that the composer intends the note to be interpreted as a mediant in that context and if the same note had been spelt as an A sharp, this would have implied that the composer intended the note to be interpreted as a sharpened supertonic. A score of the Ballade in G minor in which all the B flats had been changed to A sharps would therefore be a very poor representation of how an audience would interpret the piece and a performer would have great difficulty in interpreting all those notes spelt as A sharps as sharpened supertonic in G minor, or sharpened leading notes in B flat major and so on.

Cope's EMI system generates output in the form of performances or as print-outs in a 'numeric code'¹³¹ that contain at least enough information to enable a performance of the output piece by a MIDI synthesizer since the program does actually generate such performances. However, in order to *reliably* generate correct Standard Notation scores, the representations used by EMI would have to include information about the enharmonic spelling of notes (for example, whether a MIDI note number 58 should be written as a B flat or an A sharp in any given situation). It is not clear from Cope's writings whether or not his 'numeric code' output contains the necessary information to be able to mechanically produce Standard Notation scores with complete reliability.

Similarly, Kassler 'presupposes enharmonic equivalence—e.g., of D-sharp and E-flat in the same octave'¹³² in his computational explication of Schenker's middleground theory. But, as Snell remarks,

it is 'quite strange' ... that Kassler uses a representation of pitch that fails to indicate diatonic status—that is, a notation equivalent to 12-tone pitch-classes (plus octave indication)

especially as his 'project is intended to explain diatonic music.'¹³³ Kassler attempts to justify his decision by claiming that

those who would question this presupposition can be reminded of its inherence in the design of keyboard and musical instruments for which very many compositions instancing tonality have been written.¹³⁴

But as Snell points out,

not only does [this] fail to provide an argument, but it is actually misleading. The design of the modern keyboard constitutes a concession to the anatomy of the human hand, and to economics and engineering, and has no consequences concerning the use of enharmonic equivalence in tonal theory.¹³⁵ Longuet-Higgins and Steedman have shown in two model papers¹³⁶ ... that the problem of restoring diatonic status, given tonal melodies represented in a pitch-class-like notation, involves a number of subtle issues, not all completely understood.¹³⁷

Kassler's decision to adopt a representational system in which enharmonically equivalent but diatonically distinct pitches are represented identically implies that each

¹³¹ Cope 1991, 21.

¹³² Kassler 1975, 8.

¹³³ Snell 1979, 10.

¹³⁴ Kassler 1975, 8.

¹³⁵ Snell refers to Helmholtz 1954, 466–482.

¹³⁶ Snell refers to Longuet-Higgins 1976 and Longuet-Higgins and Steedman 1971.

¹³⁷ Snell 1979, 11.

parsing generated by his program corresponds to an infinite set of enharmonically equivalent but diatonically distinct middleground graphs of which only one is actually intended. There are at least two ways in which Kessler could have avoided this problem:

1. he could have adopted a representational system in which the diatonic spelling of notes was represented as well as their pitch height and pitch class;
2. he could have supplemented his theory with an algorithm that, when given the pitch height of a note (i.e. a MIDI note number) and the context of the note as input, always generates as output the correct diatonic spelling of that note.

Clearly, the first solution is by far the easier of the two. To take the second tack would involve solving a formidable research problem in its own right (as evidenced by the papers by Longuet-Higgins and Steedman cited by Snell). Indeed, this research problem might not even be perfectly soluble *in principle*: it might actually not be possible to model algorithmically the process by which a human musician determines the correct diatonic spelling for any given note in a tonal context. There is clearly no a priori reason to assume that all natural processes can be completely described algorithmically. Indeed, Penrose (1994) asserts that there are even ‘certain types of *mathematically precise* activity [my emphasis] that can be *proved* to be beyond computation’. For example, no computational procedure exists ‘for deciding, for a given system of *Diophantine* equations, whether the equations have any common solution.’¹³⁸ Thus, it might actually be impossible *in principle* to characterize correctly by means of an algorithm the procedure by which one determines the correct diatonic spelling of a note in a tonal score given only the MIDI note numbers of the notes in the score.

To reliably determine the correct enharmonic spelling of a MIDI note number in a tonal context, one would at least need to be able to determine algorithmically the key or tonality at every point in a piece. As Cross has noted,

when a particular inflectional spelling is employed in a piece of tonal music it almost always implies allusion to a particular tonal region, to a global property of the specific piece in which it’s employed.¹³⁹

Unfortunately, an algorithm that was capable of identifying *reliably* and *intelligently* (as opposed to statistically) the tonality at each point in a piece of tonal music would need to employ a successful, complete and explicit theory of tonality. So, as Snell says, ‘the problem of restoring diatonic status, given tonal melodies represented in a pitch-class-like notation’ is in fact a rather interesting and complex problem that has not yet been completely solved.

Fortunately, however, to produce an algorithmic style theory, it would not be necessary to solve this problem. It would only be necessary to devise a representation algorithm that generates representations in which the diatonic spelling of notes in a score is explicitly represented.

¹³⁸ Penrose 1994, 28.

¹³⁹ Cross 1995a.

6.3 *Musicologists have not paid sufficient attention to the concept of a representation algorithm*

David Cope has called his EMI system a ‘computer model of music composition.’¹⁴⁰ EMI takes as input a set of pieces and generates as output an automatically composed piece that is intended to be ‘in the style of’ the input pieces. EMI can therefore readily be construed to be a computational model of the human skill of ‘pastiche’ composition where a musician studies a number of scores of pieces, perhaps in the same genre by the same composer, and then attempts to compose new pieces ‘in the style of’ these pieces. Typically, in this process, the musician will express the new piece as a score. A model of the human skill of pastiche composition would therefore take the form of a device that takes *scores* of pieces as input and generates *scores* of automatically-composed pieces as output. However, Cope’s EMI system does not take scores directly as input. One would therefore expect that, if EMI were intended to be a computational model of pastiche composition, then Cope would have defined a representation algorithm. No such definition appears explicitly in his published writings. However, if there is no reasonable doubt that any two individuals deriving a representation of the same score for use as input to EMI would derive identical representations, then I think Cope could justifiably claim that EMI effectively takes scores as input. Unfortunately, there is some evidence that for the correct operation of EMI it is necessary to provide information about a score that cannot be algorithmically inferred from it and which is dependent upon an individual’s interpretation of the piece. This implies that there can be no guarantee that two individuals will derive the same representation for a given score. Therefore it cannot justifiably be claimed that the program takes scores as input and so the program does not correctly simulate this aspect of the human skill of pastiche composition.

In EMI, works are encoded a voice at a time as sets of phrases, each phrase being represented by a list of notes and a list of durations.¹⁴¹ It is therefore clear that, for the correct operation of EMI, it is necessary to provide information about a score that cannot be algorithmically inferred from it. Also, Cope blatantly admits that ‘works have to be diligently analyzed and fed to the computational programs in order for composition to occur.’¹⁴² This strongly suggests that the representation of a given piece used as input to the program will depend upon the way the program operator parses the score ‘vertically’ into voices and ‘horizontally’ into phrases and will contain information about these parsings. In the case of scores where the composer has not indicated phrases by means of phrase marks (for example, the scores of keyboard works by Bach—a repertoire whose style Cope has tried to simulate with EMI), the operator must parse a phrase structure for the piece using, for example, his or her ability to recognize cadences. Because the process of parsing a phrase structure from a score without phrase marks is a complex and subjective process, there can be no guarantee that, in general, any two suitably qualified individuals will always consider any single piece to possess the same phrase structure. Indeed, even a single individual may consider that there are two or more equally feasible phrase structures for a single piece. A similar situation can arise for the ‘vertical’ parsing of a score into voices where the voice structure is not explicitly represented in the score (for example, in Brahms’ piano pieces—another repertoire that Cope has studied with EMI). The extent to which the

¹⁴⁰ Cope 1993, 403.

¹⁴¹ Cope 1991, 92–98, 151–3.

¹⁴² Cope 1989, 125.

quality of the pieces composed by EMI depends upon the manner in which the operator of the program manually and non-algorithmically produces the input representations is not clear from Cope's writings. But the fact that these representations are produced non-algorithmically makes it difficult to judge the success of EMI as a computational model of the human skill of pastiche composition.

7 The concept of a composing algorithm

7.1 *To test for overgeneration, an algorithmic style theory must have a composing algorithm*

Lerdahl and Jackendoff claim that because

the early work in the field [of transformational generative grammar], such as Chomsky 1957 and Lees 1960, took as its goal the description of ‘all and only’ the sentences of a language, ... many were led to think of a generative grammar as an algorithm to manufacture grammatical sentences.¹⁴³

They claim that certain musicologists were consequently misled into believing that ‘a musical grammar should be an algorithm that composes pieces of music.’¹⁴⁴ Thus, for example, whereas in Kassler’s view a satisfactory theory for a musical ‘language’ would need to take the form of an ‘intelligent music-processing machine’ able to compose automatically ‘coherent new utterances’ both ‘within a particular musical language, and even within a particular musical ‘style’ that is a dialect of such a language,’¹⁴⁵ Lerdahl and Jackendoff consider that such a system ‘would be utterly unrevealing from a psychological standpoint’ and that “‘generating” trivial musical examples says nothing about how people hear.’¹⁴⁶

Lerdahl and Jackendoff cite Sundberg and Lindblom (1976), Kassler (1963), and Smoliar (1974) as examples of researchers who had made the error of thinking that writing algorithms that compose pieces of music constituted applying the methodology of generative linguistics to the domain of music theory. Lerdahl and Jackendoff contrast these researchers with Winograd (1968) whose “‘analytic” approach to musical grammar’ they claimed was ‘in some ways more like [theirs] than the “synthetic” grammars’¹⁴⁷ developed by the others.

In fact, neither Smoliar’s program nor Kassler’s was an implementation of ‘an algorithm that composes pieces of music’¹⁴⁸ and there is no evidence from their published writings that Sundberg and Lindblom actually implemented their grammar as a computer program at all. Smoliar describes his program as a ‘computer aid for Schenkerian analysis.’¹⁴⁹ Far from being ‘an algorithm that composes pieces of music,’ it is intended to be a software tool that allows an analyst to construct all and only well-formed Schenkerian analyses. The program described in Kassler 1975 is intended to be a computational explication of Schenker’s middleground theory in the form of a program that automatically *parses* all and only well-formed Schenkerian middlegrounds. Sundberg and Lindblom’s grammar was intended to be a neutral embodiment of the necessary and sufficient knowledge required to create *and understand* all and only those pieces in the style of Alice Tégner’s Swedish folksongs.

¹⁴³ Lerdahl and Jackendoff 1983, 6.

¹⁴⁴ Lerdahl and Jackendoff 1983, 6.

¹⁴⁵ Kassler 1975, 2.

¹⁴⁶ Lerdahl and Jackendoff 1983, 111.

¹⁴⁷ Lerdahl and Jackendoff 1983, 333–334, note 7.

¹⁴⁸ Lerdahl and Jackendoff 1983, 6.

¹⁴⁹ Smoliar 1980, 41.

Now it is certainly true that a generative grammar is *not* ‘an algorithm to manufacture grammatical sentences.’¹⁵⁰ Rather, as Chomsky states, a generative grammar must ‘attempt to characterize in the most neutral possible terms the knowledge of the language that provides the basis for actual use of language by a speaker-hearer.’¹⁵¹ However, most grammars can quite readily be implemented as ‘generator’ algorithms that generate random (or, at least, arbitrary¹⁵²) examples from the artificial languages that they define. Some grammars can also be implemented as parsers, that is, devices that take sentences as input and determine whether or not these sentences are members of the artificial languages defined by the grammars. Thus, while it may be possible to *implement* a generative grammar as algorithms for generating and parsing utterances in a language, it is nonetheless true that a generative grammar is simply ‘a set of rules, which, in themselves can do nothing’¹⁵³ as Johnson-Laird has pointed out.

But Lerdahl and Jackendoff seem to deny the necessity for implementing a musical grammar as an algorithm that generates random examples of pieces from the ‘artificial language’ that the grammar generatively defines. In my opinion, to say that one is not interested in implementing a musical grammar as a composing algorithm is equivalent to stating that one is not interested in whether or not the grammar overgenerates. In other words, it implies that one holds the clearly untenable view that a theory for a musical style need be no more than the trivial hypothesis that some generatively defined set contains all of the pieces in a style—and possibly lots of other pieces as well that are *not* in the style.

Lerdahl and Jackendoff seem to be essentially uninterested in characterizing musical styles. They seem only to be interested in analysing *existing* pieces of tonal music. Similarly, Snell claims that because ‘the main function of a theory of tonality is to help in understanding the structure of pieces already known to be tonal’¹⁵⁴ it does not matter particularly whether or not a musical grammar weakly generates the style that it is intended to characterize. In Snell’s opinion, what matters above all is that the artificial language defined by the grammar contains known existing pieces in the style. In other words, Snell’s claim is that if one is primarily interested in achieving richer interpretations of existing pieces of music in a style then it is more important that one’s grammar does not undergenerate than that it does not overgenerate.

But achieving a rich understanding of a piece of music depends upon being able to perceive the piece in the context of other works by its composer, other works in the same genre, other works from the same period and so on. That is, achieving a rich interpretation of a piece of music is above all a matter of identifying those features of the piece that are *only* possessed by members of those classes of pieces of which it is naturally seen to be a member. Therefore, if one’s goal in producing a grammar for a given style is to achieve a deeper understanding of existing pieces in that style, then it is of the utmost importance that one’s grammar characterizes correctly those features that are *only* possessed by pieces in the style and thus does *not* overgenerate.

¹⁵⁰ Lerdahl and Jackendoff 1983, 6.

¹⁵¹ Chomsky 1965, 9.

¹⁵² The problem of producing a composing algorithm that generates a truly random sample from the universal set of well-formed representations will be discussed below.

¹⁵³ Johnson-Laird 1991, 297.

¹⁵⁴ Snell 1979, 58.

7.2 *Composing requires more knowledge than listening therefore style theories should be theories of composition*

As Lyons has remarked, a generative grammar ‘is intended to be neutral as between production and reception, to a certain extent explaining both, but no more biased towards one that it is towards the other.’¹⁵⁵ Strangely enough—given their abhorrence of ‘algorithms to manufacture grammatical sentences’—Lerdahl and Jackendoff also recognize that

generative linguistic theory is an attempt to characterize what a human being knows when he knows how to *speak* a language, enabling him to understand *and create* an indefinitely large number of sentences, most of which he has never heard before [my italics].¹⁵⁶

However, most people find it much easier to *understand* sentences in a foreign language than they do to *create* correct new sentences of their own. This suggests that in order to be able to *understand* a language, one needs a far less detailed and complete knowledge of the language than one needs to be able to *speak* it. Indeed, I think one can go so far as to say that whereas the knowledge required to speak a language is generally sufficient to enable the speaker also to understand it, it is certainly not true that the knowledge required to understand a language is sufficient to be able to speak it. Similarly, far more knowledge is required to be able to *compose* an acceptable piece of music in a style than is required to be able to identify that a given piece is in the style and achieve an understanding of that piece.

Therefore, in my view, a theory that embodies sufficient knowledge to *understand* pieces in a musical idiom would not necessarily embody sufficient knowledge for *composition* of pieces in the idiom. On the other hand, a theory that embodied the necessary and sufficient knowledge required to be able to *compose* all and only the pieces in a musical style would certainly embody the knowledge required to be able to achieve complete interpretations of all pieces in the style. Therefore, in my view, a theory for a musical style should embody the necessary and sufficient knowledge required to *compose* all and only pieces in the style. A theory that only embodies sufficient knowledge for *understanding* or *listening to* pieces in a style would not necessarily be satisfactory.

Lerdahl and Jackendoff, however, state that

in [their] view a theory of a musical idiom should ... [take the form] of an explicit formal musical grammar that models the *listener's* connection between the presented musical surface of a piece and the structure he attributes to the piece [my italics],¹⁵⁷

and that they

take the goal of a theory of music to be a *formal description of the musical intuitions of a listener who is experienced in a musical idiom* [their italics].¹⁵⁸

Lerdahl and Jackendoff explain that by ‘the musical intuitions of the experienced listener’ they mean

¹⁵⁵ Lyons 1977, 42.

¹⁵⁶ Lerdahl and Jackendoff 1983, 5.

¹⁵⁷ Lerdahl and Jackendoff 1983, 3.

¹⁵⁸ Lerdahl and Jackendoff 1983, 1.

the largely unconscious knowledge ... that enables him to ... identify a previously unknown piece as an example of the idiom, to recognize elements of a piece as typical or anomalous, to identify a performer's error as possibly producing an 'ungrammatical' configuration, to recognize various kinds of structural repetitions and variations, and, generally, to comprehend a piece within the idiom.¹⁵⁹

In other words, in Lerdahl and Jackendoff's opinion, a theory of a musical idiom certainly does *not* need to embody sufficient knowledge to allow for the *composition* of pieces in any particular styles.

But clearly, a system that embodies sufficient knowledge to allow for the derivation of partial descriptions of interpretations from musical surfaces does not necessarily embody sufficient knowledge to be able to reconstruct the *whole* of those musical surfaces *in detail*, or, even more demandingly, generate all and only those musical surfaces that are in a given style. So, in 'focusing on the listener because listening is a much more widespread musical activity than composition or performing,'¹⁶⁰ Lerdahl and Jackendoff were in my opinion setting their sights rather lower than they could have been.

7.3 *Understanding is a process of reconstruction—a theory of listening should embody sufficient knowledge for complete reconstruction of scores*

Dennett notes that human vision

cannot be explained as an *entirely* 'data-driven' or 'bottom-up' process, but needs, at the highest levels, to be supplemented by a few 'expectation-driven' rounds of hypothesis testing (or something analogous to hypothesis testing).

He also mentions that the 'analysis-by-synthesis' model of perception

supposes that perceptions are built up in a process that weaves back and forth between centrally generated expectations, on the one hand, and confirmations (and disconfirmations) arising from the periphery on the other hand,¹⁶¹

and cites Neisser 1967 as a source for this idea. He goes on to make explicit that

the general idea of these theories is that after a certain amount of 'preprocessing' has occurred in the early or peripheral layers of the perceptual system, the tasks of perception are completed—objects are identified, recognized, categorized—by generate-and-test cycles. In such a cycle, one's current expectations and interests shape hypotheses for one's perceptual systems to confirm or disconfirm, and a rapid sequence of such hypothesis generations and confirmations produces the ultimate product, the ongoing, updated 'model' of the world of the perceiver.¹⁶²

Dennett points out that 'such accounts of perceptions are motivated by a variety of considerations, both biological and epistemological,' that 'experiments inspired by the approach have borne up well' and that 'some theorists have [even] been so bold as to claim that perception *must* have this fundamental structure.'¹⁶³

¹⁵⁹ Lerdahl and Jackendoff 1983, 3.

¹⁶⁰ Lerdahl and Jackendoff 1983, 7.

¹⁶¹ Dennett 1991, 12.

¹⁶² Dennett 1991, 12.

¹⁶³ Dennett 1991, 12.

This suggests that the process of understanding a piece of music may well be essentially that of formulating hypotheses as to the compositional decisions that led to each event being in the piece. In other words, the process of attempting to understand a piece of music is essentially that of trying to formulate an unrefuted hypothesis as to how the piece was *constructed*.

Therefore, in my opinion, the best type of theory of listening would essentially take the form of a parsing algorithm in an algorithmic style theory that takes a rather detailed representation of a score as input and generates as output an explicit description of how the composing algorithm of the theory could have generated this detailed score representation *in its entirety*. That is, the parsing algorithm should generate a complete and explicit description of how the composing algorithm and score algorithm could have generated a complete score that was notationally equivalent to the score given as input to the parsing algorithm.

Snell remarks, with respect to Lerdahl and Jackendoff's theory, that 'what we have to go on are the compositions, and we [should] seek to understand all the wealth the composer gives us in them, regardless of whether the 'idealized listener' can be reasonably expected to perceive it.'¹⁶⁴ Lerdahl and Jackendoff do however feel the need 'in dealing with especially complex artistic issues,' to 'elevate the experienced listener to the status of a 'perfect' listener—that privileged being whom the great composers and theorists presumably aspire to address.'¹⁶⁵ This suggests that perhaps they were aware that in seeking to characterize the intuitions of an experienced listener they were aiming to characterize rather less knowledge than would necessarily be required to generate all and only those pieces in the tonal idiom. A '*perfect* listener' is presumably one who achieves interpretations of pieces in the idiom that are sufficiently detailed and complete for him or her to be able to reconstruct complete scores of those pieces. In other words, a 'perfect listener' is presumably one that possesses sufficient knowledge to be able to *compose* all and only pieces in the idiom.

7.4 *Probability, determinacy and heuristics in a composing program*

The composing algorithm performs an essential function in the process of testing an algorithmic style theory for overgeneration. But in order to perform this function, the composing algorithm must be capable of generating a random sample of representations from the universal set of well-formed representations. This implies that the program should impose a *flat* probability distribution over the universal set of well-formed representations. That is, given a particular subset *s* of the universal set of well-formed representations of a theory system such that *s* contains *n* members, the probability of a particular output of the composing algorithm being a member of *s* should be exactly the same as the probability of its being a member of any other subset of the universal set of well-formed representations that contains *n* members.

However, Baroni points out that an implementation of a musical grammar as a composing program in which at each decision point an equal probability is assigned to each option does *not*, in fact, impose a flat probability distribution over the artificial language defined by the grammar:

It is tempting to give all possible choices an equal probability a priori. But in practice the rules of the grammar interfere strongly with each other, and it is clear

¹⁶⁴ Snell 1979, 15.

¹⁶⁵ Lerdahl and Jackendoff 1983, 3.

that even with equal probabilities for the different choices the results will depend on the way and on the order these choices are made.¹⁶⁶

Baroni gives an example of this phenomenon from his program, LEGRE, which takes a text of the type set by Legrenzi in his arias as input and generates as output, not a complete setting of the text but merely an appropriate large-scale structure indicating the pattern of repetition of the main sections in the aria together with the large-scale tonal structure. Each of the Legrenzi arias in the corpus that Baroni analysed follows one of the following large-scale sectional structures: ABC, ABCC, ABCBC or ABCABC. Therefore, to determine which of these structures the program will use for a particular text, one could either first choose ‘whether or not to repeat part of the poem and then choose which part to repeat’ or ‘choose among the four possibilities at once.’¹⁶⁷ As Baroni points out, the two strategies have completely different results. If the program first chooses with equal probability whether or not to repeat and then which part to repeat, then the resulting probabilities of the different sequences occurring are 50% for ABC and 16.7% for each of the other three. However, if the program simply chooses between the four forms in one step with equal probability given to each form, then all four will be generated with equal probability. Clearly which strategy is chosen will have a profound effect on the probability of any given output from the universal output set of the program being produced on any given execution of the program. This shows that in practice it might actually be very difficult to construct a composing algorithm that generates a truly random sample of examples from its universal output set.

Baroni suggests that one might attempt to solve this problem by performing a statistical analysis of features in the corpus and then setting up the probabilities of options at decision points in the program so that the probability of certain features appearing in the output reflects the frequency of occurrence of these features in the corpus. He proposes that ‘in setting up the computer program, at each point where a choice must be made we may use the probabilities that result from the analysis of the original corpus.’¹⁶⁸ But just as it would be difficult to ensure that the composing program imposed a *flat* probability distribution over its universal output set, so it would be difficult to ‘sculpt’ the probability distribution so that it was in accord with the relative frequencies with which features occurred in the corpus because ‘the rules of the grammar interfere strongly with each other.’ Moreover, this problem is exacerbated by the fact that ‘it is sometimes necessary to reject an ‘attempted’ melody, because it cannot fulfil requirements imposed by certain rules.’¹⁶⁹

Nonetheless, Baroni reports that in LEGRE 2,

whenever the program has to choose amongst different possibilities permitted by the rules, a random choice [was] made according to a probability distribution obtained by an adaptation of the actual occurrences in the sample and of the results of automatic production.¹⁷⁰

¹⁶⁶ Baroni et al. 1992, 200.

¹⁶⁷ Baroni et al. 1992, 200.

¹⁶⁸ Baroni et al. 1992, 200–1.

¹⁶⁹ Baroni et al. 1992, 200–1.

¹⁷⁰ Baroni et al. 1989, 42.

According to Baroni, the result of this strategy was that the program ‘tend[ed] to reproduce the style of Legrenzi.’ But he admits that, for the reasons outlined above, ‘this situation is realizable only with some approximation.’¹⁷¹

A number of other musicologists have either proposed or used a similar strategy. Snell, for example, finds it

intriguing to imagine the results if every option of [a] system were studied statistically and, on that basis, assigned a probabilistic decision rule, perhaps depending on many factors

and suggests that

such a stochastic system would ‘generate’ pieces *ad infinitum* within certain constraints [and then] the extent to which these matched the stylistic constraints attributed to the composer could be evaluated and improved.¹⁷²

Similarly, when Kippen and Bel’s grammar for tabla drum sequences reached ‘a point of stagnation where computer-generated variations were judged to be neither very good nor incorrect’ they decided that this could be solved by

attributing to each production rule a coefficient of likelihood (or weight) where the probability that certain generative paths would be chosen in preference to others could be examined.¹⁷³

But when writing a piece of music, a composer is not simply ‘more likely’ to make the compositional decisions that he or she actually does at any given point as would be suggested by this ‘cheat probabilities’ strategy proposed by Kippen and Bel, Baroni and Snell. I think it can reasonably be assumed that a composer who is writing a piece will *always* choose what he or she considers to be the *best possible* solution that he or she can think of in the time available at any given point in the compositional process. As David Levitt has remarked, although

there are conflicting goals, confusion and errors in musical decision making ... we can not make progress in understanding these if we model them as uniform thermal effects.¹⁷⁴

Thus the *frequency of occurrence* of a particular class of event or combination of events in a score is not important. What is important is characterizing the class of contexts in which a composer *decides* to employ that particular event or combination of events. In other words, theorists should not concern themselves with how often or rarely a composer uses a particular combination of events in his or her works. They should rather be concerned with formulating hypotheses as to *why* composers use particular combinations of events in particular contexts. As Lidov and Gabura point out,

we are most conscious of musical forces when they are vigorously resisted as, for example, in the ‘prelude’ to *Tristan and Isolde* where the tonic chord is continuously withheld, or in jazz where the ‘normal’ accents of meter are perpetually misplaced. Statistical methods cannot account for the coherence of these ‘improbable’ structures.¹⁷⁵

¹⁷¹ Baroni et al. 1992b, 607.

¹⁷² Snell 1979, 64–65.

¹⁷³ Kippen and Bel 1992, 228.

¹⁷⁴ Ames 1992, 73.

¹⁷⁵ Lidov and Gabura 1973, 140.

In his computational simulation of jazz improvisation, however, David Levitt—perhaps as an over-reaction to the use of what he calls ‘uniform thermal effects’ and what I call ‘cheat probabilities’—adopted a policy of ‘avoidance of pseudo-random number selection anywhere in the composition process.’¹⁷⁶ The consequence of this is that the program makes totally deterministic and often somewhat arbitrary decisions. For example, Ames notes that ‘when left with two equally suitable pitches, the program chooses the upper one’ and that

Levitt makes ‘no justification for this,’ except that it enables him to claim that his algorithm ‘lacks noise sources, and, with that one exception, is motivated’ exclusively by his theory of jazz.¹⁷⁷

But Levitt’s completely deterministic strategy is obviously an incorrect model of the human creative process of jazz improvisation since it implies that when given a melody and chord sequence, every human jazz performer in the style modelled by Levitt’s program will *always* produce exactly the same improvisation of that melody! Levitt’s hypothesis is patently absurd and is immediately refuted by the fact that even *individual* jazz performers generally do not repeat exactly the same performance every time they improvise on a given melody. Indeed, in proposing his completely deterministic model of jazz improvisation, Levitt is effectively making the ludicrous suggestion that there is *never* more than one acceptable solution to any given compositional problem.

Brown and Dempster similarly note that, like Levitt, there are those who ‘assert that if music theory had any real predictive power, then it should predict what particular notes will appear at any given point in a piece.’¹⁷⁸ But clearly, for any given piece there is generally a very large number of other pieces that are at least minimally different and that would have been equally acceptable and ‘in the style.’ Consider, for example, the fact that there are certain chorale melodies for which Bach produced a number of different harmonizations. Any theory that aimed to account for the structure of a given piece would also have to account for the fact that these other minimally different pieces would have been equally acceptable. A theory that predicted that each piece of music that had actually been composed was the *only* possible piece ‘of its type’ denies the possibility of composing new pieces of acceptable music in the style. It also denies the fact that people who are familiar with a sufficient number of pieces by a given composer or in a particular genre or from a particular historical period or geographical location, are often capable of correctly identifying the provenance of other, previously unheard pieces from the same period, or by the same composer or in the same genre.

Therefore, as Johnson-Laird has pointed out,

the use of arbitrary choices after all the constraints have been met is entirely consistent with a general theory of creativity. Even when all the constraints in the musician’s mind have been taken into account, a rich theory of creativity will be consistent with more than one possible next note. If not, then the theory will be deterministic, and the fecundity of musical virtuosi would be profoundly mysterious.¹⁷⁹

¹⁷⁶ Levitt 1993, 454.

¹⁷⁷ Ames 1992, 73.

¹⁷⁸ Brown and Dempster 1989, 87.

¹⁷⁹ Johnson-Laird 1991, 319.

It is important to make a very clear distinction between composing programs that employ ‘cheat probabilities’ (such as Kippen and Bel’s BOL Processor and Baroni’s LEGRE 2) and programs that employ *heuristics* such as Ebcioğlu’s CHORAL and Ames’ Cybernetic Composer.

As Ebcioğlu explains, heuristics ‘are rules that are to be followed whenever it is possible to follow them.’¹⁸⁰ As an example of a heuristic from his CHORAL program, Ebcioğlu gives that of preferring to continue a linear progression in the same direction. This heuristic essentially specifies that if at some decision point during an execution of the program, the absolute rules allow ‘two acceptable items, one which continues a linear progression, and one which does not’ then the program should ‘prefer [to use] the item which does.’¹⁸¹ It is important to note that heuristics are *breakable* rules and that, indeed, if certain heuristics were always satisfied then unacceptable results would be generated. For example, if the heuristic to continue a linear progression

were always satisfied, then one would get ascending or descending scales all over the harmonization. But it is not always satisfied, due to various absolute rules, resulting in a good overall musical effect.¹⁸²

Ebcioğlu claims that ‘the main advantage of heuristics vs. pure absolute rules and random search is [that]... heuristics lead the solution path away from a large number of unmusical patterns’¹⁸³ and that ‘if there were no heuristics, unmusical patterns would probably be generated by the bundle.’¹⁸⁴ He points out that ‘although it could be argued that absolute rules ... are theoretically enough to define a style,’¹⁸⁵ ‘attempting to do so results in an unwieldy proliferation of allowable, conditional violations’¹⁸⁶ and that therefore ‘there is conceptual and computational economy in using heuristics’ because the ‘synergy of ... heuristics and absolute rules would be complicated to express and probably slower to compute with absolute rules only.’¹⁸⁷ Absolute rules are therefore appropriate for formally expressing one type of knowledge, for example, a rule that a soprano part should not rise above C6, and heuristics are a good way of formally expressing another type of knowledge, for example, that it is good practice to continue linear progressions.

Interestingly, eight years before Ebcioğlu completed CHORAL, Snell proposed that it would be technically extremely inefficient (if not theoretically impossible) to correctly characterize the four-voice chorale style using absolute rules alone and that a preferable strategy would be to employ heuristics. Snell’s (1979) theory dealt only with the soprano and bass parts of a piece and in discussing how his theory could be developed to be able to account for the inner parts, he points out that

entirely new problems occur with the addition of new voices. Given the crucial soprano and bass lines, the movement of inner voices is very highly constrained by harmonic and contrapuntal conventions. Many of the latter kind are easy to specify as prohibitions (e.g., no parallel fifths or octaves), and can be realized systematically by a process of checking all possible combinations of two voices. But very many rules of this sort—producing to any great extent a ‘failure-driven’

¹⁸⁰ Ebcioğlu 1987b, 84.

¹⁸¹ Ebcioğlu 1996a.

¹⁸² Ebcioğlu 1996a.

¹⁸³ Ebcioğlu 1987b, 85.

¹⁸⁴ Ebcioğlu 1987b, 85.

¹⁸⁵ Ebcioğlu 1996.

¹⁸⁶ Ebcioğlu 1992, 317.

¹⁸⁷ Ebcioğlu 1996a.

system—would be undesirable; rather, as many positive heuristic (‘how-to’) rules as possible concerning voice-leading should be incorporated. To express formally even the basic heuristics—say, a college music-major’s competence in writing in four-part chorale style—is not a trivial task.¹⁸⁸

Similarly, Lerdahl and Jackendoff suggest that a theory such as their *GTTM* might in principle be achievable using only absolute rules of the type exemplified by their ‘well-formedness’ and ‘transformational’ rules but that in practice they ‘found such an approach counterproductive’ and discovered that ‘a different type of rule’—i.e. preference rules—were ‘more appropriate.’¹⁸⁹ Lerdahl and Jackendoff’s preference rules fulfil an essentially identical function in *GTTM* as Ebcioğlu’s heuristics do in *CHORAL*. In fact, it would probably be relatively straightforward to explicate and implement Lerdahl and Jackendoff’s preference rules as heuristics of the type used in *CHORAL*. Also, as Ebcioğlu points out, although heuristics can often be translated into absolute rules, generally it would take a large number of absolute rules to embody the knowledge expressed in a single heuristic. I therefore disagree with Peel and Slawson’s accusation that ‘the preference rules in Lerdahl and Jackendoff’s theory represent an admission of failure to develop an adequate corpus of well-formedness rules.’¹⁹⁰ To have expressed the knowledge embodied in Lerdahl and Jackendoff’s preference rules in the form of well-formedness rules would have led to an unbearably cumbersome theory. I therefore sympathize with their choice of the preference rule format for that knowledge embodied in this form in their theory, but it is a shame that they did not at least make some effort to formalize the way in which these preference rules interact.

It is clear from Ebcioğlu 1987b and Ames 1992, that a heuristic takes the set of options allowed at a decision point by a set of absolute rules and imposes an ordering of *merit* over this set of options. The program then generates a pseudo-random number and chooses *at random* from the set of options that have the highest *merit* at that point as defined by the heuristics and allowed by the absolute rules. For example, in *CHORAL*, ‘an alphanumeric encoding of the chorale melody, and a random number seed are given as input’ and then ‘random choice is used for breaking ties during heuristic evaluation.’ Ebcioğlu points out, however, that ‘there is often a single best choice due to the large number of heuristics’ and that therefore the program is ‘not very sensitive to the random number seed except in the beginning of the chorale, where all plausible starting positions are rated equally, and therefore chosen randomly.’¹⁹¹ Thus, like a human composer, a composing program that employs heuristics, *always* chooses one of the *best possible* solutions allowed by the absolute rules at any given point in the compositional process.

A ‘cheat probability rule,’ like a heuristic, takes the set of options allowed at a decision point by a set of absolute rules but then, instead of imposing an ordering of *merit* over this set of options, imposes an ordering of *probability* over them with the more ‘preferred’ options being given a higher *probability*. A program using cheat probability rules then generates a pseudo-random number and chooses from the *complete* set of options allowed by the absolute rules. The program will clearly be more likely to choose a preferred option because the preferred options are assigned higher probabilities. But when one employs cheat probability rules there is also always a

¹⁸⁸ Snell 1979, 61.

¹⁸⁹ Lerdahl and Jackendoff 1983, 39.

¹⁹⁰ Peel and Slawson 1985, 287.

¹⁹¹ Ebcioğlu 1987b, 129.

possibility of the program choosing an option with a *low* probability—that is, an option that has *low merit*. Therefore, whereas a program employing heuristics will, like a human composer, *always* choose from among the most preferred options available at a decision point, a program employing cheat probabilities will *sometimes* choose an option of very low merit (i.e. low probability) at a decision point—which is something a human composer would never do.

This implies that whereas the universal output set of a composing program that employs absolute rules and heuristics contains *only* pieces of the highest possible merit as determined by the program, the universal output set of a program that employs cheat probabilities contains pieces that are of very low merit—even as determined by the program. Also, whereas the probability of a composing program like CHORAL that employs heuristics generating any particular member of its universal output set is essentially *arbitrary* and depends only on the order in which the rules are applied, the effect of ‘cheat probabilities’ such as those that Baroni employs in LEGRE 2 is to impose an intentionally non-flat probability distribution over the set of pieces generated by the program so as to statistically favour the generation of pieces in which features occur with frequencies similar to those with which they occur in the testing corpus. Thus, Baroni's strategy is a deliberate attempt to make ‘favourable’ results happen *more often* but there is still a possibility of the program generating results that are even deemed to be of ‘low merit’ by the program itself. On the other hand, Ebcioğlu's heuristics *exclude* these unfavourable results altogether from the universal output set of CHORAL leaving an essentially arbitrary frequency distribution over the non-excluded solutions. In effect, the use of heuristics in CHORAL limits the universal output set of the program to a proper subset of the set generated by the absolute rules alone. Any harmonization generated by CHORAL is therefore one of the ‘best’ harmonizations that results from the continuous attempt to satisfy heuristics throughout the execution of the algorithm.

Therefore, in my view, there is nothing wrong with using heuristics, such as the ones used by Ebcioğlu and Ames, in the composing program of an algorithmic style theory system. However, I do not think that such a composing program should use ‘cheat probabilities rules’ such as those proposed by Baroni, Snell and Kippen and Bel, because changing the probabilities of the various options at decision points in the program does not actually change the universal output set of the program and intentionally imposes a non-flat probability distribution over the universal output set. Such a program would therefore certainly *not* generate truly random samples from this set and could thus not be used as a tool for testing an algorithmic style theory for overgeneration. Also, *ceteris paribus*, a program like CHORAL that employs absolute rules and heuristics is inherently a far more plausible model of the human skill of composition than one like LEGRE 2 that employs cheat probabilities or one that employs a completely deterministic strategy such as that used by Levitt in his jazz improvisation system.

8 Formality, computational implementability and Lerdahl and Jackendoff's Preference Rules

8.1 *Tautologies and representational systems are not theories—no amount of formalism makes a theory scientific*

Popper remarks that one 'of the things which put [him] out of step and which [he] likes to criticize' is the erroneous view that 'there is as much science in a subject as there is mathematics in it, or as much as there is measurement or 'precision' in it.' In Popper's opinion, this view is typically (but in most cases, unconsciously) held by social scientists and students of the humanities who attempt, ineptly, to apply what they believe to be 'scientific methods.' Popper points out that this idea 'rests upon a complete misunderstanding' and that 'on the contrary, the following maxim holds for all sciences: Never aim at more precision than is required by the problem in hand.'¹⁹²

Brown and Dempster echo Popper's point, asserting that 'formalism alone does not make some thing scientific any more than a suit makes a man' and that 'no amount of formalism can ever transform a description into an explanation.'¹⁹³ They claim that 'this confusion of merely formal systems with scientific theories is endemic to precisely those who are most widely perceived as endorsing the scientific ideal for music theory'¹⁹⁴ and cite Boretz's *Meta-Variations*¹⁹⁵ as an example of a music-theoretical work that is 'a formal system with descriptive potential' but that cannot be considered 'scientific' in any real sense because it has 'no empiric content and no predictive consequences.'¹⁹⁶ I agree with Brown and Dempster that 'since laws must have empiric content or have some bearing on the way the world is, they cannot be purely logical truths or stipulated definitions, since the truth of neither is empirically testable.'¹⁹⁷ I also agree that therefore 'music theory becomes scientific only when empiric laws are introduced and musical phenomena are subsumed under them in ways that guarantee predictions and testability.'¹⁹⁸

However, whilst I admit that no amount of formalism can ever convert a representational system into an explanation, I think it is nonetheless worth pointing out that achieving an understanding of some phenomenon is often no more than a matter of being able to describe it in terms of a system of description that can be applied to all and only phenomena of that type.

For example, Balzano (1980) discovered certain, highly suggestive, group-theoretical properties that are possessed only by those pitch class sets associated with major and minor chords and the diatonic scales. More recently (1991), I discovered that by re-expressing the group-theoretical properties of the diatonic set identified by Balzano in terms of graph-theory, it became clear that Balzano's group-theoretical property of the diatonic set was a special case of a graph-theoretical property possessed only by those pitch sets (as opposed to pitch class sets) associated with the traditional

¹⁹² Popper 1983, 7.

¹⁹³ Brown and Dempster 1989, 81.

¹⁹⁴ Brown and Dempster 1989, 81.

¹⁹⁵ Boretz 1969, 1970a, 1970b, 1971, 1972, 1973.

¹⁹⁶ Brown and Dempster 1989, 81.

¹⁹⁷ Brown and Dempster 1989, 69.

¹⁹⁸ Brown and Dempster 1989, 81.

major and minor scales (melodic and harmonic). This graph-theoretical property of scale-type pitch sets will be discussed in more detail in Part 2 of this thesis.

Such discoveries of mathematical properties possessed by the pitch collections employed in tonal music do not in themselves constitute ‘theories’ because they are not empirically refutable hypotheses—they are ‘merely’ interesting, logical truths. But they have value because they can potentially be used to construct new, improved theories that are more generalized and give more insight into the phenomena that they are designed to account for.

Nonetheless, it is true that to achieve the status of a ‘scientific’ theory, a statement has to be an empirically refutable hypothesis and for this reason, logical and mathematically proved truths are not scientific theories. For example, the hypothesis

All swans are white

is essentially a refutable one since it would be reasonably unambiguously refuted by the discovery of a swan that was not white. However, the refutability of this hypothesis depends upon the fact that being white is not a necessary condition on being a swan. As Popper remarks, if one redefined a swan so that being white was one of its definitive characteristics (e.g. ‘a bird is a swan if it is large, white, ...’) then, clearly, the hypothesis

All swans are white

would cease to be a scientific theory and would become a tautology. Popper concludes from this that one must ‘demand that anyone who advocates the empirical-scientific character of a theory must be able to specify under what conditions he would be prepared to regard it as falsified; i.e. he should be able to describe at least some potential falsifiers.’¹⁹⁹ And in particular, a scientific theory should not be a logical consequence of the definitions of the terms in which the theory is expressed.

This danger of tautology is avoided in an algorithmic style theory by specifying first, that the acceptability algorithm must be an experiment that relies on human judgements made by subjects who are not given any information that would trivially imply that the pieces in the test set are or are not in the corpus; and second, by stipulating that the composing algorithm should take no input other than possibly a sequence of random numbers. This guarantees that it will not be logically deducible that the universal set of well-formed scores defined by the composing algorithm is equal to the style as defined by the corpus and the acceptability algorithm.

8.2 *A theory should achieve a level of explicitness and precision sufficient for it to be computationally implementable*

Like Popper, Lerdahl and Jackendoff assert that ‘formalism alone is to [them] uninteresting except insofar as it serves to express musically or psychologically interesting generalizations and to make empirical issues more precise.’ They therefore ‘designed [their] formalism with these goals in mind, avoiding unwarranted overformalization.’²⁰⁰ In particular, Lerdahl and Jackendoff were ‘not ... concerned whether or not [their] theory could readily be converted into a computer program.’²⁰¹

¹⁹⁹ Popper 1983, p.xxi.

²⁰⁰ Lerdahl and Jackendoff 1983, 5.

²⁰¹ Lerdahl and Jackendoff 1983, 333, note 6.

This seems to suggest that in Lerdahl and Jackendoff's opinion, there is no point in aiming to develop a theory that achieves a level of explicitness and precision sufficient for it to be directly implemented as an equivalent computer program. But what possible reason could there be for not wanting a theory that allowed one to have as detailed an understanding of music as possible? Snell reports that when he asked Lerdahl at a conference why he and Jackendoff had not used computers, Lerdahl said

that he and Prof. Jackendoff had avoided them because, among other reasons, he believed that as soon as one attempts to express a theory in a completely unambiguous way (one effective means of which is to write a computer program), the theory loses something essential—just what this might be, he had no clue.²⁰²

As Snell points out, 'this point of view is of course inherently incompatible with their own stated idea of formal explication.'²⁰³

The requirement of computational implementability places a high lower limit on the level of precision and explicitness with which a theory must be formulated. Certainly, the source code of a computer program is generally a very poor way of representing a computational theory to a human being. The main reason for this is that there will in general be much in a program that is not directly part of the theory—for example, procedures for controlling input, output and memory management. So to this extent, a theory 'loses a certain something' when it is implemented as a computer program. But procedures for controlling input, output and memory management are specific to a particular computer implementation of a theory. A computational theory is not any particular implementation of the theory as a computer program. Rather it is the subset of the knowledge sufficient to produce a particular implementation of the theory that would be necessary to produce *any* implementation. In other words, it is the knowledge that is common to all possible implementations of the theory. This knowledge would clearly not in general need to include 'messy' details of memory management, input and output. Thus while the requirement of computational implementability certainly implies that a theory must achieve some minimum level of precision, detail and explicitness, it does not imply that the theory must include any specific details of how the theory could be implemented in some *particular* programming language or on some *particular* machine.

In sum, I essentially agree with Snell that it is desirable to implement theories as computer programs because first, 'the process of formalization itself tends to clarify existing concepts, potentially yielding "a far more precise and refined theory of the tonal system than we presently possess"'²⁰⁴,²⁰⁵ second, it allows theories to be tested impartially, thus promoting 'music theory ... as an experimental science';²⁰⁶ and third, 'added clarity can open the way to new concepts that were hidden by confusion surrounding the earlier ones.'²⁰⁷ Also, as Baroni has stated,

the transformation of ... rules into a computer code constitutes a check for their self consistency: when the rules are not precisely formulated, or some contradiction is

²⁰² Snell 1979, 18.

²⁰³ Snell 1979, 18.

²⁰⁴ Rothgeb 1966, 210.

²⁰⁵ Snell 1979, 4.

²⁰⁶ Haflich 1977, 12.

²⁰⁷ Snell 1979, 4.

present between different rules, the computer program cannot be written or it will not run properly.²⁰⁸

In other words, the most direct way to show that a theory is explicit and precise enough to merit the status of a ‘formal’ or ‘algorithmic’ theory is to implement it as a computer program that works.

8.3 *Lerdahl and Jackendoff complain that Peel and Slawson’s analyses are not allowed by their theory*

Lerdahl and Jackendoff claim that their ‘grammar is not even remotely weak enough to predict [Peel and Slawson’s] analysis’²⁰⁹ of Bach’s chorale ‘O Haupt voll Blut und Wunden’ (BWV 244/44) because ‘the reader cannot just consult the rule index and apply rules indiscriminately.’²¹⁰ But given Lerdahl and Jackendoff’s claim that their theory is a ‘generative’ one, I think Peel and Slawson can be forgiven for expecting the rules listed in the ‘Rule Index’²¹¹ of *GTTM* to be complete and explicit enough for the reader to be able to apply them algorithmically—that is, ‘without any exercise of intelligence’²¹²—to generate analyses from scores or performances. Lerdahl and Jackendoff’s objections to Peel and Slawson’s alternative analyses of Bach’s chorale BWV 244/44 and Mozart’s Sonata K.331 seem, in fact, to boil down to the accusation that Peel and Slawson did not interpret the rules in the way that they were intended. But if Peel and Slawson’s analyses are the results of a reasonable interpretation of the rules, then, in my view, Lerdahl and Jackendoff have no grounds for complaint because one is perfectly justified in expecting a formal or generative theory to be complete and explicit. Lerdahl and Jackendoff’s long, verbose discussions justifying their own particular application of the rules in specific instances could not possibly qualify as part of a formal or generative theory. These discussions may suggest ways in which the theory could be extended or completed but the theory itself must be assumed to consist only of all those unambiguous hypotheses that are made in *GTTM* and it is reasonable to assume that this set of hypotheses is a subset of the rules in the rule index.

8.4 *It may not even be possible in principle to describe algorithmically how Lerdahl and Jackendoff apply their theory in the sample analyses given in GTTM*

There are a number of rules in *GTTM* that are not defined sufficiently precisely for one to be able to apply them algorithmically: for example, the first, fourth, seventh and eighth Metric Preference Rules; the fourth Time Span Reduction Well-Formedness Rule; and the second, fourth and seventh Time Span Reduction Preference Rules. In particular, Lerdahl and Jackendoff fail to define explicitly how their preference rules interact to produce a ‘preferred’ analysis. For example, their ‘Intensification’ Grouping Preference Rule (GPR4) states that ‘where the effects picked out by GPRs 2 and 3 are relatively more pronounced, a larger-level group boundary may be placed.’²¹³ But because Lerdahl and Jackendoff at no point define an algorithm for deciding which of several given potential grouping boundaries predicted by GPRs 2 and 3 are ‘relatively

²⁰⁸ Baroni et al. 1992b, 599.

²⁰⁹ Lerdahl and Jackendoff 1985, 153–4.

²¹⁰ Lerdahl and Jackendoff 1985, 146.

²¹¹ Lerdahl and Jackendoff 1983, 345–352.

²¹² Borowski and Borwein 1989, 13.

²¹³ Lerdahl and Jackendoff 1983, 49 and 346.

more pronounced,' GPR4 cannot be applied algorithmically. Similarly, when discussing the effects of 'Fusion,' 'Transformation' and 'Cadential Retention' on their third Time-span Reduction Well-Formedness Rule, they admit that they do 'not specify what factors motivate the choice of fusion or transformation rather than ordinary reduction in a time-span' but 'nonetheless, [do] not hesitate to use fusion and transformation in [their] analyses where they [consider them to be] intuitively appropriate.'²¹⁴

Yet they insist that 'the rules must apply consistently'²¹⁵ and that 'the interaction of each rule with the other rules must likewise be consistent.'²¹⁶ Moreover, they claim that they themselves 'adhere to these constraints throughout *GTTM*,²¹⁷ 'avoiding ad hoc adjustments that make analyses work out the way [they] want.'²¹⁸

But because Lerdahl and Jackendoff do not formally define how their preference rules can interact, it is very difficult to guarantee that 'the interaction of each rule with the other rules ... [is] consistent.' Their decision not to formalize the principles of preference rule interaction leaves them free to apply the rules in an inconsistent and self-contradictory manner. Indeed, in my opinion, to be sure that they were applying the rules consistently they would have *had* to formalize the principles of rule interaction. Consequently it may not even be possible in principle to describe algorithmically how Lerdahl and Jackendoff apply their theory in the sample analyses given in *GTTM*, and therefore it may well be that these analyses are significantly 'better' than those that could be produced by any completely formalized explication of the theory.

8.5 *Lerdahl and Jackendoff's Preference Rules could be computationally implemented as heuristics of the type used in CHORAL*

Peel and Slawson claim that 'once preference rules are introduced, the theoretical apparatus becomes fatally flawed'²¹⁹ and that

the recent history of cognitive psychology suggests that progress in developing a satisfactory theory of musical processes will require a strong theoretical framework, not one that gives up the search for genuine rules of musical grammar at the outset.²²⁰

However, it is clear that Lerdahl and Jackendoff's preference rules fulfil an essentially identical function in *GTTM* as Ebcioğlu's heuristics do in CHORAL and that, in fact, it would probably be relatively straightforward to explicate and implement Lerdahl and Jackendoff's preference rules as heuristics of the type used in CHORAL. Indeed, the use of heuristics in CHORAL strongly suggests that Lerdahl and Jackendoff are incorrect in claiming that their theory 'could not be applied directly as a generator of pieces, because of the presence of preference rules.'²²¹ Also, the fact that Widmer (1995) has succeeded in implementing three of the four components of Lerdahl and Jackendoff's theory (in parallel with that of Narmour (1977)) in a complete working model of expressive performance, militates against Peel and Slawson's view that the

²¹⁴ Lerdahl and Jackendoff 1983, 159.

²¹⁵ Lerdahl and Jackendoff 1985, 154.

²¹⁶ Lerdahl and Jackendoff 1985, 146.

²¹⁷ Lerdahl and Jackendoff 1985, 146.

²¹⁸ Lerdahl and Jackendoff 1983, 55.

²¹⁹ Peel and Slawson 1984, 288.

²²⁰ Peel and Slawson 1984, 288.

²²¹ Lerdahl and Jackendoff 1983, 301.

strategy of using Preference Rules is ‘fatally flawed’ because they cannot be computationally implemented. In fact, Ebcioğlu asserts that it ‘is immediately clear’ that Lerdahl and Jackendoff’s theory could be translated into ‘a simple non-deterministic bottom-up or top-down parser.’²²² In any case, in practice, many of the most successful computational simulations of human musical tasks—for example, Ames’ Cybernetic Composer, Ebcioğlu’s CHORAL program, Maxwell’s (1988) tonal harmonic parser and Widmer’s (1995) computational model of expressive performance—use rules along the lines of heuristics or preference rules. And as Ebcioğlu has stressed on many occasions, systems that rely on absolute rules alone almost invariably perform less well than those that employ heuristics.

As mentioned above, because Lerdahl and Jackendoff do not define an algorithm for deciding which of several given potential grouping boundaries predicted by GPRs 2 and 3 are ‘relatively more pronounced,’²²³ GPR4 cannot be directly implemented as an algorithmic procedure. They themselves admit that

in order to make the theory fully predictive, it might be desirable to assign each rule a numerical degree of strength, and to assign various situations a degree of strength as evidence for particular rules. Then in each situation the influence of a particular rule would be numerically measured as the product of the rule’s intrinsic strength and the strength of evidence for the rule at that point; the most ‘natural’ judgement would be the analysis with the highest total numerical value from all rule applications.²²⁴

For example, in order to implement GPR4 and resolve conflicts between competing potential grouping boundaries defined by GPR3a-d, one would first need to define ways of measuring changes in each of the four parameters (register, dynamics, articulation and length) so that one could compare the relative intensity of changes in these parameters. For example, one would need to be able to specify what size of registral change (3a) is equivalent to, say, a change from slurred articulation to staccato (3c). It might be possible to base such metrics on the results of experiments such as those carried out by Deliège (1987), in which subjects are required to ‘chunk’ melodic stimuli into groups. For example, the results of Deliège’s experiments showed that, for experienced musicians, change in timbre was the strongest cue to hearing a group boundary and that the relative strengths of changes in the other domains were, in order of decreasing strength, as follows:

Timbre > Dynamics > Slur/rest > Register ≥ Articulation ≥ Attack point > Length > Contour

²²² Ebcioğlu 1987b, 117.

²²³ Lerdahl and Jackendoff 1983, 49 and 346.

²²⁴ Lerdahl and Jackendoff 1983, 47.

9 Grammaticalness and acceptability

9.1 *There is no musical analogue of 'everyday speech'—'correct' but 'unmusical' music does not exist*

Steedman claims that the 'fact' that 'works of art break rules by definition' raises problems for the analogy between his approach and that of linguistics:

The fact that we are dealing with the *artistic* use of musical language raises problems for the analogy with linguistics...It means that we are dealing with a corpus that is more like poetry than the mere well-formed strings of a language.²²⁵

I do not think it is fruitful to assume that it is impossible to develop theories for works of art that comprehensively and rigorously account for the sets of artistic phenomena that they are designed to be able to account for. I do agree that a piece of music is more analogous to a piece of creative, artistic writing such as a poem or a story than it is to a single utterance or sentence in a verbal language whose function is only to communicate and not to use language artistically. However, Steedman seems to be implying that his attempt to create a generative grammar for jazz chord sequences using a corpus of nine 'paradigmatic' 12-bars taken from Coker's *Improvising Jazz* (Coker 1964), is analogous to trying to create, for example, a generative grammar for English using a corpus of poetry. But these two tasks are only analogous if one believes, like Steedman, first, that a grammar for 12-bar chord sequences should generate the set of 'possible 12-bars—good, bad, and indifferent—rather than the set that musicians actually play';²²⁶ and, second, that there is a difference between the set of 'possible' 12-bars and the set that musicians might feasibly want to play. Thus Steedman believes his grammar can legitimately

allow sequences that are too complex or bizarre for anyone to want to play or be able to understand, just as a formal grammar of English will allow sentences that are impossibly complex or whose meaning is absurd.²²⁷

However, Steedman seems to be ignoring the fact that in a verbal language such as English there are lots of examples of correct utterances or sentences that are not poetry, whereas in music, there are no examples of 'musical utterances' that are not analogous to poetry—that is, all musical utterances are (at least intended to be) successful musical artworks. There is no evidence for the reality of a musical 'language' that includes lots of 'musical' structures that are not at least intended to be the musical equivalent of poetry. Therefore, in my opinion, it would be more logical to assume that musical styles only contain structures that are the musical equivalent of poetry and do not contain any structures that are the putative musical equivalent of non-poetic linguistic utterances. In order to be a 'possible 12-bar,' a chord sequence would, in my view, have to be able to function as the harmonic framework of a musical artwork. In my view, the notion of a 'possible' but 'unmusical' or unacceptable piece of music is a nonsense. To what extent could a 12-bar chord sequence be considered 'possible' if it was not one that a musician might feasibly want to play? Clearly, whereas a criterion of acceptability for utterances in a spoken language could legitimately allow sentences that are meaningful but not 'artistic,' a criterion of acceptability for a piece of music would,

²²⁵ Steedman 1984, 55, footnote 2.

²²⁶ Steedman 1984, 55, footnote 2.

²²⁷ Steedman 1984, 55, footnote 2.

in my view, necessarily disallow any ‘musical’ utterances that were ‘unaesthetic’ or ‘unmusical.’

Therefore, Lerdahl and Jackendoff are correct in asserting that ‘whereas music characteristically functions as art, language does not’ and that therefore ‘the data for linguistic study are the sentences of the everyday world, for which there is no musical counterpart’ so that ‘poetry or drama would seem to provide a closer analogy to music.’ However, they go on to claim that just as ‘one must first understand the French language’ if one is ‘to appreciate the poetic or dramatic structure of a poem in French,’ so ‘to appreciate a Beethoven quartet as art, one must understand the idiom of tonal music.’²²⁸

But it seems to me that Lerdahl and Jackendoff are contradicting themselves here. First they say that there is ‘no musical counterpart’ to ‘the sentences of the everyday world’ but then they imply that ‘the idiom of tonal music’ is analogous to ‘the French language’ because the relationship between a Beethoven string quartet and ‘the tonal idiom’ is essentially parallel to that between a French poem and the French language. But if there is ‘no musical counterpart to the sentences of the everyday world’ then the ‘tonal idiom’ cannot contain any structures that correspond to such ‘everyday sentences’ and therefore cannot be considered analogous to a natural language such as French which consists almost entirely of ‘everyday sentences.’ In order ‘to appreciate a Beethoven quartet as art’ one must first be familiar with other similar music—that is, other string quartets, other classical music, other music by Beethoven and so on. But in suggesting that there is a direct parallel between French and the tonal idiom, Lerdahl and Jackendoff seem to be implying that in order to appreciate a Beethoven quartet as art, one must first understand the ‘everyday (musical) sentences’ of the ‘tonal idiom.’ In other words, they are suggesting that ‘the idiom of tonal music’ contains structures that are the musical equivalent of ‘sentences of the everyday world.’

Snell similarly seems to have doubts as to the validity or utility of the notion that there exists some musical analogue of ‘the sentences of the everyday world:’

Whatever analogue there is in music to ‘everyday speech,’ it does not offer much challenge to our musical understanding. On the other hand, it is an exciting prospect that we may gain some precise understanding of what goes on in a musical artwork, and why.²²⁹

9.2 *Distinction between grammaticalness and acceptability*

Chomsky uses ‘the term “acceptable” to refer to utterances that are perfectly natural and immediately comprehensible without paper-and-pencil analysis, and in no way bizarre or outlandish.’²³⁰ Thus, ‘the more acceptable sentences are those that are ... more easily understood, less clumsy, and in some sense more natural.’²³¹

If one allows unlimited recursion or unlimited nesting in a generative grammar then the artificial language generated by the grammar will be infinite and will contain sentences that are too complex or too long for anyone to be able to understand. One could be forgiven at this point for suggesting that a sentence that is too complex or too

²²⁸ Lerdahl and Jackendoff 1983, 7.

²²⁹ Snell 1979, 20.

²³⁰ Chomsky 1965, 10.

²³¹ Chomsky 1965, 11.

long to be understood surely could not be considered grammatical and that therefore if a grammar generates such sentences then it must overgenerate.

However, Chomsky stresses that

the notion 'acceptable' is not to be confused with 'grammatical.' Acceptability is a concept that belongs to the study of performance, whereas grammaticalness belongs to the study of competence ... Like acceptability, grammaticalness is, no doubt, a matter of degree ... but the scales of grammaticalness and acceptability do not coincide. Grammaticalness is only one of many factors that interact to determine acceptability.²³²

Therefore a generative grammar can perfectly legitimately generate unacceptable sentences provided that they are grammatical. According to Chomsky, whether or not a sentence can be understood, or whether or not it could be used in practice merely determines whether or not it is acceptable and is thus a question that should be addressed only by someone who is interested in the study of performance. Someone who is only interested in competence need not concern themselves with whether or not the sentences generated by a grammar are acceptable. Students of linguistic competence only need to worry about whether or not sentences are grammatical. Thus, a sentence that contained, say, two hundred nested clauses or three million words would definitely be unacceptable, but it would not necessarily be ungrammatical.

According to Chomsky,

linguistic theory is concerned primarily with an ideal speaker-listener, in a completely homogeneous speech-community, who knows its language perfectly and is unaffected by such grammatically irrelevant conditions as memory limitations, distractions, shifts of attention and interest, and errors (random or characteristic) in applying his knowledge of the language in actual performance.²³³

That is, Chomsky believes that a successful theory of linguistic performance must consist of a theory of linguistic competence in the form of a grammar that generates the set of grammatical (but not necessarily acceptable) sentences in a language, supplemented by a system of 'performance filters' that when given grammatical sentences as input generates acceptable sentences as output. This system of performance filters therefore effectively simulates the 'memory limitations, distractions, shifts of attention and interest, and errors' that, in Chomsky's opinion, prevent *real* native speakers from understanding and using certain 'bizarre and outlandish' grammatical sentences. Figure 9-1 shows the logical relationship between a theory of linguistic competence and a theory of linguistic performance and I hope makes clear the distinction and relationship between Chomsky's notions of grammaticalness and acceptability. If a theory of performance has the form shown in Figure 9-1 then I shall say that it is constructed on the 'Competence→Performance Filters' model or that it is a 'Competence→Performance Filters'-type performance theory.

Unfortunately, as Chomsky himself admits,

although one might propose various operational tests for acceptability, it is unlikely that a necessary and sufficient operational criterion might be invented for the much

²³² Chomsky 1965, 11.

²³³ Chomsky 1965, 3.

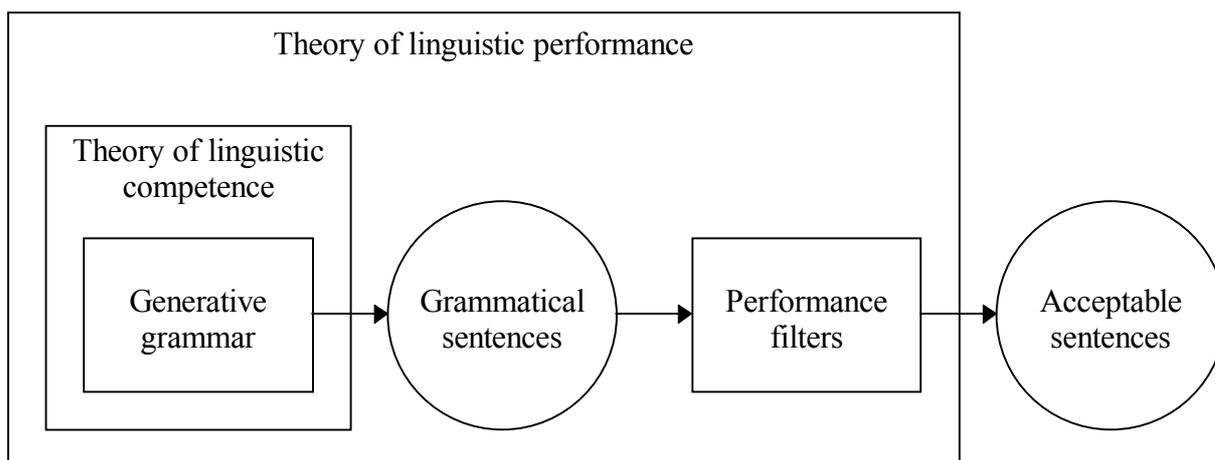


Figure 9-1

more abstract and [in Chomsky's opinion] far more important notion of grammaticalness.²³⁴

Therefore, because it would be possible to define a satisfactory, empirical explication of the concept of acceptability that would allow one to decide categorically for any given sentence whether or not it was an acceptable sentence in some natural language, it would certainly be logically possible in principle to produce an empirically refutable theory of linguistic performance for that natural language that hypothesized that some generatively defined set of sentences was equal to the set of all and only acceptable sentences in the language.

However, because it would never be possible to define a satisfactory empirical explication of the concept of grammaticalness, it would never be possible to decide categorically for any sentence whether or not it was a grammatical sentence in some natural language. Therefore a theory of linguistic competence that hypothesizes that some generatively defined set of sentences is equal to the set of all and only grammatical sentences in some natural language can never be empirically tested. Such a theory of linguistic competence can become partially testable, however, when it forms part of a theory of linguistic performance. Indeed, in my view, the value of a theory of competence resides entirely in whether or not it can be employed in a successful theory of performance.

9.3 *The set of all and only acceptable structures in a natural language or musical style is finite*

According to Ebcioğlu, Hofstadter claims that it would never be possible even in principle to define an algorithm capable of generating all and only those pieces in the style of the music in some specified genre by some specified composer:

Hofstadter [(1979, 1982)] perhaps overly impressed by an older topic in recursive function theory, believes that works of art must be a productive set, i.e. given any algorithm, a work of art that is not generated by this algorithm can be found, or the algorithm can be shown to generate a non-work-of-art.²³⁵

²³⁴ Chomsky 1965, 11.

²³⁵ Ebcioğlu 1987b, 87, footnote 46.

In other words, Hofstadter believes that any set that contains all and only those pieces in the style of the music in a specified genre by a specified composer would necessarily be a productive set. Now productive sets are infinite sets, therefore if one proved that musical styles were actually finite, then one would also have disproved Hofstadter's claim that such sets are productive. In my opinion, Ebcioglu succeeds in doing this in the following passage:

For the case of music, we feel that the set of all 'pieces' that can be encoded via digital recordings of some fixed sampling rate, and that take less than a reasonable time limit is a satisfactory superset of the set of interesting music. The finiteness of this otherwise huge set does not of course make the discovery of a practical algorithmic description of music less difficult, it merely points out that productiveness is an incorrect model of the true difficulty.²³⁶

Therefore, as Ebcioglu points out, 'in all cases of practical interest, the set of pieces in the desired style ... is finite.'²³⁷

To make this a little more concrete, imagine that one is attempting to develop an algorithmic style theory T for the style of Bach's chorale harmonizations. One first defines the corpus kernel $\underline{s}_k(T)$ to contain all and only those scores in any copy of, say, Klaus Schubert's edition of the *371 Four-Part Chorales* (Bach 1990). The corpus of the style theory $\underline{s}_c(T)$ would then, by definition, contain all and only those scores notationally equivalent to the scores in the Schubert edition. The style of the theory $\underline{s}_s(T)$ would consequently contain all and only those scores representationally equivalent to scores that are either members of the corpus or determined to be in the style of the corpus by an appropriate acceptability algorithm. Let us now define \underline{d}_u to be the set of all and only possible 32-bit digital recordings at a sample rate of 48kHz, less than 1 year in duration. \underline{d}_u is finite—it contains $2^{32 \times 48000 \times 3600 \times 24 \times 365} \approx 2^{5 \times 10^{13}}$ members. It seems reasonable to assume that any pair of performances whose 32-bit 48kHz digital recordings were identical would actually sound indistinguishable to the human ear. For every score in $\underline{s}_s(T)$ there is, in general, more than one possible performance. Let us assume that for each member of the set of all and only possible correct performances of any given score in $\underline{s}_s(T)$ there exists exactly one digital recording in \underline{d}_u . This implies that for each and every member of $\underline{s}_s(T)$ there exists more than one digital recording of a correct performance in \underline{d}_u which in turn implies that $\underline{s}_s(T)$ contains fewer members than \underline{d}_u and is therefore finite.

Therefore musical styles are finite and consequently any algorithmic style theory whose universal set of well-formed scores is an infinite set definitely overgenerates and is therefore definitely incorrect. This implies that any algorithmic style theory whose composing algorithm employs unlimited recursion or unlimited nesting is definitely incorrect.

Chomsky claims that 'a generative grammar must be a system of rules that can iterate to generate an indefinitely large number of structures.'²³⁸ But the above proof applies equally to spoken, verbal language—that is, the set of all and only acceptable sentences in any natural, spoken language is definitely finite. Therefore, although a generative grammar intended to characterize an 'ideal speaker-listener's competence'

²³⁶ Ebcioglu 1987b, 87, footnote 46.

²³⁷ Ebcioglu 1987b, 87.

²³⁸ Chomsky 1965, 16.

can legitimately generate an infinite set of grammatical structures including many that are too complex or long to be understood in a finite time, the set of sentences or pieces of music generated by a *performance* theory for a musical style or natural language must be finite since there are certainly upper limits on how long and complex a piece of music or sentence can be before it becomes unacceptable. For example, a tonal piece of music that lasted one million years would not be acceptable. Nor would one that used one million notes per second. Therefore the ‘performance filters’ in a performance theory constructed on the ‘Competence→Performance Filters’ model described above must limit the (possibly infinite) universal output set of the grammar component of the theory to a finite set of acceptable structures. Clearly this finite set would, in the case of a natural language or the musical style of a composer, be very large indeed—certainly far too large for it to be possible in practice to prove that a performance theory for a musical style or a natural language did not overgenerate.

9.4 *The ‘Competence→Performance Filters’ model is plausible in a theory of improvisation but is not plausible in a theory that aims to characterize a non-improvisatory musical style*

The hypothesis that it is a good strategy to attempt to model linguistic performance by first finding a theory of competence (i.e. a generative grammar) and then supplementing this grammar with ‘performance filters’ as described above, is plausible in a theory for a spoken language where it is certainly true that speakers suffer from ‘memory limitations, distractions, shifts of attention, and errors’ that set limits on the complexity and length of sentences that can be understood. It is also plausible in a theory of musical improvisation which, like spoken language, is clearly also limited by the ‘real-time’ processing limitations of the human brain.

As mentioned above, Steedman admits that

on occasion the rules [of his grammar] will allow sequences that are too complex or bizarre for anyone to want to play or be able to understand, just as a formal grammar of English will allow sentences that are impossibly complex or whose meaning is absurd.²³⁹

Now, although a sequence that is ‘too complex or bizarre for anyone to want to play or be able to understand’ may be grammatical it would certainly not be acceptable. Therefore, if Steedman’s goal was to produce a device that was not in itself a performance theory aiming to characterize the set of acceptable 12-bars, but that was rather intended to be a device for generating a set of grammatical 12-bars that could then form the input to a system of performance filters that would generate acceptable chord sequences as output, then one would have understood his apparent lack of concern with the fact that his rules allowed ‘sequences that are too complex or bizarre for anyone to want to play or be able to understand.’

But in general, 12-bar chord sequences are not improvised. Typically, such sequences are known beforehand and are derived from popular songs that are composed, written down and sometimes even published before they are publicly performed for the first time.²⁴⁰ This means that issues of memory limitation, distraction and shifts of attention do not play a part in the construction of jazz chord sequences

²³⁹ Steedman 1984, 55, footnote 2.

²⁴⁰ Johnson-Laird 1991, 306.

because the composer can use the paper on which he or she is writing the music to make up for any lapses in attention, distraction or memory limitations from which he might suffer. The ‘Competence→Performance Filters’ model is therefore far less plausible in a theory that aims to characterize a non-improvisatory musical style than it is in a theory of improvisation or a theory of spoken language. Rather it would seem to be more justified in the case of written musical styles to aim to characterize directly the set of acceptable pieces in the style—that is, the class of all and only those pieces that could function as acceptable musical artworks in the style.

As demonstrated in the previous section, although the set of grammatical blues chord sequences may be infinite, the set of acceptable blues chord sequences is certainly finite. Also, as mentioned above, any grammar or algorithm that employs unlimited recursion or unlimited nesting generates an infinite set and therefore cannot be a correct characterization of the set of all and only acceptable sentences or pieces of music in some natural language or musical style. Therefore, the fact that Steedman chose to employ recursion in his grammar because ‘the set of blues chord sequences ... seems in principle infinite’²⁴¹ is further evidence that he was intending to characterize the class of grammatical 12-bars rather than the class of acceptable 12-bars. But, in my opinion, Steedman was wrong to attempt to generate the set of grammatical chord sequences because chord sequences are not improvised and therefore a theory along the lines of the ‘Competence→Performance Filters’ model would be implausible.

²⁴¹ Steedman 1984, 55.

10 Untestability of incomplete theories

10.1 *Impossibility of judging grammaticalness of sequences of syntactic categories*

As Moore and Carling point out, if a human were capable of judging whether or not a sentence was grammatical then he would

be able to judge the grammaticalness of, for example, the string:
D N prep prep D N cop adv adj

but ‘to do this he would have to attach words to the categories.’²⁴²

This shows that humans are, in fact, incapable of judging whether or not sentences are grammatical. They can judge only whether or not sentences are acceptable. Even if one thinks that one is making a judgement of grammaticalness, one will always actually be making a judgement of acceptability. For example, Chomsky claimed that the sentence, ‘Furiously sleep ideas green colorless’ is non-grammatical and therefore that the sequence of categories ‘Adv V N Adj Adj’ is also non-grammatical. But, as Moore and Carling point out, ‘Always dye shirts greenish blue’ is certainly acceptable and can be represented by the same sequence of syntactic categories. Similarly, Chomsky claimed that the sequence ‘Adv N N V Adj’ is non-grammatical but, as Moore and Carling again point out, the definitely acceptable sentence ‘Inevitably newspaper people appear tactless’ has this very structure. Moore and Carling fail to acknowledge, however, that, in fact, the adjective ‘greenish’ in the first example is functioning adverbially and the noun ‘newspaper’ in the second example is functioning adjectivally. Nonetheless, I think their examples do make clear that,

if judgements of grammaticalness cannot be made on strings of syntactic categories without assigning words to those categories, then it seems we must recognise that the judgements are not judgements of grammaticalness but judgements on particular sequences of words in a structure.²⁴³

I think these examples also illustrate that human subjects are incapable of judging the acceptability of objects that are not immediately recognizable as examples of a class of phenomena with which the subjects are familiar. Also, these examples highlight the fact that the perceived acceptability of an object depends to a great extent on how well it fulfils the function that a subject automatically assumes the object was intended to perform. A subject automatically assumes that a sequence of words in his or her native language is intended to be a meaningful sentence and is therefore able to make a judgement as to whether or not the sequence of words is an acceptable sentence in his or her language. But a sequence of syntactic categories performs no well-defined function in itself, therefore subjects will be unable to judge its acceptability.

10.2 *Kassler’s theory is not testable because it is not complete*

Kassler claims that ‘preliminary testing’ of his computational explication of Schenker’s middleground theory ‘indicates that the explication accounts satisfactorily for the musical structures it is intended to explain.’²⁴⁴ Kassler’s theory is effectively the

²⁴² Moore and Carling 1982, 77.

²⁴³ Moore and Carling 1982, 81.

²⁴⁴ Kassler 1975, 7.

hypothesis that the set of assertions of a specified formalized language is equal to the universal set of ‘terminal’ middleground structures that can be transformed by Schenker’s foreground theory (or a suitable explication of this theory) into acceptable pieces of tonal music. Clearly, the process of testing this hypothesis would consist of

1. testing for undergeneration by trying to find a tonal piece for which no terminal middleground structure can be found that is an assertion of the theory’s formalized language; and
2. testing for overgeneration by trying to find an assertion in the theory’s formalized language that is not a terminal middleground structure for a tonal piece.

To show that a piece could not be generated from any terminal middleground structure in the theory’s formalized language, one would need to be able to show that the tonal piece could not be derived by Schenker’s foreground rules from any assertion of the theory’s formalized language. But Schenker’s foreground rules are not a formal system. Therefore in order to be able to do this one would first have to explicate Schenker’s foreground rules and express this explication as an automatic parsing algorithm along the lines of Kassler’s explication of the middleground rules.

However, trying to find an assertion in the theory’s formalized language that is not a terminal middleground structure for a tonal piece is clearly doomed from the start. For any given ‘terminal middleground structure’ there will be a vast number of possible foreground structures that could be derived from it by any reasonable explication of Schenker’s foreground rules. But to prove that no tonal piece could be derived from some terminal middleground structure, one would have to prove by means of an acceptability algorithm that *every* foreground structure derivable from the given middleground structure was not a tonal piece. It would therefore clearly be impossible in practice to show that a given terminal middleground structure did not generate any foreground structures that were tonal pieces even if one had developed an explication of Schenker’s foreground theory in the form of an automatic parsing algorithm. Therefore it would be impossible to show that any explication of Schenker’s middleground theory overgenerated.

Kassler’s theory as it stands is therefore not empirically refutable because:

1. he does not define an acceptability algorithm and a corpus and thus does not adequately define the set of pieces that his theory is intended to be able to generate; and
2. he does not provide an explication of Schenker’s foreground theory in the form of an automatic parsing algorithm along the lines of his explication of Schenker’s middleground theory.

Kassler therefore had no way of determining whether or not his ‘explication account[ed] satisfactorily for the musical structures it [was] intended to explain’²⁴⁵ and was thus not justified in claiming that it did.

10.3 The incompleteness of Steedman’s theory and the impossibility of judging the acceptability of a chord sequence

Cross observes that Steedman’s statement that he is attempting

²⁴⁵ Kassler 1975, 7.

to characterize the rules that underlie the comprehension of *one aspect* [Cross's emphasis] of musical form in the musical culture of western tonal harmony in much the same way that a linguistic study might try to characterise the rules underlying the comprehension of a language like English,²⁴⁶

'hints at the incompleteness of [Steedman's] theory.' Cross goes on to note that

the fact that 'the comprehension of [only] one aspect of musical form' is the focus of his theory necessarily limits the applicability of his theory, in that the multi-dimensional nature of 'real' musical utterances in a musical culture (i.e., the fact that real musical utterances embody, at the least, melodic and rhythmic/metrical structure) might require the implementation of rules that have consequences for harmonic structure yet are not expressible in terms of rules that bear solely on, and employ as their basic terms, harmonic configurations and relations.²⁴⁷

In much the same way that the acceptability of a string of syntactic categories cannot be judged 'without assigning words to those categories,'²⁴⁸ so the acceptability of a chord sequence depends entirely upon whether or not it can be used to construct a complete and acceptable piece of music. It is therefore not strictly meaningful to speak of the 'acceptability' of a chord sequence, or, indeed, of any abstract structural reduction of a piece of music. It is only meaningful to speak of the acceptability of structures that are intended to be complete musical artworks. It is therefore very difficult to judge the performance of a computer program that only models part of a human creative skill.

10.4 A theory of rhythmic structure would only be testable as part of a complete style theory

Lerdahl and Jackendoff claim that they did not attempt to achieve a computationally implementable theory because 'various aspects of [their] theory [could not] be so formalized on the basis of present understanding of the issues.'²⁴⁹ It is certainly true, as Penrose has demonstrated,²⁵⁰ that there are some natural processes including certain mental tasks routinely performed by humans that could not be algorithmically described even in principle.

But the fact that one does not know how something works does not imply that one should not be trying to guess how it might work on the basis of what it actually does. In my opinion, the best way to reveal lacunae in 'present understanding' of how humans perform certain mental tasks that result in observable output, is to attempt to develop complete, working models that successfully simulate the behaviour exhibited and the output produced by humans when performing those tasks.

Of course, just because a model is complete and formal enough to be computationally implemented does not imply that it will not be ad hoc. In fact, an ad hoc theory consisting of a large number of independent and 'special case' rules can sometimes form the basis of an extremely effective and convincing computer simulation of a human skill. Ebcioğlu's CHORAL program, for example, employs 350 independent rules.²⁵¹ However, the developer of such a computer simulation will be able to tell from

²⁴⁶ Steedman 1984, 52.

²⁴⁷ Cross 1996b.

²⁴⁸ Moore and Carling 1982, 81.

²⁴⁹ Lerdahl and Jackendoff 1983, 333, note 6.

²⁵⁰ Penrose 1994, 28.

²⁵¹ Ebcioğlu 1987b, abstract.

the relative ‘ad hocness’ of different parts of the program which parts of his or her theory are most in need of generalization.

Lerdahl and Jackendoff claim that a top-down, synthetic ‘axioms → rule system → musical surface’ system such as the one proposed by Peel and Slawson ‘in their chord-grammar fragment’²⁵² would be unable to provide answers to the following two questions:

1. ‘Why does the musical surface take the specific rhythmic form that it does?’
2. ‘How is the rhythmic structure to be characterized?’²⁵³

Or, to be more precise,

1. What are the necessary and sufficient constraints on rhythmic structure that must be satisfied by an acceptable piece of music in some defined style?
2. Why are these particular constraints necessary and sufficient?

In my opinion, the logical way to go about attempting to answer the first of these two questions would be as follows:

1. Define a style by specifying a corpus and an acceptability algorithm that allows one to be able to determine for any complete score of a piece of music whether or not the piece is in the style of the corpus.
2. Define a representation algorithm that generates for any given score of the type in the corpus a representation of its rhythmic structure.
3. Generate representations for a number of scores in the corpus (or, if possible, all known members of the corpus).
4. Make a system of hypothetical constraints on rhythmic structure that can account for all of the corpus representations generated in step 3.
5. Implement this system of constraints as a parsing algorithm to prove that it can account for all the corpus representations generated in step 3 and so that it is possible to determine automatically and certainly for any other piece whether or not its rhythmic structure can be accounted for by the constraints.
6. Implement the system of constraints in a composing algorithm that automatically generates representations which, when given to a score algorithm as input, generate complete pieces of music in the form of scores of the type of the pieces in the corpus.
7. Use the composing algorithm and the acceptability algorithm to test for overgeneration. As long as the pieces generated are in the style, one can claim to be successfully generating a set that might be equivalent to the universal set of acceptable rhythmic structures in the style of the rhythmic structures of the pieces in the corpus. As soon as one generates a piece that is not in the style, then one knows that somewhere in the composing algorithm, something is wrong. This might be due to an error in the rhythmic structure theory incorporated in the algorithm or it might be due to a fault elsewhere in the algorithm. Unfortunately, one cannot test the rhythmic structure theory in isolation because human subjects would be incapable of meaningfully assessing whether or not an abstract

²⁵² Peel and Slawson 1984, 288–299.

²⁵³ Lerdahl and Jackendoff 1985, 149.

‘rhythmic structure’ was ‘acceptable.’ A theory of rhythmic structure can therefore only achieve some degree of testability if it is incorporated in a complete theory for a musical style. The acceptability of a rhythmic structure depends entirely upon whether or not it could be used in a complete, meaningful musical artwork.

Having found a set of necessary and sufficient constraints that after extensive testing for overgeneration and undergeneration remain capable of accounting for the rhythmic structures of all and only pieces in the style under consideration, one can then attempt to determine why these particular conditions seem to be necessary and sufficient for accounting for the rhythmic structures of pieces in this particular style. That is, one can then attempt to derive these constraints from more general principles of cognitive organization or, to put it another way, one can attempt to show that these constraints can be considered logical consequences of some other, higher level principles of cognitive organization. That is, one can try to show that the fact that we are human implies that these constraints are necessary and sufficient.

Lerdahl and Jackendoff suggest that someone who advocates a top-down, synthetic theory along the lines of an ‘axioms → rule system → musical surface’ system would have trouble answering the two questions that I have attempted to answer above. As should be clear from the strategy that I have suggested, a necessary part of attempting to refute the hypothesis that a particular set of constraints generates the universal set of rhythmic structures in a particular style is to attempt to refute that the constraints generate only those rhythmic structures in the style. In my view, the only way of doing this is to incorporate the constraints into a theory along the lines of an ‘axioms → rule system → musical surface’ system that aims to model the complete composition process that leads to the production of pieces in the style being modelled.

11 Competence and performance

11.1 *Chomsky, Dennett and Lerdahl and Jackendoff believe that whether or not theories correctly describe mental intuitions is more important than whether or not they account for the behaviour that putatively results from these intuitions*

Chomsky, Dennett and Lerdahl and Jackendoff believe that whether or not theories correctly describe mental intuitions is more important than whether or not they account for the behaviour that putatively results from these intuitions. For example, Chomsky claims that anyone who believes that linguists should attempt to be more objective should ‘justify his belief ... by showing how ... [increased objectivity could] lead to new and deeper understanding of linguistic structure.’²⁵⁴ According to Chomsky, ‘there is no way to avoid the traditional assumption that the speaker-hearer’s linguistic *intuition* [my emphasis] is the ultimate standard that determines the accuracy of any proposed grammar [or] linguistic theory.’²⁵⁵ In other words, Chomsky’s view is that the success of a grammar or linguistic theory depends entirely upon how well it describes a native speaker-listener’s *intuitions*—the extent to which it correctly characterizes the speaker-listener’s linguistic *behaviour* is, in Chomsky’s view, not important. He fails to acknowledge that if a competence theory cannot account for the behaviour and observable products of the mental processes that it purports to describe, then it *must* be incorrect, no matter how elegant or apparently ‘insightful’ it is.

In my view, ‘the speaker-hearer’s linguistic intuition’ cannot possibly be considered ‘the ultimate standard that determines the accuracy of ... [a] linguistic theory’ because the speaker-hearer’s intuitions are not accessible to empirical observation and no empirically testable predictions can be *logically* deduced from hypotheses about the nature of a speaker-hearer’s intuitions. Chomsky claims that work in generative linguistics must ‘converge on the tacit knowledge of the native speaker.’²⁵⁶ But how is one supposed to know whether or not one’s theory is ‘converging on the tacit knowledge of the native speaker’? In my view, the study of language should be converging on the empirically verifiable facts of linguistic behaviour.

Dennett observes that

it has been said of behaviorists that they feign anesthesia—they pretend they don’t have the experiences we know darn well they share with us.²⁵⁷

Just as Chomsky claims that the onus falls on anyone who advocates increased objectivity in linguistics to show how this could ‘lead to new and deeper understanding of linguistic structure,’²⁵⁸ so Dennett claims that ‘if I wish to deny the existence of some controversial feature of consciousness, the burden falls on me to *show* that it is somehow illusory.’²⁵⁹

²⁵⁴ Chomsky 1965, 21.

²⁵⁵ Chomsky 1965, 21.

²⁵⁶ Chomsky 1965, 20.

²⁵⁷ Dennett 1991, 40.

²⁵⁸ Chomsky 1965, 21.

²⁵⁹ Dennett 1991, 40.

Although I certainly do not subscribe to many of the ‘mentalist’ views espoused by Dennett and Chomsky, I certainly do not ‘feign anesthesia.’ I merely claim that one can never be sure what another individual is experiencing and that it is often very difficult to describe even what one is experiencing oneself. I do not deny that I experience and that other people experience. I simply claim that detailed predictions about the nature of human experiences cannot be tested because no reliable way exists for determining the nature of these experiences directly and no empirically testable predictions can be *logically* deduced from hypotheses about human experiences.

Just as Chomsky and Dennett claim that linguists and psychologists should be more concerned with characterizing intuitions than the observable behaviour that putatively results from them, so Lerdahl and Jackendoff subscribe to the view that ‘a comprehensive theory of music would account for the totality of the listener’s musical intuitions’²⁶⁰ and claim that anyone who denies ‘the existence of as rich a musical grammar as [they] have claimed’ should be required to provide an alternative ‘explanation of all the musical intuitions [they discuss] in motivating [their] theory.’²⁶¹ However, in my view, a music theory should rather aim to account for the totality of the listener’s and composer’s musical *behaviour* and the observable, tangible products of that behaviour. I do not deny, however, that speculation about a listener’s intuitions may help one to achieve this aim.

Lerdahl and Jackendoff are certainly correct in pointing out that since the early 1960s,

it has been an unquestioned assumption of actual research in linguistics, that what is really of interest in a generative grammar is the structure it assigns to sentences, not which strings of words are or are not grammatical sentences.²⁶²

But they attempt to use this fact as justification for their decision to develop a theory that ‘is not intended to enumerate what pieces are possible but to specify a *structural description* for any tonal piece; that is, the structure that the experienced listener infers in his hearing of the piece.’²⁶³ However, even Chomsky never *entirely* abandoned the idea that a grammar should ‘enumerate what sentences are possible within a language.’ During the early 1960s, Chomsky began to believe that a generative grammar should be required to do much more than just weakly generate some natural language, but he never claimed that a grammar should *not* be required to weakly generate the natural language that it is intended to characterize. Thus, whereas in *Syntactic Structures*²⁶⁴ Chomsky presents the requirement of weak generation as being a *sufficient* condition on the adequacy of a generative grammar, in *Aspects of the Theory of Syntax*²⁶⁵ it is presented as being only a *necessary* condition on the adequacy of a grammar.

As explained in section 7.1 above, the fact that Lerdahl and Jackendoff deny that ‘a musical grammar should be [implemented as] an algorithm that composes pieces of music,’²⁶⁶ seems to imply that they see no point in testing such a grammar for overgeneration and therefore that they see no point in attempting to accurately

²⁶⁰ Lerdahl and Jackendoff 1983, 8.

²⁶¹ Lerdahl and Jackendoff 1983, 282.

²⁶² Lerdahl and Jackendoff 1983, 6.

²⁶³ Lerdahl and Jackendoff 1983, 6.

²⁶⁴ Chomsky 1957.

²⁶⁵ Chomsky 1965.

²⁶⁶ Lerdahl and Jackendoff 1983, 6.

characterize musical styles. To some extent they are correct in asserting that their approach is

consistent with the methodology of generative linguistics, for, despite the term ‘generative,’ the goal in linguistic theory is to find the rules that assign correct structures to sentences. Consequently the sentences as such in linguistic theorizing are usually taken as given.²⁶⁷

In other words, like Chomsky and Dennett, Lerdahl and Jackendoff are far more interested in speculating about human mental processes than in attempting to accurately characterize human behaviour and the tangible products of that behaviour.

11.2 *It is impossible to determine whether or not a structural description correctly describes how a speaker-hearer understands a sentence*

Chomsky defines a grammar to be ‘*descriptively adequate* to the extent that it correctly describes the intrinsic competence of the idealized native speaker.’²⁶⁸ He states that ‘a grammar is descriptively adequate if it strongly generates the correct set of structural descriptions’²⁶⁹ and therefore that ‘a fully adequate grammar must assign to each of an infinite range of sentences a structural description indicating how this sentence is understood by the ideal speaker-hearer.’²⁷⁰

But how is one to determine whether or not the structural descriptions assigned by a grammar to sentences in the native speaker’s language *correctly* describe how the native speaker understands these sentences? In order to determine for any given structural description whether or not it is a *correct* description of how a speaker understands a sentence, one must be able to do at least one of two things: one must either be able to *observe directly* how the speaker understands the sentence, or one must be able to logically deduce that the speaker will exhibit some specific empirically observable behaviour if and only if the structural description is correct.

Now it is clear that the only individual who *may* be able to directly observe how a speaker understands a sentence is the speaker himself. But Chomsky denies that a speaker’s claims about his own linguistic intuitions are to be trusted:

It is quite apparent that a speaker’s reports and viewpoints about his behavior and his competence may be in error.²⁷¹

Thus, while claiming that ‘every speaker of a language has mastered and internalized a generative grammar that expresses his knowledge of his language,’²⁷² Chomsky cautions that

this is not to say that [the speaker] is aware of the rules of the grammar or even that he can become aware of them, or that his statements about his intuitive knowledge of the language are necessarily accurate²⁷³

and goes on to state that

²⁶⁷ Lerdahl and Jackendoff 1983, 112.

²⁶⁸ Chomsky 1965, 24

²⁶⁹ Chomsky 1965, 60.

²⁷⁰ Chomsky 1965, 4–5.

²⁷¹ Chomsky 1965, 8.

²⁷² Chomsky 1965, 8.

²⁷³ Chomsky 1965, 8.

Any interesting generative grammar will be dealing for the most part, with mental processes that are far beyond the level of actual or even potential consciousness ... Thus a generative grammar attempts to specify what the speaker actually knows, not what he may report about his knowledge.²⁷⁴

Thus one cannot determine whether or not a given structural description of a sentence generated by a grammar is correct by appealing to the judgements that a native speaker makes about his or her own linguistic competence. To determine whether or not a given structural description is correct, one must therefore be able to logically deduce from the grammar that certain empirically observable behaviour will be exhibited by the speaker if and only if the structural description is correct. However, Chomsky also claims (and I agree with him) that this could not be achieved:

It is unfortunately the case that no adequate formalizable techniques are known for obtaining reliable information concerning the facts of linguistic structure ... There are, in other words, very few [i.e. no] reliable experimental or data-processing procedures for obtaining significant information concerning the linguistic intuition of the native speaker.²⁷⁵

Thus, Chomsky insists—illogically—that a ‘grammar of a language [must purport] to be a description of the ideal speaker-hearer’s intrinsic competence’²⁷⁶ despite the fact that the speaker-hearer’s intrinsic competence is, as he himself admits, ‘neither presented for direct observation nor extractable from data by inductive procedures of any known sort.’²⁷⁷

11.3 Chomsky was not sufficiently concerned with performance: a theory of competence has no value in isolation—it needs to be incorporated into a performance theory before it has any value

Lerdahl and Jackendoff claim that

it has come to be widely accepted that a theory may address only competence (as much linguistic theory does) or only competence and performance (as many theories in psycholinguistics and artificial intelligence do), and still be of great explanatory value.²⁷⁸

However, I disagree that ‘a theory may address only competence.’ In my view, a theory must address at least competence and performance if it is to be testable. If a generative grammar could be used to construct a computer program that in a ‘Turing test’ situation, conversed with the user in such a manner that the user was unable to determine whether or not the program was a real human, then one could claim some degree of success for such a program and for the grammar that was incorporated in it. But in my view the value of a generative grammar resides *entirely* in the extent to which it can be incorporated in a theory of performance in this way because whereas it may be possible to define a satisfactory empirical criterion for deciding categorically whether or not a sentence is acceptable, it would never be possible to define such a criterion for grammaticalness. A Chomskyan generative grammar therefore becomes partially testable only when incorporated into a performance theory of linguistic behaviour.

²⁷⁴ Chomsky 1965, 8.

²⁷⁵ Chomsky 1965, 19.

²⁷⁶ Chomsky 1965, 4.

²⁷⁷ Chomsky 1965, 18.

²⁷⁸ Lerdahl and Jackendoff 1983, 305.

In *Aspects of the Theory of Syntax*, Chomsky discusses briefly the possibility of constructing a performance theory by supplementing a competence theory in the form of a generative grammar by a system of performance filters.²⁷⁹ But for the most part, he consistently rejects the view that ‘the data of performance exhaust the domain of interest to the linguist’ and believes that linguists should be primarily concerned with issues ‘pertaining to the deeper systems that underlie behavior.’²⁸⁰ Furthermore, he believes that to outlaw the ‘use of introspective data in the attempt to ascertain the properties of these underlying systems’²⁸¹ is ‘to condemn the study of language to utter sterility.’²⁸² But refusing to accept that introspective data has any empirical significance only ‘condemns the study of language to utter sterility’ if one is interested exclusively in competence, believes that a theory of competence has some scientific value in itself and therefore believes that it is worthwhile to study competence in isolation from performance. But in isolation, any theory of competence is purely speculative and empirically untestable. I do not deny that speculation is one of the best ways to generate new, empirically testable hypotheses, but in science speculation is never an end in itself. Speculation as an end in itself is called philosophy. My criticism of Chomskyan mentalism is essentially a mirror image of Chomsky’s criticism of behaviourism: just as Chomsky claims that ‘the behaviorist position is ... an expression of lack of interest in theory and explanation,’ so I claim that Chomsky’s mentalist position reflects a complete lack of interest in formulating empirically testable hypotheses about the facts of linguistic performance. Moore and Carling have made essentially the same criticism:

Whatever the various elaborations and modifications of Chomsky’s theory that are under discussion, the output of each in the end has to be tested against native speaker intuitions. It has, however, always regrettably remained the case that these intuitions have never been subject to rigorous testing. This rather cavalier attitude to external justification was taken early in the development of the theory and set the direction the work was unfortunately to take.²⁸³

11.4 *Musicians cannot reliably determine how they perform musical skills by introspection*

Just as ‘a speaker’s reports and viewpoints about his behavior and his competence may be in error’²⁸⁴ so there can be no doubt, as Johnson-Laird has noted, that ‘musicians themselves do not have conscious access to the processes underlying their production of music.’²⁸⁵ Johnson-Laird goes on to suggest that

a direct way to convince those who may doubt [this] is to ask them to devise a computer program that produces a musical improvisation ... If one had conscious access to the complete processes underlying such skills, the demand would be trivial. Existing programs for improvisation ... however, have only the most rudimentary abilities because programmers, even if they are competent musicians ... cannot discern the basis of their abilities merely by introspection.²⁸⁶

²⁷⁹ Chomsky 1965, 10–15.

²⁸⁰ Chomsky 1965, 193.

²⁸¹ Chomsky 1965, 193.

²⁸² Chomsky 1965, 194.

²⁸³ Moore and Carling 1982, 82.

²⁸⁴ Chomsky 1965, 8.

²⁸⁵ Johnson-Laird 1991, 292.

²⁸⁶ Johnson-Laird 1991, 292.

Johnson-Laird has also pointed out that this applies equally in the case of non-improvised music, since ‘if composers had introspective access to all the principles that guide the sequence of chords in their compositions, then the nature of harmony would not be controversial.’²⁸⁷ As Ebcioğlu has pointed out, a consequence of this fact is that ‘actual composition of music in any decent style is invariably easier than characterizing precisely what that style is in terms of concrete attributes.’²⁸⁸ It may therefore be safely assumed that musicians cannot reliably determine by introspection exactly how they exercise their musical skills.

Steedman requires his grammar to meet two criteria. First, it must ‘generatively specify the set of all and only the chord sequences that are recognized by those familiar with the music as possible jazz 12-bars;’²⁸⁹ and second, ‘each rule [in the grammar] must have what in a language grammar would be called a clearly defined *semantics*.’²⁹⁰ The first of these criteria is essentially the same as Chomsky’s requirement that a linguistic grammar must *weakly generate* the set of all and only sentences in a language. By the second criterion, Steedman means that the rules of the grammar must “‘make sense” musically,’²⁹¹ or, more precisely, that

where a rule of the grammar says that one sequence of chords may replace another, musicians should agree that the substitution is a possible expression of such aspects of the musical meaning as [for example, an] underlying cadential sequence.²⁹²

This second condition is therefore closely related to Chomsky’s requirement of descriptive adequacy, that is, that a grammar must *strongly generate* all and only the correct structural descriptions of sentences in a language. Also, just as Chomsky attaches greater importance to whether or not a grammar is descriptively adequate than to whether or not it weakly generates all and only the sentences of a language, so Steedman attaches greater importance to his second criterion than he does to his first:

a more important criterion than overgeneration or undergeneration remains the extent to which the rules and descriptions that they ascribe to the sequences accord with the intuitions of those who know the musical ‘language’ involved.²⁹³

However, he takes this order of priorities as justification for attaching more importance to whether or not his rules ‘feel right’ than to whether or not they actually account for the tangible results of the mental processes that he is attempting to model. Thus, although he ostensibly defines a ‘testing corpus’ of nine 12-bar chord sequences taken from Coker’s *Improvising Jazz* (1964) that he claims (despite its very small size) is a ‘representative sample of modern jazz 12-bar chord sequences’²⁹⁴ which ‘can be assumed ... [to] illustrate a wide ... range of permissible variations,’²⁹⁵ he does not seem overly concerned with the fact that his grammar is incapable of generating even this tiny, so-called ‘testing corpus.’²⁹⁶ Steedman admits that his grammar *overgenerates* as well:

²⁸⁷ Johnson-Laird 1991, 305.

²⁸⁸ Ebcioğlu 1992, 322.

²⁸⁹ Steedman 1984, 56.

²⁹⁰ Steedman 1984, 56.

²⁹¹ Steedman 1984, 56.

²⁹² Steedman 1984, 56.

²⁹³ Steedman 1984, 73.

²⁹⁴ Steedman 1984, 56.

²⁹⁵ Steedman 1984, 55.

²⁹⁶ Steedman 1984, 69–70.

the attempt to keep the rules as few and as simple as possible has meant that they will generate some sequences that they should not.²⁹⁷

But, like Chomsky and others who concern themselves with competence in isolation from performance, Steedman admits that these mental processes are not necessarily conscious:

‘We are talking about unconscious rules here, not conscious articulate knowledge.’²⁹⁸

Therefore, even if a rule ‘feels right,’ it does not logically follow that it correctly describes these unconscious mental processes. Likewise, if a rule ‘feels wrong,’ it does not logically follow that it incorrectly describes them. Therefore whether or not a rule ‘feels right’ does not necessarily have any bearing whatsoever on whether or not it correctly describes the mental processes that it is an attempt to model. However, the fact that a rule cannot account for the tangible results of these mental processes (in Chomsky’s case, verbal utterances and written sentences; in Steedman’s case, naturally occurring 12-bar jazz chord sequences) proves beyond doubt that the rule does *not* correctly describe these mental processes. Therefore, even if one’s principal aim is to model mental processes rather than the tangible products of these processes, one’s primary concern should *still* be whether or not one’s model actually generates a ‘testing corpus.’

11.5 A representation algorithm cannot be defined for a theory such as GTTM that makes hypotheses about how listeners will interpret pieces of music—therefore such theories are empirically untestable

A number of critics have complained that the analyses generated by *GTTM* do not correctly describe how they interpret musical passages. For example, Peel and Slawson claim that Lerdahl and Jackendoff’s analysis of ‘O Haupt voll Blut and Wunden’ ‘are seriously counterintuitive, resulting in readings of the piece that obscure many of its features both plain and subtle.’²⁹⁹ Similarly, Ebcioğlu complains that a Time-Span Reduction generated by *GTTM* ‘makes events that are adjacent in the music look unrelated, both by connecting them to different parents, and by assigning them to widely different levels on the tree.’³⁰⁰

If *GTTM* is intended to generate correct representations of listeners’ interpretations then in order to test the theory, one would first have to take a score or a performance and generate a representation from it using the theory; and then one would have to examine a listener’s interpretation of the piece and determine whether or not the representation generated by *GTTM* is a correct representation of the listener’s interpretation. However, Lerdahl and Jackendoff nowhere specify an algorithm for deriving a representation of the type generated by their theory from the *interpretation* of a listener. But, unless a representation algorithm can be defined that specifies precisely how an analysis generated by *GTTM* is intended to relate to a listener’s interpretation, it is impossible to determine whether any given analysis is a ‘correct’ description of any particular listener’s understanding of a piece.

²⁹⁷ Steedman 1984, 73.

²⁹⁸ Steedman 1984, 53.

²⁹⁹ Peel and Slawson 1984, 276.

³⁰⁰ Ebcioğlu 1987b, 118.

There are a number of reasons why one could not determine the correctness of an analysis by appealing to the judgements of the listener himself. For example, the listener may decide that he doesn't want to tell the truth—how can one prove that dissenters like Peel and Slawson, and Ebcioğlu, for example, are not simply *pretending* to find the analyses inadequate because they have a vested interest in the refutation of Lerdahl and Jackendoff's theory? Also, subjects would need to have a good understanding of Lerdahl and Jackendoff's theory if they were to be able to judge whether or not analyses generated by it were good descriptions of their interpretations. However, if subjects have a knowledge of the theory then their interpretations will undoubtedly be strongly influenced by this knowledge. Therefore, one could not attempt to test the theory by consulting subjects themselves because only unsuitable subjects—those with a knowledge of the theory—would be able to judge whether or not the analyses generated by it were good descriptions of their interpretations.

In any case, as discussed above, human musicians are not very good at determining by introspection exactly how they perform mental musical tasks, therefore subjects' reports as to whether or not representations are good descriptions of their interpretations must always be assumed to be very unreliable. In my view, because a listener's interpretation cannot be empirically observed even by the listener himself, one could, in fact, *never* define an algorithm that takes a listener's interpretation as input and generates a representation of any kind as output. This leaves much room for dispute over the interpretation of the analyses. Given two experienced listeners, it is logically possible that one will consider a given *GTTM* analysis to be a poor description of his interpretation and the other will think it a good description even if their interpretations of the piece are identical and their interpretations of the analysis are both correct within the bounds of the theory. Therefore a theory like *GTTM* that purports to make predictions about the nature of listeners' interpretations would always be untestable in isolation.

11.6 The value of a theory of listening depends entirely upon whether or not it can be used in a performance theory of a musical skill

The fact that empirically refutable predictions cannot be logically deduced from predictions about how a listener interprets a piece, implies that the value of a theory of listening depends entirely upon whether or not it can be used in a performance theory of a musical skill. That is, predictions about how a listener will hear a piece need to be supplemented by further hypotheses about how the way that one hears a piece affects how one *behaves* in certain situations. For example, Lerdahl and Jackendoff suggest that 'the performer of a piece of music, in choosing an interpretation, is in effect deciding how he hears the piece and how he wants it heard.'³⁰¹ One might therefore hypothesize that because he interprets a piece in a particular way, an expert performer will introduce certain variations in tempo, expression, articulation and so on that are not explicitly represented in a score. One could then test this hypothesis by implementing it as a computational model that takes as input a data file representing a score and generates as output expressive performances intended to be similar to those that would be produced by an expert performer. One could then empirically assess the model by comparing its output with, for example, digital recordings of performances by expert musicians. Gerhard Widmer (1995, 1996), has in fact developed a program that does exactly this, based on computational explications of *GTTM* and Narmour 1977. If such a

³⁰¹ Lerdahl and Jackendoff 1983, 63.

program employs notions or hypotheses from a theory of listening (as Widmer's does) then if the program is successful, this corroborates the theory of listening. However, if the program is unsuccessful, this does not necessarily refute the theory of listening since a 'competence' theory of listening only ever forms *part* of a completely testable 'Competence→Performance Filters' theory. And one problem with such performance theories is that when they are refuted, often one cannot pinpoint exactly where the problem lies. Consequently, it can be very difficult indeed to refute a theory of listening. I would even go so far as to say that they are *irrefutable* because one can never prove that a theory of performance constructed on the 'Competence→Performance Filters' model that incorporates a theory of listening as its competence component fails on account of the theory of listening that forms one part of it.

Lerdahl and Jackendoff, however, do not acknowledge that the value of their theory resides entirely in whether or not it can be used as the competence component in a 'Competence→Performance Filters' type performance theory of a musical skill. They claim that

a rule of musical grammar [is] an empirically verifiable or falsifiable description of some aspect of musical organization, potentially to be tested against all available evidence from contrived examples, from the existing literature of tonal music, or from laboratory experiments.³⁰²

Moreover, they seem to think that they are bolstering this claim by recounting that 'time and again in the course of developing the theory [they] discovered examples for which [their] musical intuitions did not conform to the predictions of [their] then-current set of rules.'³⁰³ They state that as a result of such 'refutations,' they were forced on many occasions 'to invent a new rule or, better, to come up with a more general formulation of the rules [they] had.'³⁰⁴

But the fact that they felt forced to change the rules during the development of their theory does not show that the theory is *intersubjectively* testable. It shows merely that Lerdahl and Jackendoff themselves could come to an agreement in certain situations that some of their rules were wrong. But agreement between two people that some rule or other is wrong hardly constitutes the 'consensus within the community of experts' that Kuhn (1970) requires of a scientific theory or the intersubjective empirical falsifiability that philosophers such as Popper (1983) would demand.

³⁰² Lerdahl and Jackendoff 1983, xii.

³⁰³ Lerdahl and Jackendoff 1983, xii.

³⁰⁴ Lerdahl and Jackendoff 1983, xii.

12 Normative theories that arise from an over-concern with the form of a theory

12.1 *Steedman thinks that 'real' pieces of music are bound to break the rules*

In the following passage Steedman effectively states his view that one should be content with theories for works of art to which there are exceptions:

It is almost a definition of a work of art that it will *break* the established rules in some way, as in the ... example of a piece that exploits the harmonic disconnectedness of a move from C to F sharp and back again... works of art are still constrained by the need for the violations of the rules to be clearly perceived to *be* deliberate, rather than random errors. For this reason ... the difference between the base of rules and the violation is usually quite clear. Studies like the present one can therefore confine themselves to the base rules, and exclude such usages for principled reasons.³⁰⁵

But what are 'the established rules' and why are they allowed to become 'established' if they do not account rigorously and comprehensively for the works of art that they are presumably designed to be able to account for? Steedman's example of a piece of music which includes a chord motion from C to F sharp and back again to C does not show that works of art break rules by definition. This is a pointless attitude to take because it precludes the possibility of developing theories that actually account in interesting and insightful ways for these unusual artistic phenomena. Moreover, it denies the possibility of learning much more from those cases where the rules are not obeyed than merely that they are exceptions to these preconceived rules. Rather, his example shows quite simply that what he refers to as the 'established rules' are refuted by certain observable musical phenomena and should therefore be considered due for replacement by a new set of rules that *does* account for these phenomena. Theories for works of art, just like other theories, should not be normative. I do not think that theories for the creative processes involved in the production of works of art are perpetually condemned to being sets of 'base rules' to which many actual works of art must be considered exceptions.

Steedman appears to take the view that a music theory should fulfil a *normative* role. The main objection to this view is that what is 'normal' and what is 'exceptional' depends entirely on the theoretical perspective adopted on the phenomena being studied. It might be possible for phenomena that must be considered 'exceptional' when one adopts one theory, to be considered 'normal' when one adopts another. More importantly, it might be possible to find a theory that does not require one to view *any* observed phenomena as exceptions. But the view that theories should fulfil a normative role allows one to be *content* with theories that do not actually account for the phenomena that they are ostensibly designed to account for. That is, it allows one to be *content* with theories to which there are known exceptions. But by adopting this attitude, one runs the risk of being seduced by a neat but simplistic theory that does not actually account for all the data and consequently failing to find a possibly *even more* insight-bearing, symmetrical and elegant theory that *does* account for all the data.

³⁰⁵ Steedman 1984, 55, footnote 2.

12.2 *Not undergenerating is more important than having simple rules or an intuitive theory*

Because theories of linguistic or musical competence are not required to generate *acceptable* structures, but only ‘grammatical’ ones, the decision to study competence in isolation from performance leads to investigators being more concerned with the *form* of their rules than with whether or not their grammars weakly generate the languages or musical styles that they are ostensibly intended to account for. For example, Chomsky claims that

it would be quite impossible to characterize the unacceptable sentences in grammatical terms [because] we cannot formulate particular rules of the grammar in such a way as to exclude them. Nor ... can we exclude them by limiting the number of reapplications of grammatical rules in the generation of a sentence.³⁰⁶

In other words, Chomsky would be unwilling to compromise the apparent simplicity and symmetry of a grammar so that it only generated acceptable sentences.

In my view, one should never be content with a normative theory that does not account for *all* the data that one has set out to account for, even if the theory has a very high degree of simplicity or formal elegance and *very nearly* accounts for all the data. I think that the developer of a style theory for a musical style can *always* afford to account for known members of the style—for example, members of the corpus—even at the expense of making their theories more ad hoc and inefficient.

In his treatment of parallel fifths and octaves, James Snell exhibits just such a willingness to abandon simplistic normative rules in the face of data that refutes them. In his program, although ‘the need to avoid parallel fifths and octaves is incorporated into the design of the pitch rules, ... there is by intention no mechanism to guarantee their total prohibition’³⁰⁷ because ‘such a strict prohibition would disallow certain compositions in the tonal literature.’ He goes on to suggest that the simplistic rule to avoid parallel fifths and octaves altogether might be replaced by a rather more complex rule that has the virtue of being obeyed in significantly more cases:

The correct rule might read something like: ‘Parallel fifths are strictly prohibited if they arise at the same level, but are allowed sparingly if they result from processes at two different levels.’³⁰⁸

Snell thus demonstrates a healthy concern that the ‘universal set of well-formed scores’ specified by his rules should contain ‘the tonal literature.’ That is, he is at least to this extent using the literature as the principal guide for the adequacy of his rules and attempting to avoid being normative.

12.3 *The first priority in the development of a style theory should be to ensure that the theory generates all known members of the corpus*

There is no doubt in my mind that the primary aim in the development of a style theory should be to find a composing algorithm that generates a universal set of well-formed scores that is equal to a well-defined style specified by means of a corpus and an acceptability algorithm. Moreover, I think that such a theory should be tested continually for undergeneration and overgeneration until it is refuted. Also, because the

³⁰⁶ Chomsky 1965, 12.

³⁰⁷ Snell 1979, 32.

³⁰⁸ Snell 1979, 77.

universal set of well-formed scores must contain at least all members of the corpus known to the theory developer, it would seem reasonable to ensure that this is the case by using these corpus scores as test cases throughout the development process. That is, the first goal in the development of a style theory should be to develop a composing algorithm whose universal set of well-formed scores contains all members of the corpus known to the developer.

Baroni, Dalmonte and Jacoboni (1989) decided not to attempt to account for certain features of text repetition that occur in two of the Legrenzi arias in their corpus because they considered these features to be ‘exceptions’ and ‘isolated occurrences.’³⁰⁹ In support of this decision, they state that their ‘aim was to describe regularities, [they] thus decided to disregard such cases, at least until such time as they can be considered in a wider historical context.’³¹⁰ I think Baroni et al. were wrong to decide to view certain features of pieces in their corpus as exceptions, particularly in view of the small size of their corpus. If their theoretical perspective does not allow them to account for seventeen closely related pieces without resorting to considering some features in these pieces to be ‘exceptions,’ what chance is there of them being able to generalize their theory so as to be able to account for other repertoires?

Similarly, Ebcioğlu has suggested that ‘for the sake of keeping the theory simple and avoiding risk in overall output quality, sometimes it is better to allow mismatches ... with the composer's decisions.’³¹¹ I am quite willing to admit that a theory can remain useful long after it has been refuted. But I do not think that one should be *content* with refuted theories. If a theory has been refuted, I think one should be actively looking for a better one that accounts for everything accounted for by the old theory and is not refuted by the exceptions to it. Cross has suggested, on the grounds of an argument deriving from Lakatos 1970 that ‘it could be argued that it is the *status* of what is refuted that determines whether or not a new theory is required’³¹² and I certainly agree with him insofar as the *status* of what is refuted determines how much damage has been done to the theory by the refutation. However, I would prefer to say (rather degeneratively, I admit) that a new theory is required when a situation arises that no existing theory can cope with. Theories continue to be used as long as they are the most useful theory in certain classes of situation. Thus, for example, Einstein’s theory of general relativity can account for a number of observations that refute the theory of Newtonian mechanics. But Newtonian mechanics continues to be used in most everyday situations because the predictions it makes are sufficiently accurate and the theory is much easier to use than Einstein’s.

Thus, when I say that I do not think that one should be *content* with refuted theories, the reader should note that I have emphasized the word ‘content.’ The point I am trying to make here is that if the apparently widespread view that music theories should be *normative* is allowed to continue and grow in strength, then it seems to me that there is a danger that people will grow to be more and more *content* with refuted theories and that this contentment will lead to a stagnation in the development of music theory. If all existing theories are refuted then there should be an energetic research effort towards finding theories that can account for the refuting instances. I am *not*

³⁰⁹ Baroni et al. 1989, 36.

³¹⁰ Baroni et al. 1989, 36.

³¹¹ Ebcioğlu 1996a.

³¹² Cross 1996c.

saying that once a theory has been refuted it should be completely discarded, ignored and forgotten.

12.4 *The distinction between ‘practical’ pedagogical theories and ‘scientific’ ones is unproductive and promotes the development of normative ‘practical’ theories and unimplemented, speculative ‘scientific’ ones*

Baroni claims that

the traditional distinction between the normative functions of didactic texts and the descriptive function of theoretical texts, now commonplace in many cultural fields (linguistics is a case in point), has still to make its appearance with sufficient clarity in Italian musical studies.³¹³

In fact, the distinction between normative pedagogical texts and descriptive ‘scientific’ ones has a long and well-established tradition in tonal music theory. As Ebcioğlu has noted, ‘a number of treatises on composition attempt to describe the free compositional style ... but such treatises do not characterize the existing style of any master, they often reflect a particular normative view of music.’³¹⁴ In support of this he mentions the normative pedagogical treatises of D’Indy (1912), Durand (1898), and Czerny (1979).

But Baroni seems to be suggesting that it would be a *good* thing if ‘the distinction between the normative functions of didactic texts and the descriptive function of theoretical texts’ were carried over into musical studies. I disagree with Baroni that such a distinction is one that should be encouraged. I do not think that theories of tonal musical styles should *aim* to be normative. What is the point in teaching and learning rules that do not account for real phenomena? This admittedly makes it easier in the short-term for the teacher and also for a young student. But eventually an intelligent pupil will wish to know why he must obey rules that are consistently broken by the composers whose music he is ostensibly attempting to imitate. In the long-term, the policy of teaching simplistic, normative rules makes it harder for the pupil because he is forced into the view that nearly all ‘real’ pieces of music contain ‘exceptions’ to the rules that he has been forced to learn. Indeed, it is not unusual for teachers to promote the view that there is some positive correlation between the greatness of a piece of music and the extent to which it ‘breaks the established rules.’ Clearly, the fact that ‘real’ music ‘breaks the established rules’ implies quite simply that the ‘established rules’ are inadequate, that they ought to be changed and that they ought *not* to be taught, because teaching a normative theory merely encourages the incorrect assumption that music theory *has* to be normative. But to my knowledge, surprisingly few significant efforts have been made to produce theories that aim to characterize *accurately* the styles of master composers.

Baroni states that he uses the term ‘grammar’ rather than ‘music theory’ because ‘to use the term ‘grammar’ instead effectively brings out the distinction between practical theory (compositional treatises) and scientific studies which do not have any practical functions.’³¹⁵ But in the pure natural sciences, the distinction between ‘scientific studies’ and ‘practical theory’ does not exist. Nor does it exist in applied

³¹³ Baroni 1983, 195.

³¹⁴ Ebcioğlu 1987b, 82.

³¹⁵ Baroni 1983, 183.

science and technology, where one has to use *real* physics to design machines and buildings, otherwise they will not work—one cannot use ‘pretend’ physics to construct a bridge or build an aeroplane! Similarly, one should not teach a normative tonal theory to a student wishing to learn how to write tonal music in the style of some master composer. Admittedly, in physics and engineering one might use a refuted theory in a situation where the theory is known to be sufficiently accurate and where it is easier to use than other theories that are known to be more accurate. But this simply demonstrates that theories are there to be *used*. In my view, the distinction between ‘practical’ pedagogical theories and ‘scientific’ ones is unproductive and promotes the development of normative ‘practical’ theories and unimplemented, speculative, ‘scientific’ ones.

12.5 *It is possible in principle to accurately characterize musical styles algorithmically*

Rader claims that ‘almost no one would dispute the statement that it is impossible for a machine to create any aesthetically pleasing piece of art, be it music, painting or poetry.’³¹⁶ But I believe Kassler provides a strong argument against this view. Kassler points out that

the most significant distinction of natural [i.e. spoken] from musical languages is the presence of a referential component in the former which is lacking in the latter—a component that connects linguistic utterances with the non-linguistic world outside.³¹⁷

Kassler goes on to claim that

it is this component—or, more precisely, our present lack of understanding how to organise this component algorithmically—that has rendered infeasible, at least for the foreseeable future, the tasks of machine translation, machine information retrieval, automatic question-answering, and automatic speech recognition.³¹⁸

Kassler suggests that

the investigation of the structure of musical languages—which have many of the important properties of natural language yet lack the one major component that has rendered intelligent language-processing machines beyond the limits of current feasibility—should attract artificial-intelligence researchers for its potential to clarify the position of these limits.³¹⁹

In other words, the fact that music does not contain a ‘referential’ component means that it might be possible in principle to develop a satisfactory theory of musical structure that contained only a ‘syntactic’ component and did not require components corresponding to the more poorly understood semantic and phonological components of a linguistic theory. Therefore it would be at least possible in principle to accurately characterize musical styles algorithmically. As Ebcioğlu says,

often it is taken for granted that mechanical music cannot have emotional content. Unfortunately, existing computer generated compositions in the traditional style sometimes confirm this opinion. However, the factor responsible for the apparent

³¹⁶ Rader 1974, 631.

³¹⁷ Kassler 1975, 1–2.

³¹⁸ Kassler 1975, 1–2.

³¹⁹ Kassler 1975, 1–2.

lack of feeling is more often than not an inadequate program which lacks the knowledge base to characterize a sufficiently sophisticated style.³²⁰

Ebcioğlu concludes—correctly, in my opinion—that ‘there is no inherent theoretical problem against an algorithmic description of music with emotional content.’³²¹

³²⁰ Ebcioğlu 1987b, 87.

³²¹ Ebcioğlu 1987b, 87.

13 Ebcioğlu, Ames and Cope believe that not overgenerating is more important than not undergenerating

Ebcioğlu's CHORAL program takes a representation ('an alphanumeric encoding'³²²) of a chorale melody as input and generates as output a complete score of a harmonization of the input melody in standard staff notation together with a 'Schenkerian' analysis of the input melody and the bass part of the harmonization represented in 'Schenkerian slur and notehead notation.'³²³ Ebcioğlu describes the quality of the harmonizations generated by CHORAL as approaching that which would be expected of a 'talented student of music who [had] studied the Bach chorales'³²⁴ and claims that the program 'has also produced good hierarchical voice leading analyses' of input melodies.

Although Ebcioğlu states that CHORAL is intended to generate 'harmonizations of four-part chorales in the style of Johann Sebastian Bach,'³²⁵ it must be noted that his primary concern was that the program should 'produce very high quality music in a reasonable time'³²⁶ and he emphasizes that he was 'not primarily interested in validating a cognitive model for a composer.'³²⁷ Consequently, as he himself admits, there are many features of Bach's own harmonizations that could not be reproduced by the program. For example, the program cannot harmonize chorales in triple time whereas a number of Bach's own harmonizations are in triple time (e.g. 'Aus meines Herzens Grunde' (BWV 269), 'Nun lob, mein Seel, den Herren' (BWV 17/7) and 'Puer natus in Bethlehem' (BWV 65/2)). Also, CHORAL never allows crossing of parts and always doubles the root on the final chord of a phrase whereas Bach occasionally does not double the root on the last chord of a phrase³²⁸ and quite frequently allows crossing of parts.³²⁹

Ebcioğlu also points out that although 'the program does try hard to make the output look like Bach (preferring 'Bachian' cadence patterns, repeated suspensions in an inner voice, and so on),' it 'has some additional restrictions based on good voice leading principles that lead to rules that are sometimes more strict than Bach.'³³⁰ Ebcioğlu claims that 'allowing such bad voice-leading would globally affect all outputs negatively.'³³¹ In other words, he believes that if he were to remove those voice-leading rules in the program that are not obeyed by Bach, then although this would lead to the universal output set of CHORAL containing more of Bach's own harmonizations, it would also mean that it contained a higher proportion of unacceptable harmonizations.

³²² Ebcioğlu 1992, 312.

³²³ Ebcioğlu 1992, 312.

³²⁴ Ebcioğlu 1992, 312.

³²⁵ Ebcioğlu 1987b, abstract.

³²⁶ Ebcioğlu 1992, 296.

³²⁷ Ebcioğlu 1992, 322.

³²⁸ For example, Bach does not double the root on the last chord of the first phrase of BWV 159/5 (no.61 in Bach 1990), nor on the last chord of the third phrase of BWV 244/3 (no.78 in Bach 1990).

³²⁹ Examples of crossing parts: BWV 153/1 (no.3 in Bach 1990), bars 7–8; BWV 86/6 (no.4 in Bach 1990), bars 10–11; BWV 351 (no.19 in Bach 1990), bar 8.

³³⁰ Ebcioğlu 1996a.

³³¹ Ebcioğlu 1996a.

Ebcioğlu admits that CHORAL ‘tries to avoid generating a bad harmonization, at the cost of rejecting some harmonizations that Bach could have written’ and explains that he took this ‘conservative approach’ because ‘a computer program is more vulnerable to accusations of unmusicality than Bach is’ and therefore ‘cannot get away with bad voice leading.’³³² He claims that ‘the value of the rules and heuristics of [his] program is in the pieces that it generates that are *different* from the composer’s music.’³³³ This suggests that Ebcioğlu is content with the fact that the universal output set of CHORAL does not even contain certain pieces in the corpus that defines the style that CHORAL is presumably an attempt to characterize. But as I pointed out in section 12.3 above, the universal set of well-formed scores defined by the composing algorithm of a style theory must certainly contain all members of the corpus known to the theory developer. Therefore, if Ebcioğlu’s goal in developing CHORAL was to algorithmically characterize the style of Bach’s chorale harmonizations, then I believe his first priority should have been to ensure that his program was capable of generating all surviving chorales known to have been composed by Bach himself.

On the other hand, I can sympathize to some extent with Ebcioğlu’s decision to compromise CHORAL’s feasibility as an algorithmic characterization of the style of Bach’s chorales in the interest of obtaining consistently ‘high quality’ output. As Ebcioğlu points out, computer scientists and musicologists alike are very quick to criticize composing programs for being ‘unmusical.’ It is much harder to convince a jury of the value of a composing program that overgenerates but has not been proved to undergenerate, than it is to impress them with the output of a program that consistently generates pretty music but cannot even account for pieces in the corpus of the style being modelled. Unfortunately, a computer program of the latter type serves no significant scientific purpose and promotes the normative view that it is impossible to characterize correctly and algorithmically the musical styles of master composers.

When I wrote to Ebcioğlu and presented him with this criticism of CHORAL, he sent me the following reply, which I quote in full because it is not publicly available:

I completely agree with you that generating the corpus is essential when trying to create a cognitive model of a composer. On the other hand, a music composition student who is aiming to harmonize chorales in the style of Bach, does not want to copy Bach’s harmonizations exactly; she wants to create new harmonizations that:

- 1 - are very similar to Bach’s harmonizations, and demonstrate her knowledge and very detailed study of the style, and
- 2 - have a high degree of musicality.

If confronted with remarks that a progression is ‘out-of-style’ or ‘wrong, ungrammatical,’ the student should be able to defend her work by examples taken directly from Bach’s chorales. So, a composition student displays her understanding of a style by trying to write new pieces that are rigorously in that style (this is regarded as useful training). But in this context, it is not desirable for the student to write the same music as the original composer.

CHORAL is more a model of a composition student who has studied Bach’s chorales in great detail, than a model of Bach. It would not be fair to criticize CHORAL because it cannot generate some Bach chorales exactly, just as it would not be fair to criticize a music composition student who has written an harmonization in the style of Bach, because she did not write exactly the same harmonization as Bach (on the other hand, if the harmonization shows a thorough

³³² Ebcioğlu 1996a.

³³³ Ebcioğlu 1996a.

study of Bach's chorale style and is musical, one should appreciate the student's work). So there is a musical purpose to CHORAL that is self-consistent and logical.³³⁴

Of course, even a computational model of the skill of imitative harmonization in the style of Bach would still have to contain Bach's own harmonizations in its universal output set, since these would certainly be amongst those that were 'in the style of' his harmonizations. In any case, trying to produce a computational model of a harmony student who *fails* to consistently simulate the style of some master composer is almost as strange as trying to model listeners' *actual* interpretations of pieces of music rather than trying to provide them with the insight that would allow them to achieve richer interpretations.

Also, although Ebcioğlu states quite categorically in the above passage that 'CHORAL is more a model of a composition student who has studied Bach's chorales in great detail, than a model of Bach,'³³⁵ it must be noted that he is not entirely consistent about the goals that he was attempting to achieve in developing CHORAL. For example, he claims that although 'Bach sometimes doubles the leading note ... when there is a melodic reason for it ... it is arguable whether a computer should.'³³⁶ But it is only '*arguable* whether a computer should' if the purpose of the program has not been precisely defined. For example, if the goal of CHORAL had been to characterize the set of all and only chorale harmonizations in the style of Bach's own, then there is no question that it should have incorporated some model that could account for those occasions on which Bach doubled the leading note.

The fact that CHORAL 'does try hard to make the output look like Bach'³³⁷ seems to imply that Ebcioğlu is fairly confident that at least some of Bach's own harmonizations are members of the 'universal set of well-formed scores' generated by the program. If this is so, then it might be possible for CHORAL to serve as the composing program of an unrefuted algorithmic style theory that aims to characterize the style of some definable subset of Bach's harmonizations that contains all those that CHORAL *can* account for, excludes those that it *cannot* account for, and does neither by definition.

Ebcioğlu does in fact point out that Bach has a number of 'different chorale styles, ranging from florid to austere.'³³⁸ He cites the various harmonizations of 'Jesu meine Freude'³³⁹ as examples of Bach's 'florid' style and 'Christus, der ist mein Leben' (BWV 281, no.6 in Bach 1990) as an example of his 'austere' style. Ebcioğlu states that CHORAL 'has a florid style' and generates 'more passing tones and chord changes than Bach would for his austere-style harmonizations.'³⁴⁰ This suggests the possibility of CHORAL being a successful characterization of a style defined by a corpus of Bach's 'florid' harmonizations.

Just as some of the harmonizations generated by CHORAL are stylistically indistinguishable from Bach's own harmonizations, so Charles Ames claims that the pieces composed by his Cybernetic Composer program 'are realistic enough that an

³³⁴ Ebcioğlu 1996b.

³³⁵ Ebcioğlu 1996b.

³³⁶ Ebcioğlu 1987b, 131.

³³⁷ Ebcioğlu 1996a.

³³⁸ Ebcioğlu 1996a.

³³⁹ e.g. BWV 87/7, no.96 in Bach 1990.

³⁴⁰ Ebcioğlu 1996a.

unknowing listener cannot discern their artificial origin.’³⁴¹ As mentioned in section 1.1, Cybernetic Composer automatically composes pieces in four popular styles—‘standard’ jazz, Latin jazz, rock and ragtime. But, like Ebcioğlu, Ames’ principal concern seems to have been to produce a computer program that consistently generates pieces that are stylistically indistinguishable from those composed by expert human composers in the styles modelled. For example, he states that the main motivation behind using heuristics is ‘the quality of solutions that such strategies reliably produce.’³⁴² There is no evidence that Ames was at all concerned with whether or not the universal output set of Cybernetic Composer contains any pieces by the human composers whose styles the program is presumably an attempt to simulate.

Similarly, although the pieces generated by Cope’s EMI program seem to be of a generally high quality,³⁴³ it seems likely that EMI is in general not theoretically capable of regenerating the pieces given to it as input. EMI takes as input a set of pieces chosen by the user. The program then compares these pieces in order to generate a list of similarities between them. This list of similarities is then used to produce a vocabulary of ‘signatures,’ or motivic fragments, that are used frequently in the input set of pieces. The signatures are then used as building blocks to generate new pieces.³⁴⁴ In a sense, the program ‘learns’ a style and then produces new pieces based on what it has ‘learnt.’ However, Cope does not make clear whether or not the program is theoretically capable of re-generating the pieces in the input set. Simply because a program can generate pieces that are ‘in the style of’ a set of chosen pieces, does not necessarily imply that the program could re-generate that set of chosen pieces.

However, in view of the fact that the music generated by the program is based upon music ‘that conforms to traditional melodic, harmonic, and voice-leading rules,’³⁴⁵ and in the light of the work of, for example, Schottstaedt (1984, 1989), Rothgeb (1968, 1980) and Ebcioğlu (1982, 1985, 1987, 1988, 1990, 1992), that has shown that such traditional tonal music theory is not, in general, satisfactory by itself for accounting rigorously and completely for many structural features of human-composed tonal music, it must be assumed that Cope’s program would not be capable in general of reconstructing the pieces that are used as input. In grammatical terms, it could reasonably be assumed that EMI would in general undergenerate with respect to a corpus consisting only of the set of pieces that are used as input to the program. Therefore, like Ames and Ebcioğlu, Cope was willing to sacrifice the feasibility of his program correctly characterizing human musical styles in the interests of generating output of a consistently high quality.

³⁴¹ Ames and Domino 1992, 186.

³⁴² Ames 1992, 55.

³⁴³ See Cope 1991, Chapters 5 and 6 for examples of output from EMI.

³⁴⁴ See Cope 1991, 152–154 for a description of the EMI algorithm.

³⁴⁵ Cope 1991, 152–3.

14 ‘A grammar based only on a corpus will undergenerate.’ (Johnson-Laird 1991)

In my opinion, the developer of a style theory should be primarily concerned with whether or not his or her theory weakly generates the style being modelled and that he or she should not worry particularly about whether or not the rules ‘feel right’ or generate intuitively correct structural descriptions. But this does not logically imply that I would be satisfied with an ad hoc theory.

Johnson-Laird has pointed out that

a grammar based only on a corpus will undergenerate. The theorist’s task is to go beyond the data, and to base a grammar on a plausible extrapolation from them. The concomitant risks are to overgenerate if the grammar is too bold, or to undergenerate if it is too close to the data.³⁴⁶

In other words, the more ad hoc a theory—that is, the more closely it is based on a corpus, to use Johnson-Laird’s terms—in general the *easier* it will be to find new members of the corpus or new examples of pieces in the style that the theory cannot account for. For example, if one’s goal was to characterize the style of a corpus containing all and only those scores notationally equivalent to scores in Klaus Schubert’s edition of the Bach’s chorales (Bach 1990), then there would be little point in producing a composing algorithm that merely generated one of the scores in Bach 1990 at random each time it was executed. A style theory with a composing algorithm like this would be very easily refuted. For example, such a composing algorithm would not even be able to generate an acceptable chorale that differed from one in the corpus by only one note.

Kippen and Bel’s BOL Processor grammar for tabla drum sequences is, in my view, an example of a style theory that suffers from being based too closely on a corpus. Having developed an initial grammar capable of accounting for a large class of acceptable tabla drum sequences, Kippen and Bel discovered an improvisation that could not be accounted for using this initial grammar but that ‘nevertheless, ... was recognized by knowledgeable listeners to be a masterful improvisation.’ Kippen and Bel decided ‘that if their models [were] intended to be truly representative of tabla playing, then they too must be able to cope with the complexity and structural variety contained in performances from a range of different social contexts.’³⁴⁷ This example demonstrates that Kippen and Bel were primarily concerned with whether or not their grammar weakly generated the ‘language’ of the tabla drum style that they were intending to model.

However, they go on to note that ‘experience has shown that a grammar rarely remains unchanged following the processing of a new set of examples’³⁴⁸ which strongly suggests that in setting up weak generation as their primary objective, they did not pay enough attention to *generalizing* from the data with the result that each grammar produced during the research programme was usually refuted by the next example from the corpus to be discovered. My own style theory described in Part 2 below suffers from a similar shortcoming.

³⁴⁶ Johnson-Laird 1991, 303.

³⁴⁷ Kippen and Bel 1992, 222.

³⁴⁸ Kippen and Bel 1992, 222.

This example highlights the importance of generalizing early in a style theory research programme. After one has found a possibly quite ad hoc theory that actually accounts for a test corpus, one must immediately attempt to generalize from this data in the best way possible before simply trying to account for more examples from the corpus. However, the problem of being over-concerned with accounting for a test corpus at the expense of producing an ad hoc and thus easily refuted theory is relatively rare amongst existing style theories. It is much easier to find examples of researchers who have been all too easily satisfied with simple over-generalizations at the expense of producing theories that do not account for existing human-composed pieces in the styles being modelled (e.g. Lidov and Gabura, Ebcioğlu, Ames and Steedman). Kippen and Bel appear to have made the less common error of spending a little too long *not* worrying about the ‘ad hocness’ of their theory.

I therefore endorse Baroni et al.’s attempt to find rules ‘which were not ... mere descriptions of [their] sample’ of 17 Legrenzi arias and their decision to ‘formulate rules which are not supported by concrete examples in the sample.’ Unlike Kippen and Bel, Baroni et al. correctly aimed ‘to describe a much wider hypothetical Legrenzian repertory to which the 17 arias considered [belonged]’³⁴⁹—a goal which, in my view, is symptomatic of a healthy tendency to generalize early and not base the grammar too closely on known members of the corpus of the style being modelled.

³⁴⁹ Baroni et al. 1989, 42.

15 The need to develop new basic concepts in tonal theory

The generalizations found in traditional harmony and counterpoint textbooks are neither sufficient nor necessary for characterizing real tonal styles. One is therefore justified in being suspicious of any theory that claims to provide an explicit and comprehensive account of tonal music using traditional, normative notions of harmonic or metric structure such as those that might be found in a harmony and counterpoint primer. For example, Lerdahl and Jackendoff state that they

take as given the classical Western tonal pitch system—the major-minor scale system, the traditional classifications of consonance and dissonance, the triadic harmonic system with its roots and inversions, the circle-of-fifths system, and the principles of good voice-leading. Though all of these principles could and should be formalized, they are largely idiom-specific, and are well-understood informally within the traditional disciplines of harmony and counterpoint.³⁵⁰

But in my view, other theoretical perspectives are possible and should be investigated. In particular, I disagree that the principles of ‘the classical Western tonal pitch system’ ‘are well-understood informally within the traditional disciplines of harmony and counterpoint.’ I think that all that has been definitely shown to be true of these basic traditional tonal theoretical ideas is that they are inadequate for describing the structures of real pieces. In my view, tonal theorists can afford to be much bolder and more eclectic in searching for more powerful basic theoretical concepts that provide better generalizations on the structures of tonal pieces.

During the course of developing the algorithmic style theory described in Part 2 of this thesis, I made a number of interesting music-theoretical discoveries. For example, I developed a generalized calculus that correctly explicates expert intuitions of diatonic pitch relations and a generalized theory of metric structure that correctly predicts the relative order of metric strength of any location in a bar in any possible time signature. Also, I have discovered that scale type pitch sets possess a special and highly suggestive graph-theoretical property that is not possessed by any other types of pitch set, and I have used this property in a new explication of the concept of a ‘key’ which has been implemented in my composing algorithm. My search for new and more powerful theoretical concepts for describing tonal music was motivated by the manifest inadequacy of traditional concepts. Previous researchers have tended to rely rather more heavily on traditional and Schenkerian concepts even though such concepts have never led to the development of an unrefuted theory for any particular tonal style. James Snell’s theory, for example, is developed only for major mode pieces but he claims that it could be extended easily to the minor mode by using certain auxiliary hypotheses:

To apply to pieces in minor, most of [the theory’s] structure can stand, but certain changes will be needed to take account of the familiar idiosyncrasies of the minor mode, e.g., the natural vs. raised 6th and 7th degrees, and the tendency to use the mediant rather than the dominant as the key most often associated with points of formal articulation.³⁵¹

Snell seems to disregard the possibility of finding more powerful theories that are perhaps radical departures from traditional tonality theory in which the problems

³⁵⁰ Lerdahl and Jackendoff 1983, 117.

³⁵¹ Snell 1979, 59.

encountered by traditional theory when coping with the minor mode disappear and the minor and major mode are treated with equal economy—as they are, for example, in my own theory of harmonic pitch structure described in Part 2 of this thesis.

That Snell is crippled by his unshakeable faith in the basic correctness of Schenkerian theory is apparent from the criticisms that he makes of Kollmann's theory. Snell claims that Kollmann's proposal 'that over the notes of a fundamental bass, in the initial stages of generation of a composition, may be placed not only triads, but seventh-chords' is an 'obvious fault' because 'music theory has presumably by now established at least that seventh-chords occur fundamentally as the result of voice-leading, and do not appear fully-formed, especially at the background level.'³⁵² However, I think Snell's claim for the 'factual' status of the idea that the sevenths of seventh chords should be treated as voice-leading artifacts is premature. In my view, such an idea could never have factual status. Whether or not sevenths are voice-leading artifacts is an issue that can only be decided on grounds of utility. That is, whether or not sevenths 'are' voice-leading artifacts depends entirely upon whether or not regarding them as such helps in the development of more successful theories for tonal styles.

I am willing to admit that Schenker's work and the vast amount of Schenkerian or pseudo-Schenkerian analysis that has followed from it, has shown that *something like* Schenker's theory *may*, when explicated, succeed in accounting rigorously and thoroughly for styles of tonal music. But until someone succeeds in using Schenker's ideas to develop an algorithmic style theory of the type defined in chapter 3 above which remains unrefuted after extensive testing for overgeneration and undergeneration, I think one should reserve judgement on whether or not a basically Schenkerian theoretical perspective is the best one to adopt and I think that one should be as eclectic and vigorous as possible in one's search for alternative theoretical concepts.

³⁵² Snell 1979, 12–13.

16 The non-necessity of global constraints, exclusive recursion and Lerdahl and Jackendoff's 'Strong Reduction Hypothesis'

16.1 *Non-necessity of global constraints*

The amount of context upon which any given compositional decision depends is debatable. As Kassler notes, 'Schenkerian theory permits context as large as the composition itself,'³⁵³ but I do not think that compositional decisions generally depend upon a context as large as this. In my view, to require events in a piece to satisfy constraints involving relationships that extend over the complete duration of the piece is to demand *more* 'coherence' in the structure of a tonal piece than is, in fact, necessary for it to achieve acceptability.

Lerdahl and Jackendoff, however, claim that Schenker's *Ursatz* constitutes the most stable 'background' structure expressible within the tonal system, in that it embodies many of the basic harmonic and melodic principles of tonality.' Furthermore, they claim that they demonstrate that 'the *Ursatz* is an effect, not a cause, of tonal principles' and that 'from this it follows that reductions of tonally unstable pieces probably will not result in a stepwise melodic descent, or possibly even a I-V-I progression.' They suggest that instead of making 'such cases conform somehow to an a priori conception, it is illuminating to see how they deviate from prototypical cases.'³⁵⁴

However, Lerdahl and Jackendoff do *not* derive their *Ursatz*-like 'basic form' from 'tonal' or 'perceptual' principles. They actually take the *Ursatz* as an a priori principle. Also, the fact that many acceptable pieces do not reduce to an *Ursatz* demonstrates that it is not *necessary* for a piece to be reducible to the *Ursatz* in order for it to achieve acceptability. To take the view that non-reducibility to the *Ursatz* implies that a piece is atypical is to adopt a normative view of tonality based on a non-representative sample from existing tonal pieces. The stylistic variety within the tonal idiom cannot be satisfactorily represented by a corpus of eighteenth and early nineteenth century pieces of Western music (many of which may well be reducible to the *Ursatz*). Rather, to satisfactorily represent the tonal idiom, one would have to define one's corpus to contain a wide variety of music ranging from perhaps 15th century chansons to twentieth century rock music, and from Monteverdi's madrigals to Mahler's symphonies. Clearly, many of these pieces could not be convincingly reduced to the *Ursatz* so there is no sense in which the *Ursatz* can be viewed as 'prototypical' of tonality.

Lerdahl and Jackendoff suggest that a listener

will hear fairly accurately the details [in a piece of tonal music] (except when his mind wanders) and the largest connections, but will be vague about some of the intervening relationships.³⁵⁵

They go on to note that

³⁵³ Kassler 1975, 6.

³⁵⁴ Lerdahl and Jackendoff 1983, 139–40.

³⁵⁵ Lerdahl and Jackendoff 1983, 111.

the Gestalt psychologists, for example Koffka (1935), recognized transposition of a musical passage as a way of changing a musical surface that preserves recognizability [and] took this as important evidence for a mental representation that involves not just a list of pitches, but an abstract representation in which relations among pitches are more important than the actual pitches themselves.³⁵⁶

Lerdahl and Jackendoff assert that ‘the concepts, examples, and arguments’ they present provide evidence for musical cognition of relationships not just between events adjacent on the musical surface, but between structurally important events at various reductional levels that are potentially far apart on the musical surface.³⁵⁷

However, experiments carried out by Cook (1987) suggest that

listeners only have a direct perception of tonal closure when the time-scale involved is in the order of a minute or less. In other words it seems that when we talk about the tonal coherence of movements lasting several minutes, we are not talking about what people actually hear at all.³⁵⁸

But clearly, if such a global relationship cannot be heard, then it cannot possibly be a necessary condition on the acceptability of a piece of music to a *listener*. Therefore, I think it is fair to conclude that even expert listeners who do not have absolute pitch are actually not very good at detecting ‘the largest connections’ such as, for example, that pieces—particularly long pieces—begin and end in the same key. In any case, there are many highly successful pieces of tonal music that do not begin and end in the same key. For example, most piano rags end in the key a perfect fifth below that in which they begin. Ravel’s Piano Concerto for the Left-Hand begins in E minor and ends in D major. But such pieces do not sound inconclusive to the listener. Unless the listener has absolute pitch, he or she will not realize that the key in which such pieces end is different from that in which they begin.

Moreover, I think the main reason why ‘whole-piece’ coherence is not a structurally necessary condition on the acceptability of tonal pieces is precisely that ‘relations among pitches are more important than the actual pitches themselves’³⁵⁹ and there is an upper limit on the duration over which listeners can perceive such relationships. Consequently, it is precisely the long-range relationships that listeners will fail to perceive and that therefore do not contribute to a piece’s acceptability.

16.2 *Exclusive recursion and the use of syntactic categories*

Lerdahl and Jackendoff claim that the tree-like structural descriptions generated by a Chomskyan generative grammar ‘relate grammatical categories, which are absent in music’ and that this is a ‘basic fact’ that is ‘one of the crucial differences between language and music.’³⁶⁰ They assert that whereas

linguistic trees represent *is-a* relations: a noun phrase followed by a verb phrase *is a* sentence, a verb followed by a noun phrase *is a* verb phrase, and so forth, ... the fundamental hierarchical relationship among pitch-events is that of one pitch-event being an *elaboration of* another pitch-event—the latter [being] the structurally more important event of the two. Thus a suspension is an elaboration of its

³⁵⁶ Lerdahl and Jackendoff 1983, 111.

³⁵⁷ Lerdahl and Jackendoff 1983, 111.

³⁵⁸ Cook 1989, 120.

³⁵⁹ Lerdahl and Jackendoff 1983, 111.

³⁶⁰ Lerdahl and Jackendoff 1983, 112–113.

Eight-bars	→	First-four	Second-four
First-four	→	Opening-cadence	Opening-cadence
	→	Opening-cadence'	Opening-cadence
Second-four	→	Middle-cadence	Opening-cadence
Opening-cadence	→	I	I
	→	I	V
Opening-cadence'	→	I	III
	→	I	IV
Middle-cadence	→	I	IV
	→	I	V
	→	IV	I

Figure 16-1

resolution [and] the events en route in [the derivation of] a phrase are elaborations of either the phrase's structural beginning or its cadence.³⁶¹

Lerdahl and Jackendoff thus believe that 'unlike language, music is not made up of grammatical categories that combine in highly constrained ways.'³⁶² But I do not think that it is justifiable to claim this as an a priori assumption. In fact, the assumption that music is made up of grammatical categories seems far more plausible to me than the assumption that every surface event (but one) must be an elaboration of another event. Peel and Slawson similarly 'suspect, contrary to Lerdahl and Jackendoff, that there *are* grammatical categories in tonal music of the eighteenth and nineteenth centuries and that their combination is highly constrained.'³⁶³

Ebcioğlu notes that a prolongational reduction constructed according to *GTTM* 'brings together, as the leftson and rightson of some non-terminal node, chord-events which are not adjacent in the music' and points out that a consequence of this is that 'the analyst is faced with the task of constructing the tree such that these non-adjacent chords that are artificially brought together in the tree do form a reasonable progression with respect to each other.'³⁶⁴ Indeed, what possible grounds can there be for making the a priori assumption that in order to be an acceptable piece of tonal music, notes widely separated in the surface of the piece must satisfy the same class of constraints as those that must be satisfied by notes that are immediately adjacent in the surface?

In practice, passages in tonal music perform certain typical structural functions that could quite reasonably be termed 'syntactic.' For example, a grammar that attempted to characterize the class of acceptable classical sonata form movements might at the least detailed level begin with two rules along the lines of the following (elements in square brackets being optional):

Sonata-Form-Movement → *Exposition* + *Development* + *Recapitulation* [+ *Coda*]

Exposition → [*Introduction* +] *First-Subject* + *Transition* + *Second-Subject* + *Codetta*

Also, it must be noted that a number of relatively successful attempts have been made to characterize musical styles using grammars that employ non-terminal symbols,

³⁶¹ Lerdahl and Jackendoff 1983, 112–113.

³⁶² Lerdahl and Jackendoff 1985, 157.

³⁶³ Peel and Slawson 1985, 166.

³⁶⁴ Ebcioğlu 1987b, 117.

syntactic categories and ‘is-a’-type rules rather than ‘elaborates’-type rules. Johnson-Laird (1991), for example, attempted quite successfully to generate the class of acceptable modern jazz chord sequences using the context-free grammar reproduced in Figure 16-1 to generate simple sequences of triads that were then transformed into ‘enriched chord sequences’ using a system of context-sensitive transformational and substitution rules.³⁶⁵

16.3 Lerdahl and Jackendoff’s ‘Strong Reduction Hypothesis’

In their ‘Strong Reduction Hypothesis,’ Lerdahl and Jackendoff claim that:

1. ‘Pitch-events are heard in a strict hierarchy’
2. ‘Structurally less important events are not heard simply as insertions, but in a specified relationship to surrounding more important events.’³⁶⁶

They state that they ‘find it difficult to envision a theory lacking the Strong Reduction Hypothesis that would be both sufficiently rich and sufficiently constrained to constitute a plausible account of musical cognition.’³⁶⁷

The Strong Reduction Hypothesis implies that in the opinion of Lerdahl and Jackendoff, a successful grammar of tonal melody would necessarily consist *exclusively* of rewrite rules that take one of the following two forms:

$$n_1 \rightarrow n_2-n_1$$

$$n_1 \rightarrow n_1-n_2$$

where n_1 and n_2 are notes. This is equivalent to the claim that any successful theory of tonal melody will involve reducing the melody to other generally shorter melodies until one ends up with one note. It is a claim that any time-span (i.e. group) is an elaboration of a single note and that therefore, because a piece is a group (GWFR2³⁶⁸), that the melody of a whole piece is an elaboration of a single note. This is an *extremely* strong claim. It is also, in my view, a rather far-fetched one, because it implies that the basic syntactic function of any group (including that extending over the whole piece) could be carried out in some essentially equivalent way by a single note. In other words, it is the claim that every group is structurally synonymous with the event that is the ‘head’ of that group (TSRWFR1³⁶⁹).

In my opinion, a far more plausible assumption than the Strong Reduction Hypothesis would be that at all reductional levels less detailed than that which corresponds roughly to the least detailed, Schenkerian ‘foreground,’ the structure of a piece is expressed in terms of non-terminal syntactic categories whose organization is characterized by what Lerdahl and Jackendoff call ‘is-a’ type rules. The most detailed sequence of non-terminal symbols is then ‘realized’ as notes that actually appear explicitly in the musical surface in the first ‘foreground’ level and these notes are then elaborated roughly in the way that Lerdahl and Jackendoff suggest in the more detailed foreground levels. In other words, I do not think that the idea of *elaboration* can profitably be carried beyond one or two recursive levels below the musical surface. For

³⁶⁵ Johnson-Laird 1991, 310–312.

³⁶⁶ Lerdahl and Jackendoff 1983, 106.

³⁶⁷ Lerdahl and Jackendoff 1983, 152.

³⁶⁸ Lerdahl and Jackendoff 1983, 38.

³⁶⁹ Lerdahl and Jackendoff 1983, 158.

all deeper levels, I think it is safe to assume that the structure of a piece must be expressed in terms of non-terminal syntactic categories.

Lerdahl and Jackendoff state that ‘the relation among subordinate and dominating groups’ in their grouping structure theory ‘does not differ from level to level or change in some substantive way at any particular level, but is essentially the same at all levels of musical structure.’³⁷⁰ They therefore ‘assert that grouping structure is recursive ... that is, it can be elaborated indefinitely by the same rules.’³⁷¹ But surely the relationships that must obtain between large-scale sections in an acceptable tonal piece cannot possibly be exactly the same as those that must exist between, say, individual phrases? Similarly, it seems highly implausible to me that the rules governing the organization of phrases in tonal music must necessarily be exactly the same as the rules that govern, say, the way that motivic fragments are arranged to construct those phrases. In other words, I find it hard to believe that it is profitable to assume that the rules for determining grouping structure are the same at each level of structure.

As Lerdahl and Jackendoff explain, in a reduction that obeys the Strong Reduction Hypothesis, each ‘event that is elaborated is retained along with the event(s) that elaborate it’ so that, for example, the ‘structural beginning and the cadence of a phrase’ that are the most ‘important’ events in a phrase ‘do not disappear or convert into something else in the course of fleshing out the phrase as a whole.’ Lerdahl and Jackendoff contrast this with the situation in language where ‘grammatical categories are not retained in the tree structure from level to level, but break down into other categories.’ For example, ‘a verb phrase may break down into a verb plus a noun phrase, which in turn may break down into an article plus a noun, and so on.’ They claim that ‘a mere transference of [such] linguistic trees’ that employ non-terminal symbols ‘into their musical counterpart would be misguided from the start.’³⁷² But in my view it is a mistake to assume that every symbol that occurs in every level of a reductional tree must be explicitly represented in the musical surface. I think it could very well be fruitful to view sequences of consecutive events in the musical surface as being the results of rewriting non-terminal symbols that are not explicitly represented in the surface.

Lerdahl and Jackendoff criticize Keiler (1975, 1978) for making the kind of ‘transference of linguistic trees’ described in the previous paragraph. They also criticize his use of ‘grammatical categories such as “tonic prolongation,” “tonic completion,” and “dominant prolongation.”’ and state that, in their view, Keiler’s ‘musical trees suffer from the sort of superficial analogy between music and language for which he rightly criticizes Bernstein (1976).’³⁷³ However, Lerdahl and Jackendoff provide no good arguments against adopting the strategy of employing syntactic categories (i.e. non-terminal symbols) in a grammar of music. On the other hand, it is obviously absurd to suggest that for a piece of tonal music to be acceptable, the necessary and sufficient conditions that must be satisfied by a sequence of symbols in some underlying reductional level that could correspond to widely separated events in the surface, should be identical to those that must be satisfied by a sequence of consecutive events at the surface level. It is much more sensible to suggest that each note in the musical surface performs some category of syntactic function in a sequence of notes that performs some

³⁷⁰ Lerdahl and Jackendoff 1983, 15–16.

³⁷¹ Lerdahl and Jackendoff 1983, 15–16.

³⁷² Lerdahl and Jackendoff 1983, 113.

³⁷³ Lerdahl and Jackendoff 1983, 338, note 4.

category of syntactic function in a sequence of *sequences* of notes that performs some category of syntactic function ... in the piece. The aspect of Bernstein's (1976) analogy between linguistics and music that Keiler found superficial was his suggestion that the categories of syntactic function that would be employed in a theory of music should be essentially the *same* as the syntactic categories that are used in language (i.e. nouns, verbs etc.)

Lerdahl and Jackendoff suggest that 'some readers may balk at extending the notion of reduction to "background" levels—so that, for example, an E \exists major triad is ultimately all that is left of the first movement of the *Eroica*.' They suggest that one might feel such an extension to be 'mechanical, abstract, and irrelevant to perception.'³⁷⁴ In response to such a criticism, Lerdahl and Jackendoff ask, 'Exactly where should a reduction stop?', and suggest that 'there is no natural place, for there is no point in the musical hierarchy where the principles of organization change in a fundamental way.'³⁷⁵ Clearly, finding the best set of musical syntactic categories to employ at reductional levels less detailed than the foreground level will probably be one of the most demanding tasks involved in the development of an algorithmic style theory. However, I certainly do not think that the idea of exclusive recursive elaboration all the way back to a single event will ultimately prove to be the most profitable strategy to adopt in an attempt to algorithmically characterize a tonal style. In particular, Meredith 1993 is an account of an attempt to derive algorithmically from the scores of a corpus of pieces, the most economical set of syntactic categories capable of describing the structures of those pieces on a number of different hierarchical levels.

³⁷⁴ Lerdahl and Jackendoff 1983, 109.

³⁷⁵ Lerdahl and Jackendoff 1983, 109.

17 Chomsky's concept of a universal grammar and the concept of a universal theory of tonal music

17.1 *Explanatory adequacy and the concept of a language acquisition model*

For Chomsky, a universal grammar consists of 'substantive' and 'formal' universals. He defines substantive universals as follows:

A theory of substantive universals claims that items of a particular kind in any language must be drawn from a fixed class of items. ... For example, ... traditional universal grammar ... advanced the position that certain fixed syntactic categories (Noun, Verb, etc.) can be found in the syntactic representations of the sentences of any language, and that these provide the general underlying syntactic structure of each language.³⁷⁶

Chomsky defines formal universals as follows:

Consider a claim that the grammar of every language meets certain specified formal conditions. The truth of this hypothesis would not in itself imply that any particular rule must appear in all or even in any two grammars. The property of having a grammar meeting a certain abstract condition might be called a *formal* linguistic universal, if shown to be a general property of natural languages. Recent attempts to specify the abstract conditions that a generative grammar must meet have produced a variety of proposals concerning formal universals, in this sense.³⁷⁷

I have already pointed out in sections 2.3 and 11.2 that although it would be impossible even to produce a grammar that could be proved to weakly generate all and only the sentences of some natural language, Chomsky makes the even stronger demand that a generative grammar must be descriptively adequate. That is, it must strongly generate all and only the *correct structural descriptions* of sentences in a language. In fact, for Chomsky, even a descriptively adequate grammar would not be completely satisfactory. Chomsky's view is that generative linguists should not merely be aiming to find descriptively adequate grammars for all particular natural languages, rather they should be aiming to develop what he calls 'an 'acquisition model' for language' or 'a theory of language learning or grammar construction.'³⁷⁸ He claims that such an 'acquisition model' would consist of 'first, a linguistic theory that specifies the form of the grammar of a possible human language, and, second, a strategy for selecting a grammar of the appropriate form that is compatible with the primary linguistic data.'³⁷⁹ In other words, an acquisition model would be 'an input-output device that determines a particular generative grammar as 'output,' given certain primary linguistic data as input.'³⁸⁰ If a linguistic theory can function as an 'acquisition model' in this way then it is said to meet the condition of *explanatory adequacy*.

There are thus

³⁷⁶ Chomsky 1965, 28.

³⁷⁷ Chomsky 1965, 28–29.

³⁷⁸ Chomsky 1965, 24–25.

³⁷⁹ Chomsky 1965, 25.

³⁸⁰ Chomsky 1965, 38.

two respects in which one can speak of ‘justifying a generative grammar.’ On one level (that of descriptive adequacy), the grammar is justified to the extent that it correctly describes ... the tacit competence ... of the native speaker. ... On a much deeper ... level (that of explanatory adequacy), a grammar is justified to the extent that it is a *principled* descriptively adequate system, in that the linguistic theory with which it is associated selects this grammar over others, given primary linguistic data with which all are compatible. In this sense, the grammar is justified on *internal* grounds, on grounds of its relation to a linguistic theory that constitutes an explanatory hypothesis about the form of language as such. The problem of internal justification—of explanatory adequacy—is essentially the problem of constructing a theory of language acquisition.³⁸¹

The relationship between an acquisition model for language and a theory of linguistic structure is made more explicit in the following passage where he defines the necessary components that such a linguistic theory must contain:

a theory of linguistic structure that aims for explanatory adequacy must contain

- (13) (i) a universal phonetic theory that defines the notion ‘possible sentence’
- (ii) a definition of ‘structural description’
- (iii) a definition of ‘generative grammar’
- (iv) a method for determining the structural description of a sentence, given a grammar
- (v) a way of evaluating alternative proposed grammars

Putting the same requirements in somewhat different terms, we must require of such a linguistic theory that it provide for

- (14) (i) an enumeration of the class s_1, s_2, \dots of possible sentences
- (ii) an enumeration of the class SD_1, SD_2, \dots of possible structural descriptions
- (iii) an enumeration of the class G_1, G_2, \dots of possible generative grammars
- (iv) specification of a function f such that $SD_{f(i,j)}$ is the structural description assigned to sentence s_i by grammar G_j , for arbitrary i, j
- (v) specification of a function m such that $m(i)$ is an integer associated with the grammar G_i as its value (with, let us say, lower value indicated by higher number)³⁸²

In relation to this definition, Chomsky defines an *explanatory theory* as one that meets conditions (i) to (v) and a *descriptive theory* as one that meets conditions (i) to (iv). Thus the difference between these two classes of theory is that a descriptive theory merely has to provide *at least one* descriptively adequate grammar (G_i in (14iii)) for each language but will, in general, provide more than one— ‘a linguistic theory is *descriptively adequate* if it makes a descriptively adequate grammar available for each natural language’³⁸³—whereas, in order to meet a condition of *explanatory* adequacy, a descriptively adequate linguistic theory must be supplemented with an *evaluation procedure* or *simplicity measure* which is an algorithm that selects the ‘correct’ grammar from the set of descriptively adequate grammars allowed by the ‘definition of grammar’ provided by (iii) that are compatible with what Chomsky calls the ‘primary linguistic data.’ That is, given certain primary linguistic data, the evaluation measure of an explanatory theory must select *exactly one* descriptively adequate grammar from the

³⁸¹ Chomsky 1965, 26–7.

³⁸² Chomsky 1965, 30–31.

³⁸³ Chomsky 1965, 24.

class G_1, G_2, \dots for each language and this grammar must be the one that a human would select given the same primary linguistic data.³⁸⁴

Chomsky claims that a ‘child who has learned a language has developed an internal representation of a system of rules that determine how sentences are to be formed, used and understood’ and that therefore, ‘using the term ‘grammar’ with a systematic ambiguity ... to refer, first, to the native speaker’s internally represented ‘theory of his language’ and, second, to the linguist’s account of this, we can say that the child has developed and internally represented a generative grammar.’³⁸⁵ Therefore, ‘the problem for the linguist, as well as for the child learning the language, is to determine from the data of performance the underlying system of rules that has been mastered by the speaker-hearer and that he puts to use in actual performance.’³⁸⁶

However, as Moore and Carling point out, Chomsky did not actually equate a discovery procedure with a model of the language acquisition process in children—he emphasizes that it is not true that any descriptively adequate theory can be ‘raised to the level of explanatory adequacy’ merely by adding an evaluation measure.³⁸⁷ I believe there are two reasons for this.

First, I believe that Chomsky’s unwillingness to admit to equating the notions of an acquisition model and a discovery procedure may be a direct result of his exclusive concern with competence—that is, with modelling what goes on in the mind. Chomsky seems to believe that it is useful to maintain a distinction between a discovery procedure that successfully generates for each natural language a grammar that weakly generates it given a finite corpus of sentences from the language, and a theory that correctly describes how human children learn languages. Unfortunately, there would never be any way of showing that a successful discovery procedure was not a correct model of the language acquisition process because the two could only differ in respects that are not accessible to empirical observation.

Second, to convert a descriptively adequate theory into an acquisition model one needs to define an algorithm that, when given any set of primary linguistic data for a particular natural language as input, generates as output the ‘most preferred’ grammar for that primary linguistic data from the set defined by the descriptively adequate theory. This involves ‘parsing’ the data and automatically generating a grammar for it. Just as it is only certain classes of grammar that can be expressed as parsers that are capable of determining for any sentence whether or not it is a member of the artificial language defined by the grammar, so the fact that an acquisition model requires such a device for ‘parsing’ primary linguistic data puts severe constraints on what one is permitted to define as the set of ‘humanly feasible’ grammars in a descriptively adequate theory. Chomsky concludes that

for the construction of a reasonable acquisition model, it is necessary to reduce the class of attainable grammars compatible with given primary linguistic data to the point where selection among them can be made by a formal evaluation measure.³⁸⁸

Chomsky therefore thinks that ‘the major endeavor of the linguist must be to enrich the theory of linguistic form by formulating more specific constraints and

³⁸⁴ Chomsky 1965, 34.

³⁸⁵ Chomsky 1965, 25.

³⁸⁶ Chomsky 1965, 4.

³⁸⁷ Chomsky 1965, 35.

³⁸⁸ Chomsky 1965, 35.

conditions on the notion ‘generative grammar’³⁸⁹ and even goes so far as to claim that ‘it is not necessary to achieve descriptive adequacy before raising questions of explanatory adequacy.’³⁹⁰ However, it is hard to accept this view given that every grammar hitherto constructed to account for a natural language is demonstrably incapable even of weakly generating the natural language that it is intended to account for.

An example of the type of restriction imposed on the structure of generative grammar that Chomsky believes linguists should be seeking is his result that a finite state grammar could never even weakly generate the sentences of a natural language.³⁹¹ This implies that the class of grammars allowed by a linguistic theory can exclude the class of finite state grammars.

The claim that linguists should be primarily concerned with determining the form of successful grammars is equivalent to stating that tonal music theorists should be concerned with the problem of determining the fundamental underlying properties of style theories—and in particular, composing algorithms. In fact, some theoretical musicologists have indeed devoted some effort to ‘determining the underlying properties of successful [musical] grammars.’ Roads 1985 is an investigation along these lines. This is interesting and worthwhile in itself, but I do not think it should be carried too far.

My own view is that placing too much importance on the form of rules before one has produced an unrefuted theory for any language or musical style leads to normative theories that do not account for the class of phenomena that they are intended to account for. In adopting Chomsky’s strategy, one runs the risk of being seduced by a neat but simplistic theory and failing to find a possibly even more insight-bearing, symmetrical and elegant theory that actually accounts for *all* the data.

17.2 *The concept of a universal theory of tonal music*

Baroni mentions that Nattiez (1975) ‘gives an account of a hierarchy of style ... from the tonal system down to a single piece by Mozart.’³⁹² An accurate characterization of tonal music would therefore not merely represent tonal music as an undifferentiated ‘universal set of tonal pieces.’ Rather it would represent it as a richly structured hierarchical system of particular styles.

I think one of the main tasks of a formal theory of tonal music should be to allow theorists to economically, insightfully and precisely make hypotheses about the content of the ‘languages’ represented by sets of pieces that are identifiably homogeneous in *any* definable way. For example, a theory of tonal music should allow it to be a relatively simple matter for theorists to model the style of the works of a particular composer, or the works of a particular genre or period. More generally, it should be possible to take *any* set of human-composed pieces that are perceived to be similar or in any sense ‘in the same style’ and use the theory to produce a ‘sub-theory’ for this style. That is, it should be possible to use a theory of tonal music to generatively specify the universal set of pieces in any definable style of tonal composition.

³⁸⁹ Chomsky 1965, 35.

³⁹⁰ Chomsky 1965, 36.

³⁹¹ Chomsky 1957, 18–21.

³⁹² Baroni 1983, 201, note 43.

Kassler (1975) makes the assumption that ‘tonality’ can profitably be considered a ‘language’ of which particular tonal styles can be considered ‘dialects’ and states that in his opinion, a theory of tonal music would be satisfactory if it embodied ‘the knowledge required for automatic identification and structural analysis of compositions instancing tonality.’³⁹³ Thus in transferring the notions of generative linguistics to music, Kassler makes an analogy between concepts such as ‘tonality’ or ‘the twelve-note system’ and a verbal *language*, and an analogy between ‘particular musical styles’ and ‘dialects’ of a verbal language.

My own view is that a more fruitful analogy would be to make a correspondence between concepts such as ‘tonality’ or ‘the twelve-note system’ and what Chomsky calls ‘language as such’, and a correspondence between ‘particular musical styles’ and what Chomsky calls particular languages. Following this scheme, one would search for ‘grammars’ for the ‘particular languages’ of well-defined musical styles, and then attempt to develop a ‘universal grammar’ that generates the universal set of successful ‘particular grammars’ for styles within a musical idiom such as tonality or the 12-note system. In other words, a theory of tonality or the 12-note system would take a form more like a Chomskyan universal grammar than a particular grammar; and a theory for an identifiable style such as that of a particular composer or genre or period would take the form of an algorithmic style theory, which clearly corresponds more closely to Chomsky’s concept of a generative grammar than it does to his concept of a universal grammar.

To make all this a little more precise, I believe that if one is interested in developing a theory of tonal music, then the first step should be to attempt to find a number of correct algorithmic style theories for definable tonal styles. Then, when one has succeeded in developing a number of unrefuted algorithmic style theories for tonal styles, one should attempt to find as many ‘formal and substantive universals’ common to these unrefuted style theories as possible. Finally, one should formalize these universals to produce a ‘universal theory of tonal music’ that takes the form of a hypothesis that the set of algorithmic style theories that satisfy these universals is equal to the set of all and only correct algorithmic style theories for tonal styles.

Notwithstanding Kassler’s suggestion that an ‘intelligent music-processing machine’ should be able to compose ‘coherent new utterances ... even within a particular musical ‘style’ that is a dialect’³⁹⁴ of the language of tonality, his view that Schenker’s theory can be explicated as an attempt to generatively define ‘the class of compositions instancing tonality’³⁹⁵ strongly suggests that he would be satisfied to some extent with a theory of tonal music that was no more than a hypothesis that some artificial language generated by a grammar was equal to the set of all and only pieces of tonal music. Indeed, Brown and Dempster seem to take this astonishingly simplistic view, presenting an informal reworking of Kassler’s explication of Schenker’s theory and hypothesizing quite baldly that ‘any piece of music is tonal if and only if it is derivable within the system.’³⁹⁶ The informal theory that Brown and Dempster present seems to imply that they would be content with a theory of tonal music that merely hypothesized that some generatively defined set of pieces was equal to the set of all and only tonal pieces. They do not demand, as I do, that a theory of tonal music impose a

³⁹³ Kassler 1975, 4.

³⁹⁴ Kassler 1975, 2–3.

³⁹⁵ Kassler 1975, 5.

³⁹⁶ Brown and Dempster 1989, 88.

rich and insight-laden structure on the ‘universal set of tonal pieces’ that would allow one to formulate precise descriptions of identifiable and definable styles *within* the tonal idiom.

Like Kassler and Brown and Dempster, Lerdahl and Jackendoff also seem to think that ‘Western tonal music’ should be viewed as being an example of what would be the musical equivalent of a verbal *language* rather than ‘language as such.’ Lerdahl and Jackendoff state that as they ‘develop [their] rules of grammar, [they] often attempt to distinguish those aspects of the rules that are peculiar to classical Western tonal music from those aspects that are applicable to a wide range of musical idioms.’³⁹⁷

Lerdahl and Jackendoff therefore seem to have attempted in *GTTM* to describe a ‘universal grammar of musical idioms’ by presenting one *particular* grammar from the class of grammars generated by this universal grammar and indicating which parts of this particular grammar are common to *all* the particular grammars in the class of grammars generated by the universal grammar. But I believe that such an approach will lead to a theory that provides only a weak model of the structure of styles *within* the ‘idiom’ of tonal music.

My view that a theory of tonal music would be inadequate if it merely generatively specified the class of tonal compositions in the manner of a particular grammar for a language is endorsed by Baroni, who clearly states that

in dealing with western music, it is impossible to speak of one grammar; every grammar possesses rules common to entire periods, rules belonging to musical genres and rules belonging to individual repertoires.³⁹⁸

Since the early 1970s, Baroni and several collaborators have been attempting to develop ‘a theory of European melody.’³⁹⁹ In particular, they have been seeking a theory that would ‘allow [them] to distinguish structures pertaining to the general form of European melody from structures typical of particular epochs, and also from structures belonging to specific repertoires.’⁴⁰⁰ The type of theory for which they are striving would consist of a number of distinct but related ‘grammars of melody,’⁴⁰¹ each grammar designed to be able to account for a specified style (or ‘repertoire’ as Baroni calls it). Baroni’s research strategy ‘excludes the possibility of conceiving a single grammatical scheme capable, as Schenker imagined, of describing ‘music’ itself (by which he meant tonal music).’⁴⁰²

However, I think that certainly one task of a theory of tonal music should be to define an artificial language that is intended to be equivalent to the universal set of tonal pieces. In other words, I do believe that *one* of the algorithmic style theories generated by a ‘universal theory of tonal music’ should be capable of accounting for the ‘tonal style’ in general. The tonal style, just like any other style, would have to be defined by means of a corpus and an acceptability algorithm as described in chapter 3 above. However, an appropriate corpus for specifying the tonal style would have to contain an extremely wide variety of pieces. For example, it would certainly have to contain the works of a large number of composers ranging in period from the 15th to the 20th

³⁹⁷ Lerdahl and Jackendoff 1983, xiii.

³⁹⁸ Baroni et al. 1984, 202.

³⁹⁹ Baroni et al. 1989, 23.

⁴⁰⁰ Baroni et al. 1992, 187.

⁴⁰¹ Baroni et al. 1984, 201.

⁴⁰² Baroni 1983, 186.

century. Also it would have to contain music in a wide range of genres ranging perhaps from opera to modern rock music.

Clearly then, a theory of tonal music of the type I have proposed above corresponds more closely to Chomsky's idea of a universal grammar than it does to his notion of a generative grammar for a particular language. Just as a universal linguistic grammar first, defines what Chomsky likes to call 'language as such', and second, allows for the specification of particular grammars for particular natural languages (e.g. French and English); so a theory of tonal music would first, generatively specify the tonal style in general; and second, allow for the specification of an algorithmic style theory for any definable tonal style in particular.

In sum, in my view, a satisfactory theory for tonal music would need to be an extensively tested and unrefuted hypothesis that a generatively specified class of algorithmic style theories (in the particular sense defined above) was equal to the set of all and only correct algorithmic style theories for tonal styles. Both Kassler and Snell seem to subscribe to an essentially similar view. Kassler states that a satisfactory theory of tonal music would need to be an intelligent music-processing machine' that was 'able to carry out such 'central' processes as ... composition of coherent new utterances (within a particular musical language, and even within a particular musical 'style' that is a dialect of such a language).'⁴⁰³

Similarly, Snell believes it is necessary to ask 'what further constructs does it take to control [his] system so that the music it generates is constrained to varying degrees, e.g., allowing only the style of the period, or of the composer, or of the collection.'⁴⁰⁴ More specifically, he states that

whereas the primitives [i.e. absolute rules of his system], in their simplest form, apply to all tonal music, and by themselves constitute an exceedingly underdetermined system, the rule-constraints [which roughly correspond to Ebcioglu's heuristics] add restrictions [that cause] the music the system generates [to be] more and more stylistically consistent.⁴⁰⁵

I understand this to mean that the absolute (or 'primitive') rules in Snell's system generate a set of pieces that contains the universal set of tonal pieces as a proper subset—that is, it overgenerates. On the other hand, the universal output set of the system can be constrained in a very flexible manner by introducing and varying the 'rule constraints.' In this way, Snell believes that his system could in principle be used to generatively define subsets of the set of pieces generated by his absolute rules alone and that these subsets could be made equal to recognizable styles within the 'universal set of tonal pieces.' That Snell's concept of what would constitute a satisfactory theory of tonal music is similar to my own is clear from the following passage:

Inevitably, with each new composer, style, and type of composition, there will be added rule parameters and constraints, and perhaps even new rules. Rather than have a different system for each category of work, it would be satisfying to bring the systems together into one, by generalizing the necessary operations.⁴⁰⁶

Although Lerdahl and Jackendoff seem to think that 'Western tonal music' should be viewed as being an example of what would be the musical equivalent of a verbal

⁴⁰³ Kassler 1975, 2–3.

⁴⁰⁴ Snell 1979, 60.

⁴⁰⁵ Snell 1979, 32.

⁴⁰⁶ Snell 1979, 63.

language rather than ‘language as such,’ they do suggest that it might be possible to model the perceived stylistic similarity and disparity between pieces *within* the ‘tonal idiom’ using their theories of Time-Span and Prolongational Reduction as follows:

The most global levels of reductions should represent relations characteristic of the tonal idiom as a whole. Relations characteristic of a particular piece should begin to emerge at somewhat more intermediate levels, showing precisely how the piece is a unique instance of the tonal idiom.⁴⁰⁷

That is, the structural descriptions generated by their theory for a given piece should provide a correct and detailed description of expert listeners’ intuitions about how the piece relates to other pieces in the ‘tonal idiom.’ Lerdahl and Jackendoff suggest that the degree of tonal ‘prototypicality’ of a piece is roughly a function of the level in its structural description at which features unique to the piece start to occur. Thus, Ravel’s Piano Concerto for the Left Hand and nearly all piano ragtime pieces would need to be considered very atypical because they deviate from what Lerdahl and Jackendoff call ‘the basic form’ (essentially the Schenkerian *Ursatz*) at the most global level of reduction.⁴⁰⁸ As discussed in section 16.1 above, this ‘basic form’ requires that the tonal structure of a tonal piece should typically be reducible to the harmonic progression ‘I-V-I.’ A typical piano rag would reduce to either the progression ‘V-V-I’ or ‘I-I-IV’ depending upon whether the first section is considered to be in the ‘global dominant’ or the ‘global tonic.’ Ravel’s Piano Concerto for the Left Hand would reduce to something like ‘ii-I.’

Lerdahl and Jackendoff claim that

if [they] were to restrict [themselves] to contrived examples, there would always be the danger through excessive limitation of the possibilities in the interest of conceptual manageability, of oversimplifying and thereby establishing shallow or incorrect principles with respect to music in general. Tonal masterpieces provide a rich data sample in which the possibilities of the idiom are revealed fully.⁴⁰⁹

In the present state of knowledge, I think that theorists should be primarily concerned with achieving unrefuted theories for a number of highly homogeneous styles such as, for example, those defined by corpora containing only pieces in the same genre by a single composer. One can then be sure that the styles modelled are ones that it must be possible to account for using ‘grammars’ generated by a universal theory of tonal music. When a number of successful theories for particular tonal styles have been developed, it may be possible to unify and generalize these theories into a single universal theory of tonal music that is much more insight-bearing, symmetrical and elegant than any of the perhaps more ad hoc particular theories that preceded it. I think that theorists will have a better idea as to what form such a universal theory must take after they have developed theories that successfully account for particular, highly homogeneous, well-defined tonal styles. One generally needs to know what works in particular cases before one is in a position to speculate about what will work in general.

Lerdahl and Jackendoff claim that ‘it is essential to begin with more sophisticated examples in order to arrive at any notion of what is going on’ and that an advantage of their approach is that they ‘can deal from the start with a far wider range of literature than the musically extremely limited’ styles generated by theories such as those of

⁴⁰⁷ Lerdahl and Jackendoff 1983, 134.

⁴⁰⁸ Lerdahl and Jackendoff 1983, 189.

⁴⁰⁹ Lerdahl and Jackendoff 1983, 8.

Sundberg and Lindblom (1976) and Kassler (1976).⁴¹⁰ Lerdahl and Jackendoff seem to be advocating that one should develop a theory of tonal music in a ‘top-down’ fashion. That is, in their opinion, one should begin by determining the general formal universals that must be possessed by any successful grammar for a style and then go on to attempt to develop grammars of this type for particular styles. In my view, this ‘top-down’ approach coupled with Lerdahl and Jackendoff’s lack of concern with whether or not musical grammars weakly generate the styles that they are intended to account for, could lead to theorists becoming entrenched in normative theories that employ theoretical notions that are neither necessary nor sufficient to account for the structures of real pieces of music.

Lerdahl and Jackendoff seem to be suggesting that a satisfactory theory of tonal music will arise by examining a small but diverse selection of individual pieces that is intended to represent the ‘four corners’ of the tonal idiom, so to speak. But a theory of tonal music must account for similarity between pieces at *all* levels. I think that Lerdahl and Jackendoff’s strategy would lead to a theory that accounts only weakly and approximately for the perceived stylistic distinctions and similarities between tonal pieces. It may well lead more quickly to a theory that is a hypothesis that a specified set of pieces is equal to the universal set of tonal pieces, but this would not in my view constitute a satisfactory theory of tonal music. I believe that a theory that generates the universal set of tonal *styles* would emerge much more quickly by adopting a strategy in which one first attempts to produce theories for highly homogeneous styles. In fact, I think that theorists should be primarily concerned at the current stage of research with producing theories very like those of Lindblom and Sundberg and Baroni—that is, the type of theory that Lerdahl and Jackendoff denigrate as being ‘musically limited.’

Thus, whereas Chomsky considers that generative linguists should devote their energy towards discovering a priori limits on the internal structure of generative grammars, I believe that currently the primary concern in the development of algorithmic style theories should be to develop theories for particular styles that after extensive testing for overgeneration and undergeneration remain unrefuted. When this has been done for a number of different styles it may be possible to identify certain substantive or formal features that are common to all these unrefuted theories, that can then be used to guide the development of further style theories. Given that no unrefuted style theory yet exists for a tonal style, tonal theorists should not be concerned with whether or not their ‘grammars’ or composing algorithms are members of some identifiable class of algorithms that are well understood in some other field. Rather they should be primarily concerned with developing algorithmic style theories of the type defined in chapter 3 and testing these theories extensively for overgeneration and undergeneration until they become refuted.

The research strategy adopted by Baroni and his collaborators is, in fact, essentially the one that I believe will lead most quickly to the development of a satisfactory theory of tonal music in general. Baroni explains that after producing a successful theory for the Lutheran chorale melodies,

the next phase would be to compare the grammars particular to other carefully chosen groups of melodies; from this comparison further discoveries would be

⁴¹⁰ Lerdahl and Jackendoff 1983, 334, note 8.

made concerning the distinction between what characterizes the melody itself and what characterizes particular sub-classes of melodies.⁴¹¹

He also warns that if one is to avoid developing normative theories that do not weakly generate the styles that they are intended to model then

it is necessary to begin at the bottom, that is, with very small repertoires that can be described exhaustively, and on this basis to pick out the elements common to other collections and to arrange them according to their overall importance. A project of this kind requires the reformulation of the whole system of rules every time the process of analysis is applied to a larger repertoire, but this is the price of ensuring that we stand on a clear and firm foundation.⁴¹²

This is almost an exact description of what I believe would be the most productive strategy to adopt in a research programme directed towards the goal of developing a theory of tonal music in general. This strategy would involve four stages:

1. Find unrefuted algorithmic style theories for particular styles, such as the music by individual composers in particular genres, defined by means of appropriate well-defined corpora and an acceptability algorithm that meets the specification given in section 3.8.
2. Compare these unrefuted theories, abstracting what is common between them.
3. Produce a 'universal theory of tonal music' by formally characterizing those features that these unrefuted theories have in common.
4. Attempt to develop theories for new styles that share those features that the existing successful style theories have in common.

The issues involved in the development of successful musical grammars and the necessary and sufficient conditions that must be satisfied by them will emerge naturally through having to solve the problems involved in actually developing a theory that aims to generate the universal set of acceptable pieces in a particular style. If one can then develop another structurally similar grammar that successfully models a different style then one is on the way to defining the universals that must be possessed by all successful grammars and thus a universal theory of tonal music.

A number of authors have suggested that a universal theory of tonal music that hypothesizes that some generatively specified set of style theories is equal to the universal set of correct tonal style theories, would be a useful tool for performing comparison between the styles of different composers and different periods. For example, Ebcioğlu has suggested that one long-term goal for tonal theory might be to discover 'how the rules and heuristics evolve in the lifetime of a composer, and between different styles in different periods.'⁴¹³ One could model the relationship between a composer's style at different periods in his life by modelling each style by a separate particular grammar generated by a single 'universal grammar of tonality.'

Baroni has also stated on numerous occasions that the ultimate goal of his research is to develop particular grammars for specific melodic styles that can be related by a 'universal theory' of European melody and that this universal theory should allow for the characterization of the historical development of melodic styles and for the formal comparison of styles. He states that his 'ultimate goal is to identify the structural

⁴¹¹ Baroni 1983, 185.

⁴¹² Baroni 1983, 186.

⁴¹³ Ebcioğlu 1996a.

principles of what [he] defines as melody in general and to distinguish them from the specific principles of single melodic repertoires. In other words, [his] intention is to state a theory of melody that both describes particular repertoires and accounts for cultural and historical modifications of melody.⁴¹⁴

Having now carried out studies on a number of different repertoires (Lutheran chorales, French chansons and Legrenzi's arias) Baroni has concluded that 'in the European tradition the rules of musical syntax are much more changeable than those of verbal syntax and are more deeply affected by socio-cultural events.' In particular, he claims that his work 'has shown that the melodies of Lutheran chorales follow a different system of rules from those of Legrenzi's arias, although some of these rules are common to both repertoires.'⁴¹⁵ However, this claim must be taken in the light of the fact that his group has not yet succeeded in producing an unrefuted theory for either of the two styles mentioned.

17.3 *Single vs. multiple viewpoints: the 'hierarchy vs. linearity' debate as a symptom of an over-concern with competence*

As explained in 17.1, the evaluation measure of an explanatory linguistic theory must select *exactly one* descriptively adequate grammar from the class of allowable grammars described by the theory and this grammar is intended to be *the single* grammar that correctly models the 'internalized competence' of the 'idealized native speaker.' But I think that Chomsky is being too restrictive in imposing the constraint that a language must be describable in terms of a *single* grammar. What a priori reason is there to assume that the intuition of native speakers is describable in terms of a *single* grammar? It seems plausible to me that multiple viewpoints and parallel representations might be required if not for a competence theory then at least for a comprehensive and detailed *performance* model of linguistic behaviour. Moore and Carling claim that it was 'Chomsky's view of what constituted a valid scientific explanation' that 'obliged him ... to assume that under-laying [sic] native speaker use of language there is a body of unchanging, independent and uniform linguistic knowledge' and that

by making this idealisation he was able to disregard the fact that actual language in use is dynamic; involving as it does the complex interaction of language users' knowledge, intentions, beliefs and expectations both of one another and of the world as they individually perceive it.⁴¹⁶

The assumption that classes of phenomena can be described in terms of *single* models has also been made in music theory. The following passage from Lerdahl and Jackendoff suggests that they feel that it can be safely assumed that the hierarchical aspect of Chomskyan generative grammar is one that can be profitably transferred to the domain of musical style study:

In the present study we will for the most part restrict ourselves to those components of musical intuition that are hierarchical in nature. ... Other dimensions of musical structure—notably timbre, dynamics and motivic-thematic processes—are not hierarchical in nature, and are not treated directly in the theory as it now stands. Yet these dimensions play an important role in the theory in that they make crucial contributions to the principles that establish the hierarchical structure for a piece.

⁴¹⁴ Baroni et al. 1984, 205.

⁴¹⁵ Baroni, Dalmonte and Jacoboni 1992, 188.

⁴¹⁶ Moore and Carling 1982, 63–4.

The theory thus takes into account the influence of nonhierarchical dimensions, even though it does not formalize them.⁴¹⁷

There are few who study musical style that do not seem to concur with this general view. The most obvious example of a well-developed theory that eschews the concept of hierarchy is, of course, Narmour's (1977, 1990, 1992). But the question of whether or not the concept of hierarchy should be incorporated into a theory for a musical style is, I think, an essentially uninteresting one. My own view is that a theory for a musical style must provide the necessary theoretical apparatus for the construction of successful performance theories (as opposed to competence theories). Hitherto, the most successful performance theories have incorporated some idea of hierarchy. I think that Narmour's ideas would have to have been employed in a successful performance theory for a musical skill that does not employ any notion of hierarchy before he could make any strong claim that hierarchy is not a necessary feature of a theory of tonal music. On this point, it is interesting to note that Gerhard Widmer (1995, 1996) has incorporated both a version of Narmour's theory *and* a version of Lerdahl and Jackendoff's theory as parallel 'views' in a surprisingly successful, fully implemented, computational model of expressive performance. Also, it is perhaps worth noting that the most successful computational models of composition of tonal music produced to date—Ebcioğlu's CHORAL program and Ames' Cybernetic Composer—also each incorporate both hierarchical and linear views in parallel. It seems that the results of successful performance theories strongly suggest that the 'hierarchical vs. linear' debate considered so crucial by those who interest themselves exclusively in competence theories (e.g. Narmour and Lerdahl and Jackendoff) might in practice be shown to be essentially a pointless argument with no practical significance. In practice, it seems that better results can be obtained by employing both a hierarchical view and a linear view *in parallel* than can be obtained by using either view alone.

17.4 *Efficiency as an evaluation measure: one should not decide between competing theories on grounds of efficiency*

Chomsky suggests that length, in terms of the number of symbols used in the statement of a grammar is 'the obvious numerical measure to be applied to a grammar'⁴¹⁸ as a measure of its efficiency and implies that such a measure could constitute alone a complete evaluation procedure. However he points out that simplicity measures are not given a priori—one has to define what one means by 'simple.'⁴¹⁹ Whether or not a particular algorithm or grammar is efficient depends considerably upon the nature of the particular machine upon which it is implemented. What is simple for a brain to do may be very complicated for a von Neumann-type computer (and, of course, vice-versa).

Therefore, as Chomsky points out, if 'we regard an acquisition model for language as an input-output device that determines a particular generative grammar as 'output,' given certain primary linguistic data as input' then 'any proposal concerning [a simplicity measure] is an empirical hypothesis about the nature of language'⁴²⁰ and 'choice of a simplicity measure is therefore an empirical matter with empirical

⁴¹⁷ Lerdahl and Jackendoff 1983, 8–9.

⁴¹⁸ Chomsky 1965, 42.

⁴¹⁹ Chomsky 1965, 37.

⁴²⁰ Chomsky 1965, 37.

consequences.⁴²¹ Consequently, in order to achieve ‘a meaningful measure’ of length it would be necessary to

devise notations and to restrict the form of rules in such a way that significant considerations of complexity and generality are converted into considerations of length, so that real generalizations shorten the grammar and spurious ones do not. Thus it is the notational conventions used in presenting a grammar that define ‘significant generalization,’ if the evaluation measure is taken as length.⁴²²

Chomsky goes on to note that ‘this is, in fact, the rationale behind the conventions for use of parentheses, brackets, etc., that have been adopted in explicit (that is, generative) grammars.’⁴²³

It is also possible to define measures for the efficiency of algorithms. For example, one can compare the average or worst times of execution of two algorithms when implemented in the same programming language by means of the same compiler on the same computer. However, if an algorithm is originally expressed in some form other than such an implementation, it would be necessary (and possibly rather difficult) to show that the program representing the algorithm was a fully optimized implementation of the algorithm. Aho, Hopcroft and Ullman (1983) confirm that this method of comparing the efficiency of two algorithms is ‘popular and useful’⁴²⁴ but suffers from some inherent problems that computer scientists have overcome by adopting a different measure of algorithmic efficiency known as ‘worst-case asymptotic time complexity.’⁴²⁵

However, the relative efficiency of two algorithms when measured in this way could depend on the specific choice of language used for the implementations. That is, given two algorithms, A_1 and A_2 , such that the implementation of A_1 is faster when they are expressed in language L , it is possible that A_2 could be the faster when the algorithms are implemented in a language other than L . This can particularly be the case if one of the algorithms is more suited to implementation in a procedural language, such as C, PASCAL or FORTRAN, and the other more suited to implementation in a declarative language such as PROLOG, or one that allows unlimited recursion such as LISP, or an object-oriented language such as CLOS or C++. This is a special instance of the more general result that the relative apparent efficiency of two algorithms when implemented in a given manner depends on how well-suited each is to being implemented in that manner. Another special case of this is that, given two algorithms, the one that is more efficient when they are implemented as *computer programs* is not necessarily the one that *humans* would find easier or quicker to use. When the algorithm is supposed to be an expression of a theory that is intended to be a source of insight and understanding for humans, this becomes an important consideration.

Let us say that one is attempting to develop an algorithmic style theory for a particular style S defined to be the union of the corpus of scores C and the universal set of acceptable scores defined by an acceptability algorithm A . Let us assume that one has developed two algorithmic style theories, T_1 and T_2 , where T_1 is the hypothesis that the universal set of well-formed scores defined by composing algorithm C_1 is equal to the style S and T_2 is the hypothesis that the universal set of scores defined by composing

⁴²¹ Chomsky 1965, 38.

⁴²² Chomsky 1965, 42.

⁴²³ Chomsky 1965, 42.

⁴²⁴ Aho et al. 1983, 293.

⁴²⁵ Aho et al. 1983, 16–27, 293–305.

algorithm C_2 is equal to the *same* style S . Clearly, it is only style theories that make hypotheses about *exactly the same style* defined in terms of *exactly the same corpus and acceptability algorithm* that are strictly comparable. In my view, it is only in the extremely special case where the universal set of well-formed scores generated by C_1 can be *proved* to be *exactly* equal to the universal set of well-formed scores defined by C_2 that one could even consider discarding one of the two theories in favour of the other on grounds of efficiency. Clearly, if the universal set of well-formed scores generated by C_1 is different from that defined by C_2 then at least one of the two theories is incorrect. In this situation, the only justifiable course is to continue to test both theories for overgeneration and undergeneration until one or both of them is refuted. In other words, in the vast majority of situations, the requirement of weak generation is sufficient to distinguish between two competing theories for the same style.

However, even if C_1 and C_2 had been proved to have identical universal output sets, one would still not necessarily be justified in discarding the theory with the less efficient composing algorithm. In all probability, neither T_1 nor T_2 would actually be correct and there would be no a priori reason to assume that a correct style theory could be developed more easily from the more efficient of the two theories.

18 Instrumentalism, behaviourism and the sufficiency of weak generation as a condition on the adequacy of a style theory

18.1 *Instrumentalism*

Theories that are based upon hypotheses about the nature of processes that cannot be observed can still make empirical predictions provided that these hypotheses lead *logically* to predictions whose truth value can be determined by empirical investigation. But in the case of phenomena that are only partially observable (e.g. the composition of a piece of music), it is generally the case that there is more than one theory that could make successful predictions about the observable parts of the phenomena. Therefore, it is possible that two distinct theories could account for all observable aspects of the same class of phenomena in completely different ways by positing different hypotheses about what goes on in the unobservable parts of the phenomena. In such a case, neither of the two theories could ever be shown to be more ‘true’ than the other. They would have to be judged solely on grounds of utility. However, if it became possible to observe previously unobservable parts of the phenomena accounted for by these two theories—perhaps because of an advance in technology, for example—then the theories would have to be tested against these new observations and it may well be that one of the two theories but not the other would be refuted by these new observations.

However, I think one has to accept that any theory that attempts to ‘explain’—that is, provide a satisfactory model for—any class of phenomena that are apparently the result of empirically unobservable processes, and does so by hypothesizing possible mechanisms for these processes can, in general, never be shown to be ‘actually true’ because one could never establish conclusively that the hypothesized mechanisms were indeed good descriptions of the actual, empirically unobservable processes that give rise to the phenomena being studied. Even basic physical theories—in fact, *particularly* basic physical theories—such as quantum mechanics, attempt to explain observable phenomena as the putative results of unobservable and entirely hypothetical processes involving the interaction of objects that in some cases have never been observed. Therefore, any theory is ‘correct’ only to the extent that it accounts for and does not conflict with empirical observations. One can therefore seldom consider such a theory to be ‘true.’ In most cases, theories can only be considered more or less *useful*.

Also, one should not be surprised if for some classes of phenomena there exist a number of apparently irreconcilable theories, each the most useful model in a particular class of situations. The famous wave-particle duality of quantum theory is an example of this. It may even be that some classes of phenomena can be explained only by means of parallel, logically conflicting theories. If this were the case, it would simply highlight the fact that nature is not constrained to behave in accordance with human logic. As noted in section 6.2, Penrose (1994) has shown that one cannot make the a priori assumption that all natural processes can be correctly modelled using *algorithms*.

Any theory that hypothesizes that certain observable phenomena are the results of unobservable processes can be refuted only by showing that the process described in the theory is incompatible with certain things that are *known* about the process being modelled. For example, if a theory is a hypothesis that a particular algorithm is a correct

description of how Bach harmonized chorale melodies, then it could be refuted by showing that it did not account for certain combinations of notes in harmonizations that are known to be by Bach. Logically, it might also be possible to refute such a theory by showing that the algorithm could not possibly be a correct description on any level (or mixture of levels) of a human mental process.

However, a refutation on grounds of incompatibility with what is known about mental processes would in general be far less categorical than one on grounds of incompatibility with repeatable, empirically and intersubjectively verifiable facts about which notes appear where in a musical score. This is because the degree of certainty with which one can ‘know’ something about a mental process is far lower than the degree of certainty with which one can know empirical facts about how notes are arranged in a score.

For example, Miller (1956) suggests that the upper limit on the number of distinct categories in a unidimensional perceptual domain that can be held simultaneously in short-term working memory is 7 ± 2 . So, for example, a theory that proposed that the mental processes involved in harmonizing a chorale, involved holding 100 separate unidimensional categories simultaneously in short-term working memory, would be incompatible with what is known about mental processes. Of course, as discussed in section 9.4 above, it is unlikely that limitations on the capacity of short-term working memory have any part to play in the composition of *written* music because the composer can compensate for any such limitations by jotting things down on paper as he or she composes. On the other hand, to test the hypothesis that the mental activity of improvising a jazz melody or bass line does not necessarily place more than ‘a minimal demand on the processing capacity of working memory,’⁴²⁶ Johnson-Laird attempted to construct computer programs that improvise jazz melodies and bass lines using only those grammars that make the least demands on working memory—that is, regular grammars. Nonetheless, even the ‘fact’ of an upper limit on human short-term working memory is far less certain than, say, the fact that Bach never wrote a keyboard piece where different key signatures are used simultaneously on different staves, or the fact that the first note in the bass part of chorale BWV 269 is a G natural.

18.2 *Naive Behaviourism vs. Mentalism vs. Instrumentalist Behaviourism*

Lerdahl and Jackendoff state that

Gestalt theory could not withstand the powerful antimentalistic bias prevalent in American psychology during the 1940s and 1950s, and it seems to have been written out of existence by the more ‘scientific’ behaviorist school.⁴²⁷

However, they claim that ‘the shoe is [now] on the other foot’ and that

the success of generative linguistics has played a large role in rekindling interest in mentalistic theories, while behaviorist psychology has been to a great extent discredited by arguments rather similar to those advanced 40-60 years ago by the Gestaltists.⁴²⁸

⁴²⁶ Johnson-Laird 1991, 322.

⁴²⁷ Lerdahl and Jackendoff 1983, 304.

⁴²⁸ Lerdahl and Jackendoff 1983, 304.

They cite Chomsky 1959 as being ‘a significant turning point.’⁴²⁹

In my view, a theory of competence that consists of hypotheses about the nature of mental processes certainly has value. But this value resides *entirely* in the extent to which such hypotheses about mental processes can be employed in a successful theory of performance that makes empirically testable predictions about the observable behaviour or products of this behaviour that putatively result from these mental processes. In other words, I believe that a theory that purports to describe how people *think* certainly has value but becomes empirically testable only when it is supplemented by hypotheses about how the way that people *think* affects what they *do*.

A theory of competence can never be verified or refuted, it can only ever be shown to be more or less *feasible*. And the feasibility of such a mentalist theory depends upon whether or not it can be incorporated into a successful theory of performance. I admit that in some cases, because of limitations on ‘present understanding of the issues,’⁴³⁰ it may only be possible to incorporate a theory of competence into a theory of performance by supplementing it with ad hoc and provisional auxiliary hypotheses. Nonetheless this must be done if one wishes to determine whether or not a theory of competence is feasible.

Lerdahl and Jackendoff point out that certain naive behaviourists felt ‘that no mentalistic theory could be worthwhile without an account of its mechanism.’⁴³¹ More specifically, such behaviourists held the view that in order for a psychological theory to have any value it had to be directly relatable to then-current knowledge in physics and chemistry. As Lerdahl and Jackendoff observe, this led to a number of crass, contrived and extremely premature attempts to explain psychological phenomena in terms of neurophysiology. Lerdahl and Jackendoff give, as examples of such attempts, Koffka’s comparison of the Gestalt Law of *Prägnanz* to ‘physical principles that minimize energy at boundaries between substances’ and Köhler’s (1940) equally far-fetched attempt ‘to make this sort of analogy into a theory by claiming a direct correspondence between the stabilization of perceptual fields and stabilization of electrical fields in the brain.’ As Lerdahl and Jackendoff observe—rather generously, in my opinion—‘this physiological reduction is far too crude for the finely tuned observations it is meant to explain.’⁴³²

I certainly do not subscribe to the view that in order for a psychological theory to have any value it must be relatable to *current* knowledge in the physical sciences. Subscribing to this view implies making the utterly illogical assumption that all human behaviour and psychological phenomena must be explicable in terms of *current* theories in physics and chemistry. But clearly, one could never be sure that all current theories in the physical sciences were correct and together accounted for all possible phenomena. Indeed, as explained above, theories in the physical sciences are, in general, unverifiable hypotheses that often attempt to explain observable phenomena as the putative results of unobservable and entirely hypothetical processes involving the interaction of objects that in many cases have never been directly observed. The fact that theories in the physical sciences are not, in general, *true* has been shown many times over the course of scientific history when new observations are made that refute the claims of current theories.

⁴²⁹ Lerdahl and Jackendoff 1983, 304.

⁴³⁰ Lerdahl and Jackendoff 1983, 333.

⁴³¹ Lerdahl and Jackendoff 1983, 305.

⁴³² Lerdahl and Jackendoff 1983, 305.

Therefore I think that Chomsky is nearly correct when he says that it is the mentalistic studies that will ultimately be of greatest value for the investigation of neurophysiological mechanisms, since they alone are concerned with determining abstractly the properties that such mechanisms must exhibit and the functions they must perform.⁴³³

However, I think this assertion is only *nearly* correct because it is the *performance* theories that simulate human behaviour and that perhaps incorporate the results of such ‘mentalistic studies’ that actually ‘determine abstractly the properties that neurophysiological mechanisms must exhibit and the functions they must perform.’

18.3 *Dennett would be satisfied with a robot that only came very close indeed to behaving indistinguishably from a human being*

Around March 1996, a documentary was broadcast by the BBC on the subject of Alan Turing⁴³⁴ and at one point in this programme, there was an interview with the cognitive psychologist, Daniel Dennett. In the course of this interview, Dennett claimed something along the lines that it would never be worth attempting to produce a robot that behaved in a manner that was 100% indistinguishable from the way a human behaves because the closer you get to a complete simulation, the less you learn and the harder it is to make any more progress.

I fundamentally disagree with this view because clearly, any robot that does not completely simulate human behaviour must be an incorrect model and could well operate in a way that is fundamentally different from the mental processes that lead to human behaviour. In other words, a model that simply gets very close indeed to achieving a 100% simulation of human behaviour does not necessarily tell us *anything* interesting about the way that minds work. It is only a model that behaves completely indistinguishably from a human being that could feasibly be a correct model of human mental processes and thus it is only such complete simulations that might tell us something interesting and fundamental about the nature of the human mind.

Also, while there might in principle be a large number of incorrect models that get very close indeed to achieving a 100% simulation of human behaviour, it seems likely that only a very constrained class of models would be capable of behaving indistinguishably from a human being. In other words, I believe that the requirement of 100% simulation severely—and *sufficiently*—limits the class of possibly correct models.

Thus, whereas Dennett believes that that ‘last grain of similitude’ would not teach us anything of any great importance, it seems to me that it could well be precisely that ‘last grain of similitude’ that will teach us the most important and fundamental facts about the nature of the human mind. Also, as should be clear from the discussion in chapter 14, I do not believe that this ‘last grain of similitude’ could be simply a matter of adding a few ad hoc rules or auxiliary hypotheses, since ad hoc rules are generally the easiest to refute. Indeed, it seems very likely that that ‘last grain of similitude’ would be extremely hard to achieve precisely because it would reflect the presence of fundamental flaws in existing models.

⁴³³ Chomsky 1965, 193, note 1.

⁴³⁴ ‘The Strange Life & Death of Dr. Turing’, broadcast date unknown, transcript unavailable.

18.4 FIB—the more complex the output of a device, the lower the number of ways in which the device might be producing that output

Imagine that one has a computer program—call it FIB—that, when run, simply prints something on the screen each time the user presses the ENTER key. The user's task is to try to consistently predict what FIB will output the next time he presses ENTER. The first time the user presses ENTER, FIB returns

0

Clearly, this tells us very little indeed about how FIB works. The second time the user presses ENTER, FIB returns

1

By this stage the user might be beginning to suspect that FIB simply generates a number each time the ENTER key is pressed. However, the user is still in no position to predict the next output because the number of rule-describable sequences of numbers that begin '0 1 ...' is probably infinite. For example, the next number could be 2 (counting numbers or prime numbers), 4 (square numbers), 0 (alternating 0s and 1s), and so on. The third time the user presses ENTER, FIB returns

1

The user is now becoming confident that FIB will continue to generate numbers. He also knows that the program is not simply generating the counting numbers and that it is generating neither square numbers nor prime numbers. The next time the user presses ENTER, FIB returns

2

And the penny drops—the user notices that the sum of the first two outputs is equal to the third output, and that the sum of the second and third outputs is equal to the fourth. He knows that this rule describes the Fibonacci sequence and his hypothesis is corroborated by the circumstantial evidence provided by the name of the program. So he correctly predicts all subsequent outputs, beginning with the number 3.

This thought experiment demonstrates that, in general, the more complex the output of a device whose mechanism is non-random and rule-describable, the fewer the number of possible mechanisms that device can have.

18.5 Weak generation is a sufficient condition on the adequacy of a generative grammar

As I have already mentioned on a number of occasions, weak generation is, for Chomsky, a necessary but not a sufficient condition on an adequate grammar. Chomsky attempts to justify the view that a grammar should be required to generate correct structural descriptions and not simply correct surface structures by claiming that the two sentences,

- (1) I persuaded John to leave.
- (2) I expected John to leave.

are ‘the same in surface structure, but very different in the deep structure that underlies them.’⁴³⁵ Chomsky suggests that on the evidence of these two sentences alone, one might think that the verbs ‘expect’ and ‘persuade’ always receive parallel syntactic analyses in any pair of sentences that differ only in that where one uses the verb ‘expect’ the other uses the verb ‘persuade’ and that therefore, if one of these sentences is grammatical, the other will also always be grammatical. However, this is not true, as the following two sentences show:

(3) I persuaded John of my sincerity.

(4) I expected John of my sincerity.

This demonstrates that the class of syntactic functions that can be fulfilled by the verb ‘expect’ is different from the class of syntactic functions that can be fulfilled by the verb ‘persuade.’

Chomsky claims that this example demonstrates that ‘surface similarities may hide underlying distinctions of a fundamental nature’⁴³⁶ and that therefore it is not sufficient to demand merely that a grammar generate the class of grammatical *surface structures* in a language—it must also be required to generate the class of correct *structural descriptions* of these surface structures. But, in fact, it only became apparent that there were ‘fundamental underlying distinctions’ between the syntactic properties of the verbs ‘expect’ and ‘persuade’ when it was discovered that it was not always possible to exchange the verb ‘expect’ in a grammatical sentence with the verb ‘persuade’ and obtain another grammatical sentence. In other words, even a grammar that aimed merely to *weakly* generate the class of grammatical sentences in English would have had to distinguish between the syntactic properties of the verbs ‘expect’ and ‘persuade’ because the artificial language that it defined would have had to include sentence (3) and exclude sentence (4). Therefore, this example actually *supports* the view that if one’s goal is to produce a ‘correct’ theory for a language, it is sufficient to demand ‘merely’ that one’s grammar does not overgenerate and does not undergenerate.

I am willing to admit that it is *logically* possible for a grammar to weakly generate the sentences of a language but fail to strongly generate correct structural descriptions of these sentences. However I find the idea that such a situation could arise *in practice* highly implausible. In general, the more complex and highly structured the output of a device, the fewer the number of possible ways in which that device might work. And the set of acceptable sentences in a natural language form an extremely complex and highly structured output that must, in general, severely limit the class of possibly correct mechanisms by which humans produce and understand sentences in such a language.

In any case, given an acceptable utterance in some natural language, how is one to determine whether or not a given structural description is correct? In practice, this cannot be done other than by logically deducing from the structural description that if and only if it is correct, then native speakers will exhibit some specific behaviour. If a linguist had developed a grammar that after extensive testing had not been shown to overgenerate and had not been shown to undergenerate, he would be justifiably suspicious of any claims by native speakers that though certain sentences could be weakly generated by the grammar, the structural descriptions generated by the grammar for these sentences were incorrect. In my view, the linguist could be forgiven for

⁴³⁵ Chomsky 1965, 24.

⁴³⁶ Chomsky 1965, 24.

suspecting that these native speakers were either incorrectly interpreting the significance of the structural descriptions generated by the grammar or deliberately being awkward. In other words, by far the most important criterion as to whether or not a given grammar is an adequate explanation of a language is whether or not it *weakly* generates the language. Whether or not a grammar strongly generates a language is not something that can be rigorously tested.

I therefore think that Chomsky is incorrect in claiming, as a defence of his view that a linguistic theory must attempt to achieve explanatory adequacy, that ‘gross coverage of a large mass of data can often be attained by conflicting theories’ and that for ‘this reason it is not, in itself, an achievement of any particular theoretical interest or importance.’⁴³⁷ Whilst it may be true that a number of distinct and possibly conflicting theories can account for ‘a substantial and significant class of crucial cases,’⁴³⁸ I think it is safe to assume that the number of candidate grammars for some natural language that cannot be refuted after *extensive* testing for overgeneration and undergeneration will be very small indeed. Indeed, I see no reason why it should not be possible for some natural languages to uniquely determine a possibly correct grammar for that language simply by continuously testing candidate grammars for overgeneration and undergeneration until they are refuted. It must be remembered that for any given natural language there is an almost limitless supply of examples of ‘naturally occurring’ acceptable sentences that can be used to test any candidate grammar for undergeneration. Also, because it is fairly straightforward to implement a grammar as a computer program that generates random samples of strings from the artificial language defined by a grammar, it is very easy to test any grammar extensively for overgeneration.

18.6 *Johnson-Laird and Baroni claim that in general more than one grammar can account for a musical style*

Johnson-Laird points out with respect to his theory of jazz improvisation that ‘to understand how the mind functions, we need first a good account of what it is doing.’⁴³⁹ I take this to mean that, in his view, one should successfully characterize the set of acceptable jazz improvisations before one begins to speculate about the nature of the mental processes that putatively lead to the production of such improvisations. I agree with this view because it is in general much easier to explain why a small number of necessary and sufficient conditions seem to correctly characterize a class of phenomena than it is to attempt to go directly from the uncharacterized set of phenomena to a causal explanation.

Johnson-Laird proposes two completely different ‘general approaches to how the mind may generate [jazz] improvisations’⁴⁴⁰—one symbolic or ‘grammatical’ and the other distributed or subsymbolic. He points out that ‘unfortunately, it is extremely difficult to obtain evidence that is directly pertinent to theories of processing.’ A consequence of this is that it is hard to see how one could ‘show that the ‘grammatical’ account of performance is wrong’ since if ‘it gives an accurate account of the *output* [my emphasis] of the process—which, of course, is its principal aim—no examination

⁴³⁷ Chomsky 1965, 26.

⁴³⁸ Chomsky 1965, 24.

⁴³⁹ Johnson-Laird 1991, 292.

⁴⁴⁰ Johnson-Laird 1991, 292.

of corpora of improvisations can refute it.⁴⁴¹ It must be pointed out, however, that Johnson-Laird does not extensively test either of these models. Consequently, he fails to establish conclusively that either is actually a feasible model of jazz improvisation.

Like Johnson-Laird, Baroni claims that

more than one hypothesis can be proposed and various sets of rules can accurately describe the same events. In other words, it is possible that more than one grammar could describe the repertoire.⁴⁴²

However, although many different theories might be able to *approximate* a musical style, I think that the high level of complexity and structure that is typically possessed by acceptable pieces of tonal music militates against the view that in general it would be possible to produce more than one *correct* algorithmic style theory for the musical style of, say, the works in a particular genre by some specified composer. I therefore think that Baroni, like Chomsky, exaggerates the severity of this problem.

18.7 Camilleri claims that current computational models of musical tasks are implausible because they operate in a way that is obviously more complex than the way in which humans perform these tasks

I believe one must admit that a theory of human behaviour that posits hypotheses about mental processes cannot possibly be shown to be *correct* and thus I believe that one can require of such a theory only that it be *correct as far as we know*. This implies that, because so *little* is categorically known about the limitations of the human mind, almost *any* computational model that successfully simulated the observable behavioural output of human mental processes would have to be considered a *plausible* model of those processes, regardless of how complex or ad hoc its mode of operation.

Camilleri claims that

serious consideration of complexity equivalence (i.e. the equivalence between processes performed by a computer program and a human being) could force musical researchers to inspect and tighten up the psychological reality of their models. If we eschew these issues, we only mimic musical tasks, ignoring the reality of how they are performed by human beings.⁴⁴³

In fact, of course, no computer program has yet been produced that can perform any complex musical skill indistinguishably from a human exercising that same skill. Also, it seems to me that the output produced by humans when performing musical tasks such as pastiche composition and expressive performance may well be sufficiently complex and highly structured for it to be the case that the *only* computer models that *could* behave indistinguishably from humans performing these tasks would be those models that *actually worked* in a way that was in some sense ‘the same as’ human minds when performing these tasks.

In any case, it would always be impossible to determine whether or not a computer model that *behaved* indistinguishably from a human when performing some musical task actually ‘worked in the same way as’ a human when performing that task, because the mental processes involved in musical skills are directly accessible to neither

⁴⁴¹ Johnson-Laird 1991, 319.

⁴⁴² Baroni et al. 1992, 189.

⁴⁴³ Camilleri 1992, 182.

observation nor introspection and no empirically verifiable and refutable predictions can be logically deduced from hypotheses about the nature of such processes. Also, the fact that mental processes are inaccessible to introspection implies that we *do not know* how complex the ‘actual’ mental processes are that are involved in performing musical tasks. Therefore, Camilleri cannot possibly claim that existing computer simulations work in ways that are fundamentally different from and far more complex than the ways in which human minds work.

Thus whilst I am willing to admit that researchers might profitably use considerations of ‘psychological reality’ as a *guide* in their work, I do not think that modelling ‘psychological reality’ *directly* should—or indeed, can—be established as a *goal* of research in computational musicology. In other words, the consideration of ‘psychological reality’ may well prove a useful source of ideas in the development of computational models of musical skills, but, in my view, no importance should be attached to whether or not the action of a particular model seems to correspond to the ‘actual’ mental processes involved in exercising some musical skill (as these processes are dimly perceived by the introspective researcher). Thus, in my view, researchers in computational musicology should in fact be concentrating on producing accurate characterizations of musical *behaviour*—that is, computational simulations that successfully *mimic* humans performing musical tasks.

Part 2

19 Mathematical preliminaries

19.1 Introduction

In this chapter I shall introduce and define some mathematical concepts, terms and symbols which will be used later on. I have tried to make these definitions consistent with those given in Borowski and Borwein 1989. Any reader who does not have a background in mathematics or computing is strongly advised to familiarize himself or herself with the definitions given in this chapter before reading any subsequent chapters. As my usage of some terms and symbols is slightly idiosyncratic, a reader who has some knowledge of mathematics or computing is advised to skim through the remainder of this chapter, making note of any idiosyncrasies. He or she may then refer back to this chapter if confusion arises later on.

19.2 Sets, families and ordered sets

An object may function as a *universal set* if and only if it is a well-defined collection of objects that is defined to contain all and only possible objects that satisfy some specified set of criteria. For example, the set

$$\{1,2,3,4,5,6,7,8,9\}$$

is the universal set of natural numbers less than 10.

An object may be termed a *set* if and only if it is a collection of objects that are all distinct members of a single specified universal set. An object in a set is called a *member* or *element* of the set. When written out in full, sets are enclosed in braces and the members are delimited by commas. For example, given the universal set of letters in the roman alphabet, it is possible to speak of the set of letters in the word ‘abracadabra’ which would be as follows:

$$\{a,b,r,c,d\}$$

The order in which the elements are written in a set does not matter. For example, the set of letters in the word ‘abracadabra’ could be written in any of the following ways:

$$\{a,b,r,c,d\} \quad \{a,r,b,c,d\} \quad \{d,c,r,b,a\}$$

The number of members that a set contains is called the *cardinality* of the set and it is denoted by enclosing the set between vertical lines. For example, the cardinality of the set

$$A = \{a,b,c\}$$

is 3 and this fact is denoted as follows:

$$|A| = |\{a,b,c\}| = 3$$

The fact that a is an element of A is denoted as follows:

$$a \in A$$

And the fact that a , b and c are all elements of A is denoted as follows:

$$a,b,c \in A$$

A set A is a *superset* of another set B if and only if all members of B are also members of A . For example, given the two sets

$$A = \{a, b, c, d\} \quad B = \{a, b, c\}$$

then A is a superset of B and this fact is denoted as follows:

$$A \supseteq B$$

A set A is a *subset* of another set B if and only if all members of A are also members of B . For example, given the two sets

$$A = \{a, b, c\} \quad B = \{a, b, c, d\}$$

then A is a subset of B and this fact is denoted as follows:

$$A \subseteq B$$

A set A is *equal* to another set B if and only if all members of A are members of B and all members of B are members of A . For example, given the two sets

$$A = \{a, b, c, d\} \quad B = \{b, c, a, d\}$$

then A is equal to B and this fact is denoted as follows:

$$A = B$$

A set A is a *proper superset* of another set B if and only if A is a superset of B , A is not equal to B and B is not empty. For example, given the four sets

$$A = \{a, b, c, d\} \quad B = \{a, b, c\} \quad C = \{d, c, a, b\} \quad D = \{ \}$$

then A is a proper superset of B but it is not a proper superset of C and it is not a proper superset of D . These facts are denoted as follows:

$$A \supset B \quad C \not\subset A \quad D \not\subset A$$

Note that the empty set can be denoted using the symbol \emptyset thus:

$$\{ \} = \emptyset \quad |\{ \}| = |\emptyset| = 0$$

A set A is a *proper subset* of another set B if and only if A is a subset of B , A is not equal to B and A is not empty. For example, given the four sets

$$A = \{a, b, c, d\} \quad B = \{a, b, c\} \quad C = \{d, c, a, b\} \quad D = \{ \}$$

then B is a proper subset of A but neither C nor D are proper subsets of A . These facts are denoted as follows:

$$B \subset A \quad C \not\subset A \quad D \not\subset A$$

The *union* of two or more sets A, B, \dots is the set that contains all and only those objects that are members of at least one of sets A, B, \dots For example, given the three sets,

$$A = \{a, b, c\} \quad B = \{c, d, e\} \quad C = \{f\}$$

then the union of sets A, B and C is

$$\{a, b, c, d, e, f\}$$

and this fact would be denoted as follows:

$$A \cup B \cup C = \{a, b, c, d, e, f\}$$

The *intersection* of two or more sets A, B, \dots is the set that contains all and only those objects that are members of all of sets A, B, \dots . For example, given the three sets,

$$A = \{a, b, c\} \quad B = \{c, d, e\} \quad C = \{c, f\}$$

then the intersection of sets A, B and C is

$$\{c\}$$

and this fact would be denoted as follows:

$$A \cap B \cap C = \{c\}$$

The *relative complement* of a set A in another set B is the set that contains all and only members of B that are not members of A . In this thesis, the set B will almost always be a universal set that is a proper superset of the set A . For example, given two sets

$$A = \{a, b, c\} \quad B = \{b, c, d\}$$

then the relative complement of A in B is

$$\{d\}$$

This fact is denoted as follows:

$$B \setminus A = \{d\}$$

Given a set of sets,

$$A = \{a_1, a_2, a_3, a_4\}$$

then the following abbreviations can be used:

$$a_1 \cup a_2 \cup a_3 \cup a_4 = \bigcup_{i=1}^4 a_i = \bigcup_{i=1}^{|A|} a_i = \bigcup A = \bigcup_{a \in A} a$$

$$a_1 \cap a_2 \cap a_3 \cap a_4 = \bigcap_{i=1}^4 a_i = \bigcap_{i=1}^{|A|} a_i = \bigcap A = \bigcap_{a \in A} a$$

$\bigcup_{i=1}^{|A|} a_i$ is read ‘big union of a_i for i equals 1 to the cardinality of A ’ and $\bigcap_{i=1}^{|A|} a_i$ is read ‘big intersection of a_i for i equals 1 to the cardinality of A .’

An object is a *family* if and only if it is a collection of objects that are all members of a single specified universal set but that are not necessarily all distinct members of this specified universal set. An object in a family is called a *member* or *element* of the family. When written out in full, families are enclosed in round brackets and the members are delimited by commas. For example, given the universal set of letters in the roman alphabet, it is possible to speak of the family of letters in the word ‘abracadabra’ which would be as follows:

$$(a, b, r, a, c, a, d, a, b, r, a)$$

The order in which the elements are written in a family does not matter. For example, the family of letters in the word ‘abracadabra’ could be written in any of the following ways:

$$(a, b, r, a, c, a, d, a, b, r, a) \quad (a, r, b, a, d, a, c, a, r, b, a) \quad (a, a, a, a, a, b, b, c, d, r, r)$$

The number of members that a family contains is called the *cardinality* of the family and it is denoted by enclosing the family between vertical lines. For example, the cardinality of the family

$$A = (a, b, r, a, c, a, d, a, b, r, a)$$

is 11 and this fact is denoted as follows:

$$|A| = |(a, b, r, a, c, a, d, a, b, r, a)| = 11$$

The fact that a is an element of A is denoted as follows:

$$a \in A$$

And the fact that a , b and c are all elements of A is denoted as follows:

$$a, b, c \in A$$

An object is an *ordered set* if and only if it is a collection of objects that satisfies the following conditions:

1. Each member of the collection is a member of a specified universal set.
2. The members are in a specified order.

Note that it is not necessary for any two members of an ordered set to be members of the *same* universal set. Note also that an ordered set may in general contain two or more members that are each the same member of a single specified universal set.

An object in an ordered set is called a *member* or *element* of the ordered set. When written out in full, ordered sets are enclosed in angle brackets and the members are delimited by commas. For example, the ordered set of letters in the word ‘abracadabra’ in which the members are arranged in the order in which they appear in the word would be:

$$\langle a, b, r, a, c, a, d, a, b, r, a \rangle$$

The order in which the elements are written in an ordered set *does* matter. For example, the following three ordered sets are all distinct from each other:

$$\langle a, b, r, a, c, a, d, a, b, r, a \rangle \quad \langle a, r, b, a, d, a, c, a, r, b, a \rangle \quad \langle a, a, a, a, a, b, b, c, d, r, r \rangle$$

An object is an *ordered pair* if and only if it is an ordered set containing exactly two members. An object is an *n-tuple* if and only if it is an ordered set containing exactly n members.

19.3 Logical propositions and connectives

The *universal set of truth-values* is defined to be the set

$$\{True, False\}$$

An object can be termed a *logical proposition* if and only if it is an expression that has a single *truth-value* that is a member of the universal set of truth-values. Given a logical proposition A then the truth-value of A is denoted

$$\text{truth}(A)$$

Throughout the remainder of this chapter the letters T and F will be used as abbreviations for the truth-values *True* and *False* respectively. So, for example, one can say that for any logical proposition A ,

$$\text{truth}(A) \in \{T, F\}$$

\wedge is the logical connective ‘and.’ Given two logical propositions A and B , the complex proposition

$$A \wedge B$$

should be read ‘ A and B .’ Given the truth-values of A and B , it is possible to uniquely determine the truth-value of the complex proposition

$$A \wedge B$$

As will be seen below, this is not true for all logical connectives as used here. The truth-value of

$$A \wedge B$$

can be found from the truth-values of A and B by using the following table called a *truth-table*:

$\text{truth}(A)$	$\text{truth}(B)$	$\text{truth}(A \wedge B)$
T	T	T
T	F	F
F	T	F
F	F	F

\vee is the logical connective ‘or.’ Given two propositions A and B , then the complex proposition

$$A \vee B$$

should be read ‘ A or B .’ The truth-table for this connective is as follows:

$\text{truth}(A)$	$\text{truth}(B)$	$\text{truth}(A \vee B)$
T	T	T
T	F	T
F	T	T
F	F	F

$\underline{\vee}$ is the logical connective ‘xor’ (pronounced ‘exor’). Given two logical propositions A and B ,

$$A \underline{\vee} B$$

should be read ‘ A xor B .’ The truth-table for this connective is as follows:

$\text{truth}(A)$	$\text{truth}(B)$	$\text{truth}(A \vee B)$
T	T	F
T	F	T
F	T	T
F	F	F

The symbol ' \Rightarrow ' will be used throughout the remainder of this thesis for *strict implication* and the symbol ' \rightarrow ' will be used for *material implication*. Given two logical propositions A and B , the complex proposition

$$A \Rightarrow B$$

should be read ' A implies B ' or 'if A then B ' and the complex proposition

$$A \rightarrow B$$

should be read ' A materially implies B .' Material implication will not be used in this thesis but the truth-table for this connective is as follows:

$\text{truth}(A)$	$\text{truth}(B)$	$\text{truth}(A \rightarrow B)$
T	T	T
T	F	F
F	T	T
F	F	T

Unlike material implication and the other logical connectives defined above, strict implication is not truth-functional. That is, given two logical propositions A and B and their truth-values $\text{truth}(A)$ and $\text{truth}(B)$, it is not possible to determine $\text{truth}(A \Rightarrow B)$ from a truth-table. Instead, the truth-value of $A \Rightarrow B$ is defined to be T if and only if B is validly deducible from A and F otherwise. Alternatively, in modal-logical terms one can say that $A \Rightarrow B$ is true if and only if it is not possible (in any possible world) for A to be true and B to be false.⁴⁴⁴

The symbol ' \Leftrightarrow ' will be used throughout the remainder of this thesis for *strict equivalence* and the symbol ' \leftrightarrow ' will be used for *material equivalence*. Given two logical propositions A and B the complex proposition

$$A \Leftrightarrow B$$

should be read ' A implies and is implied by B ' or 'if and only if A then B ' and the complex proposition

$$A \leftrightarrow B$$

should be read ' A is materially equivalent to B .' Material equivalence will not be used in this thesis but the truth-table for this connective is as follows:

⁴⁴⁴ See Borowski and Borwein 1989, pages 364-5, 461, 565 and 606-7, for clarification on issues of material and strict implication.

$\text{truth}(A)$	$\text{truth}(B)$	$\text{truth}(A \leftrightarrow B)$
T	T	T
T	F	F
F	T	F
F	F	T

Like strict implication, strict equivalence is not truth-functional. That is, given two logical propositions A and B and their truth-values $\text{truth}(A)$ and $\text{truth}(B)$, it is not possible to determine $\text{truth}(A \leftrightarrow B)$ from a truth-table. Instead, $\text{truth}(A \leftrightarrow B)$ is defined to be T if and only if $A \Rightarrow B$ and $B \Rightarrow A$ and it is defined to be F otherwise.⁴⁴⁵

The symbol \neg denotes the monadic truth-functional operator *negation*. Given a logical proposition A then $\neg A$ should be read ‘not A ’ and the value of $\text{truth}(\neg A)$ can be uniquely determined from the value of $\text{truth}(A)$ using the following truth-table:

$\text{truth}(A)$	$\text{truth}(\neg A)$
T	F
F	T

19.4 Some arithmetical abbreviations and functions

The symbol $=_{\text{df}}$ is the symbol for ‘equality by definition.’ That is, given two quantities A and B then the expression

$$A =_{\text{df}} B$$

should be read ‘ A is defined to be equal to B .’ A in this expression is the quantity that is being defined and it is called the *definiendum*. B is an expression whose terms are already well-defined and it is called the *definiens*. The expression as a whole is a *definition*.

Given an ordered set of numbers,

$$A = \langle x_1, x_2, \dots, x_{|A|} \rangle$$

then the following abbreviations can be used:

$$x_1 + x_2 + \dots + x_{|A|} = \sum_{i=1}^{|A|} x_i$$

$$x_1 \times x_2 \times \dots \times x_{|A|} = \prod_{i=1}^{|A|} x_i$$

⁴⁴⁵ The notion of ‘equivalence’ defined in Borowski and Borwein 1989, page 196, is the same as my notion of ‘material equivalence’. They do not define any concept that is the same as what I call ‘strict equivalence’.

The expression $\sum_{i=1}^{|A|} x_i$ is read ‘the sum of a_i for i equals 1 to the cardinality of A ’ and the expression $\prod_{i=1}^{|A|} x_i$ is read ‘the product of a_i for i equals 1 to the cardinality of A .’

The function $\text{abs}(x)$ takes any real number as argument and is defined as follows:

$$\text{abs}(x) = \begin{cases} x, & \text{if } x \geq 0 \\ -x, & \text{if } x < 0 \end{cases}$$

The function $\text{signum}(x)$ takes any real number as argument and is defined as follows:

$$\text{signum}(x) = \begin{cases} 0 & \text{if } x = 0 \\ \frac{x}{\text{abs}(x)} & \text{if } x \neq 0 \end{cases}$$

For example, $\text{signum}(-3.7) = -1$ and $\text{signum}(2\frac{1}{2}) = 1$.

The function $\text{int}(x)$ takes any real number as argument and returns the largest integer less than or equal to the argument. For example, $\text{int}(-3.4) = -4$, $\text{int}(\frac{1}{2}) = 0$ and $\text{int}(5.95) = 5$.

The binary operation ‘mod’ is defined as follows:

$$x \bmod y = x - y \times \text{int}\left(\frac{x}{y}\right)$$

where x is an integer and y is a natural number.⁴⁴⁶ For example, $-7 \bmod 5 = 3$, $-7 \bmod 7 = 0$, $7 \bmod 5 = 2$.

The binary operation ‘div’ is defined as follows:

$$x \text{ div } y = \text{int}\left(\frac{x}{y}\right)$$

where x is an integer and y is a non-zero integer. For example, $-7 \text{ div } -5 = 1$, $-7 \text{ div } 2 = -4$, $7 \text{ div } -5 = -2$ and $7 \text{ div } 2 = 3$.

In discussing metric structure I shall need to use the function $\text{rev}(w, z)$ that takes two arguments, of which the first must be a natural number and the second must be an integer. Also, the arguments must satisfy the following inequality:

$$w \geq \text{int}(\log_2(\text{abs}(z))) + 1$$

The function is defined as follows:

$$\text{rev}(w, z) = \text{signum}(z) \sum_{n=1}^w \left(2^{n-1} \left(\left(\dots \left((\text{abs}(z) \bmod 2^{w-1}) \bmod 2^{w-2} \right) \dots \right) \bmod 2^{w-(n-1)} \right) \text{div } 2^{w-n} \right)$$

⁴⁴⁶ The reader should note that the natural numbers are taken throughout this thesis to be the set of counting numbers—that is, the set of positive integers (*excluding* zero).

I call this function ‘binary reverse’ because it is equivalent to the following algorithm:

1. Find $z' = \text{abs}(z)$. For example, if $z = -83$ then $z' = 83$.
2. Find the binary representation of z' . For example, if z' were 83 then the required binary number would be 1010011.
3. The inequality $w \geq \text{int}(\log_2(\text{abs}(z))) + 1$ is equivalent to the stipulation that w must be no less than the number of digits in the binary representation of z' . The third step in the algorithm is to choose a natural number value for w that satisfies the inequality $w \geq \text{int}(\log_2(\text{abs}(z))) + 1$ and then to produce the string of zeros and ones that is equal to the binary representation of z' obtained in the second step padded to the left with zeros so that the final string has w digits. For example, if $z' = 83_{10} = 1010011_2$ then w must be a natural number greater than or equal to 7. If w were 7 then this third step of the algorithm would produce the string 1010011; if w were 10 then the string produced would be 0001010011.
4. The fourth stage of the algorithm is to take the string produced at the end of the third stage and *reverse* it. For example, if z' were 83 then if w were 7 the string produced at the end of this stage of the algorithm would be 1100101, but if w were 10 then the string produced would be 1100101000.
5. The fifth stage is to find the decimal equivalent of the binary number represented by the string of zeros and ones produced at the end of the fourth stage. For example, if z' were 83 then if w were 7 the output of the fifth stage would be $1100101_2 = 101_{10}$, but if w were 10 then the output from this stage would be $1100101000_2 = 808_{10}$.
6. The final step is to multiply the number produced at the end of the fifth stage by the value $\text{signum}(z)$ to give the final output of the algorithm. For example, if w were 10 then if z were 83 the final output would be 808 but if z were -83 then the final output would be -808.

The function $\max(A)$ takes as its single argument a collection of numbers and returns the least value that is greater than or equal to every member of the argument collection. Similarly, the function $\min(A)$ takes as its single argument a collection of numbers and returns the greatest value that is less than or equal to every member of the argument collection.

The function $\text{lpf}(n)$ takes a natural number as its single argument and returns the least prime factor of n greater than 1 if n is greater than 1 and 1 otherwise.

19.5 Rational numbers

An object is defined to be a *rational number* in the context of this thesis if and only if it is an ordered pair of numbers of which the first is an integer and the second is a natural number. A rational number r will be denoted as follows:

$$r = \langle \nu(r), \delta(r) \rangle$$

where $\nu(r)$ is an integer called the *numerator* and $\delta(r)$ is a natural number called the *denominator*. Rational numbers can, of course, be written in ‘vulgar fraction form’ so that

$$r = \langle v(r), \delta(r) \rangle = \frac{v(r)}{\delta(r)}$$

In this thesis I shall usually use this vulgar fraction form. However, in discussing metric structure it proves more useful to adopt the ordered pair form.

Given two rational numbers,

$$r_1 = \langle v_1, \delta_1 \rangle \quad r_2 = \langle v_2, \delta_2 \rangle$$

then

$$r_1 = r_2 \Leftrightarrow v_1 \delta_2 = v_2 \delta_1$$

is defined to be true. For example,

$$\langle 1, 2 \rangle = \langle 3, 6 \rangle = \langle 42, 84 \rangle$$

where the three ordered pairs are understood to be rational numbers. The set that contains all and only rational numbers—the *universal set of rational numbers*—can therefore be partitioned exclusively and exhaustively into equivalence classes such that two rational numbers r_1 and r_2 are in the same equivalence class if and only if $r_1 = r_2$. Each of these equivalence classes contains one member whose denominator is less than the denominator of any other member of the equivalence class. This member of the equivalence class is called the *least denominator form* of every other member of the class. Thus, given a rational number r_1 , the least denominator form of r_1 , denoted $\text{ldf}(r_1)$, is that rational number r_2 such that $r_1 = r_2$ and $\delta(r_2)$ is a minimum. The denominator of the least denominator form of a rational number r is called the *least denominator* of r and it is denoted

$$\delta_{\min}(r) = \delta(\text{ldf}(r))$$

Rational numbers can be combined with each other and with integers, natural numbers and real numbers under the operations of addition, subtraction, multiplication and division. When a rational number is combined with an integer or a natural number, the result is a rational number. When a rational number is combined with a real number, the result is a real number.

Given the rational numbers,

$$r_1 = \langle v_1, \delta_1 \rangle \quad r_2 = \langle v_2, \delta_2 \rangle$$

and given also the natural number n the integer z and the real number x then given below are the results of combining rational numbers with natural, integer and real numbers under the four arithmetic operations of addition, subtraction, multiplication and division.

Multiplication

$$r_1 r_2 = \langle v_1 v_2, \delta_1 \delta_2 \rangle \quad (\text{Rational})$$

$$r_1 n = \langle v_1 n, \delta_1 \rangle \quad (\text{Rational})$$

$$r_1 z = \langle v_1 z, \delta_1 \rangle \quad (\text{Rational})$$

$$r_1 x = \frac{v_1 x}{\delta_1} \quad (\text{Real})$$

Addition

$$r_1 + r_2 = \langle v_1\delta_2 + v_2\delta_1, \delta_1\delta_2 \rangle \quad (\text{Rational})$$

$$r_1 + n = \langle v_1 + n\delta_1, \delta_1 \rangle \quad (\text{Rational})$$

$$r_1 + z = \langle v_1 + z\delta_1, \delta_1 \rangle \quad (\text{Rational})$$

$$r_1 + x = \frac{v_1 + x\delta_1}{\delta_1} \quad (\text{Real})$$

Subtraction

$$r_1 - r_2 = \langle \delta_2v_1 - \delta_1v_2, \delta_1\delta_2 \rangle \quad (\text{Rational})$$

$$n - r_1 = \langle \delta_1n - v_1, \delta_1 \rangle \quad (\text{Rational})$$

$$r_1 - n = \langle v_1 - \delta_1n, \delta_1 \rangle \quad (\text{Rational})$$

$$z - r_1 = \langle \delta_1z - v_1, \delta_1 \rangle \quad (\text{Rational})$$

$$r_1 - z = \langle v_1 - \delta_1z, \delta_1 \rangle \quad (\text{Rational})$$

$$x - r_1 = \frac{x\delta_1 - v_1}{\delta_1} \quad (\text{Real})$$

$$r_1 - x = \frac{v_1 - x\delta_1}{\delta_1} \quad (\text{Real})$$

Division

$$\frac{r_1}{r_2} = \langle \text{signum}(v_2)v_1\delta_2, \text{abs}(v_2)\delta_1 \rangle \quad \text{given } v_2 \neq 0 \quad (\text{Rational})$$

$$\frac{r_1}{n} = \langle v_1, n\delta_1 \rangle \quad (\text{Rational})$$

$$\frac{n}{r_1} = \langle \text{signum}(v_1)n\delta_1, \text{abs}(v_1) \rangle \quad \text{given } v_1 \neq 0 \quad (\text{Rational})$$

$$\frac{r_1}{z} = \langle \text{signum}(z)v_1, \text{abs}(z)\delta_1 \rangle \quad \text{given } z \neq 0 \quad (\text{Rational})$$

$$\frac{z}{r_1} = \langle \text{signum}(v_1)z\delta_1, \text{abs}(v_1) \rangle \quad \text{given } v_1 \neq 0 \quad (\text{Rational})$$

$$\frac{r_1}{x} = \frac{v_1}{\delta_1x} \quad \text{given } x \neq 0 \quad (\text{Real})$$

$$\frac{x}{r_1} = \frac{d_1x}{n_1} \quad \text{given } n_1 \neq 0 \quad (\text{Real})$$

20 Pitch, chroma, morph and genus

The term ‘pitch’ is generally and correctly used to signify a psychoacoustical property of a musical tone. It is that perceptual attribute of a musical tone that corresponds to the perceptual attribute of an audible simple tone that changes when the frequency of the simple tone is varied, keeping all other physical characteristics of the simple tone constant.⁴⁴⁷

In the remainder of this thesis, however, the term ‘pitch’ will in general *not* be used in this sense. Instead, it will be used to signify an abstract mathematical object that can be derived algorithmically from a written note in a physical Standard Notation score. An object p is defined to be a *pitch* if and only if it is an ordered pair of integers

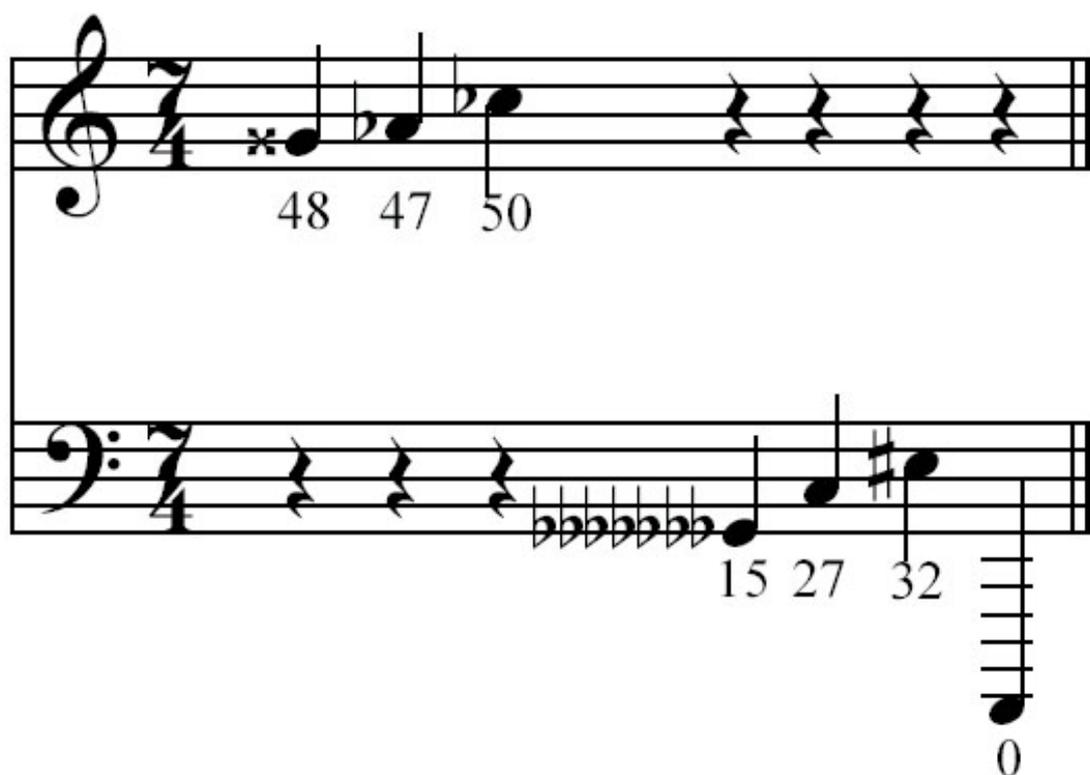


Figure 20-1

$$p = \langle p_c(p), p_m(p) \rangle$$

in which the first element is the *chromatic pitch* of p and the second element is the *morphetic pitch* of p .

The chromatic pitch of a pitch corresponds to the key on a keyboard instrument associated with the pitch, the keys being numbered in succession, starting at 0 for the lowest A natural on a normal piano keyboard, 1 for the B flat above this key and so on. For example, the chromatic pitch of the pitch of a note representing middle C on a piano keyboard is 39. Figure 20-1 shows some notes and their chromatic pitches.

⁴⁴⁷ See Moore 1989, 189.

The morphetic pitch of a pitch corresponds to the vertical position on the staff of the note-head of the note associated with the pitch in question. The morphetic pitch of a pitch is therefore not directly dependent upon pitch height—it is dependent only on the position of a note-head on a staff. The morphetic pitch of the note that is used to indicate the lowest A natural on a piano keyboard on a non-transposed staff is defined to be zero. The morphetic pitch of the B written on the line above this A is 1, the C above that has morphetic pitch 2 and so on. The morphetic pitch of middle C is therefore 23. Figure 20-2 shows some notes and their morphetic pitches.

The *universal set of pitches* is the set that contains all and only pitches. It is an infinite set and it is defined and denoted as follows:

$$\underline{p}^u = \{ \langle p^c, p^m \rangle : p^c, p^m \in \mathbb{Z} \}$$

where \mathbb{Z} is the universal set of integers—that is,

$$\mathbb{Z} = \{ \dots, -3, -2, -1, 0, 1, 2, 3, \dots \}$$

(see Borowski and Borwein 1989, 298). An object \underline{p} is a *pitch set* if and only if it is a subset of the universal set of pitches.

Given two pitches

$$p_1 = \langle p_1^c, p_1^m \rangle \quad p_2 = \langle p_2^c, p_2^m \rangle$$

then p_1 is defined to be *greater than* p_2 (denoted $p_1 > p_2$) if and only if one of the following two conditions is satisfied:

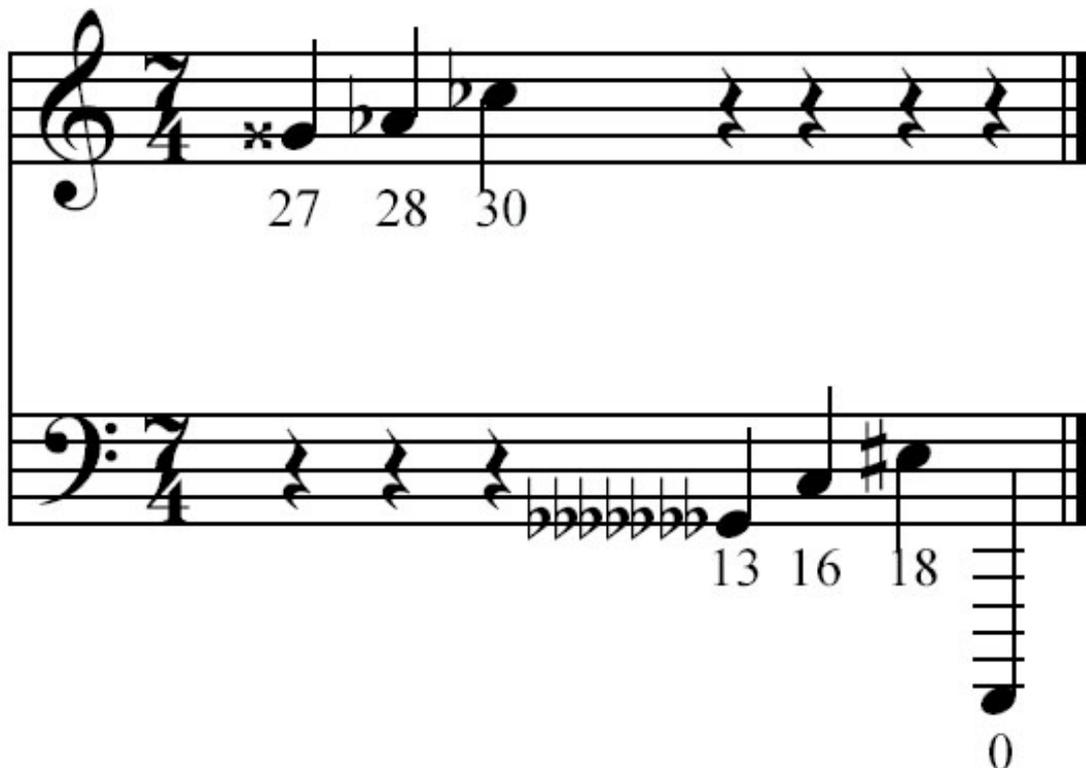


Figure 20-2

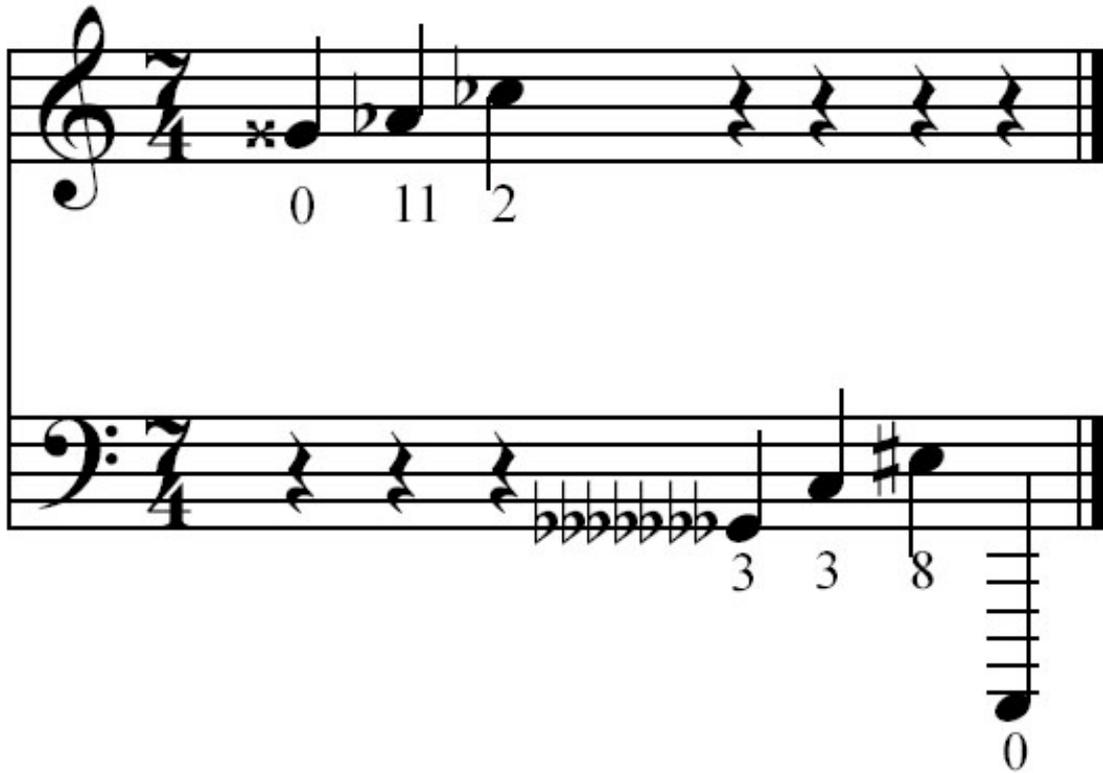


Figure 20-3

1. $p_1^c > p_2^c$
2. $(p_1^c = p_2^c) \wedge (p_1^m > p_2^m)$

A pitch is *less than* another pitch if and only if it is neither equal to it nor greater than it.

Given a pitch $p = \langle p^c, p^m \rangle$, then the *chroma* of p is defined and denoted as follows:

$$c(p) = p^c \bmod 12$$

Figure 20-3 shows some notes and their corresponding chromae. The chroma of a pitch corresponds to the music-theoretical notion of ‘pitch class’ as used, for example, in Forte 1973. In atonal theory a pitch class of zero corresponds to the class of C naturals. Here, however, a chroma of zero corresponds to the class of A naturals.⁴⁴⁸

The *universal set of chromae* is the set that contains all and only chromae. It is defined and denoted as follows:

$$c^u = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$$

An object \underline{c} is a *chroma set* if and only if it is a subset of the universal set of chromae.

Given a pitch $p = \langle p^c, p^m \rangle$, then the *morph* of p is defined and denoted as follows:

⁴⁴⁸ See Deutsch 1982, page 272, for an example of the use of the term *chroma* that is essentially the same as my own.

$$m(p) = p^m \bmod 7$$

Figure 20-4 shows some notes and their corresponding morphs. The morph of a pitch corresponds to the letter name of the pitch with 0 corresponding to A, 1 corresponding to B and so on up to 6 corresponding to G.

The *universal set of morphs* is the set that contains all and only morphs. It is defined and denoted as follows:

$$\underline{m}^u = \{0,1,2,3,4,5,6\}$$

An object \underline{m} is a *morph set* if and only if it is a subset of the universal set of morphs.

Given a pitch $p = \langle p^c, p^m \rangle$, then the *chromatic octave* of p is defined and denoted as follows:

$$o_c(p) = p^c \text{ div } 12$$

and the *morphic octave* of p is defined and denoted as follows:

$$o_m(p) = p^m \text{ div } 7$$

Figure 20-5 shows some notes with their corresponding chromatic octaves and Figure 20-6 shows some notes with their corresponding morphetic octaves. The chromatic and morphetic octaves of a pitch indicate the octave of the pitch. The chromatic octave of a pitch is not necessarily equal to its morphetic octave. For example, compare the first and second notes in Figure 20-5 with the first and second notes in Figure 20-6.

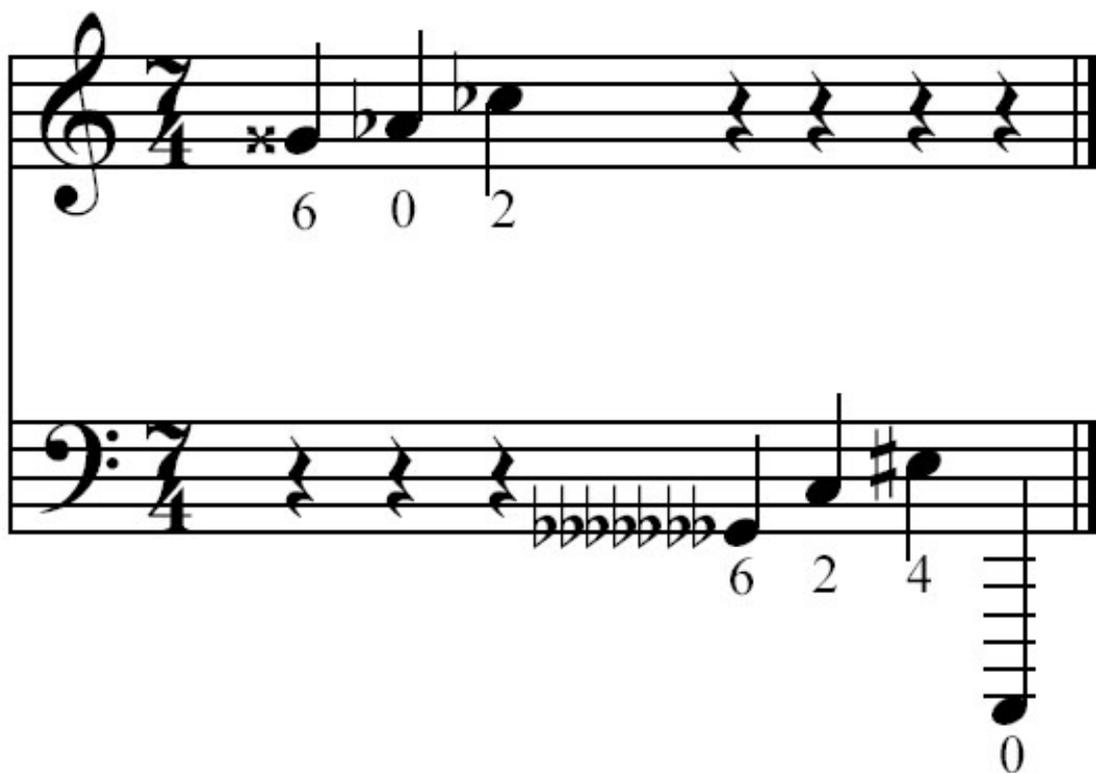


Figure 20-4

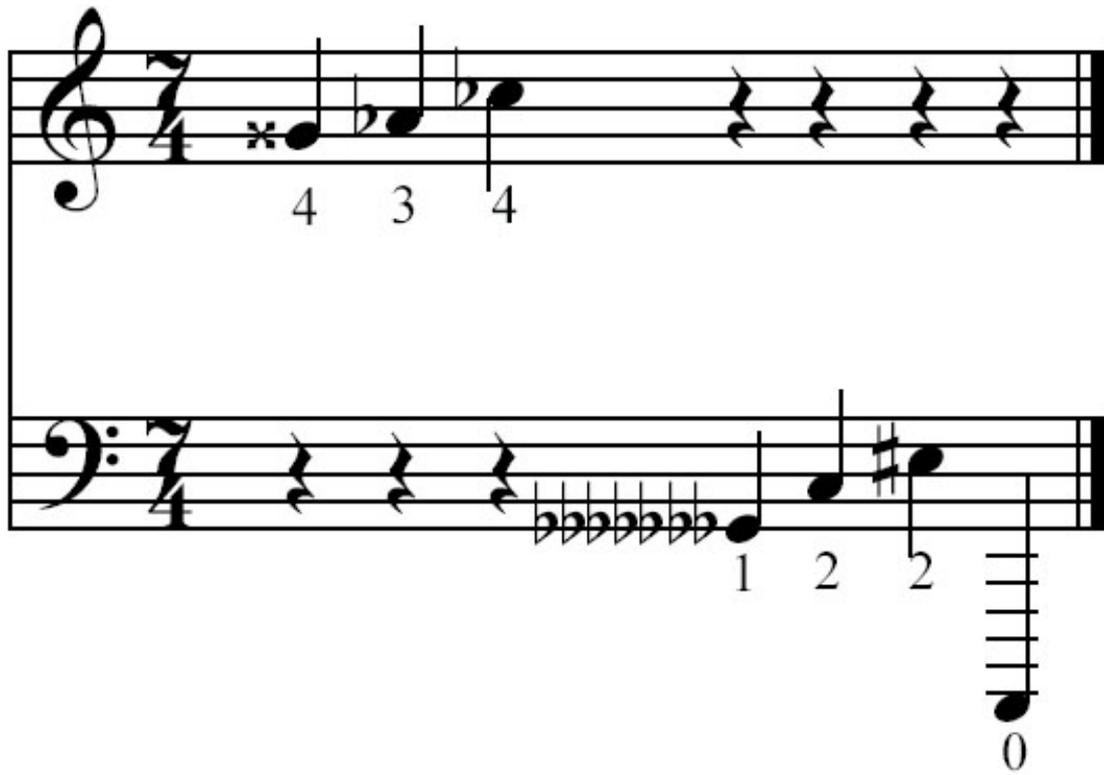


Figure 20-5

Given a pitch $p = \langle p^c, p^m \rangle$, then the *displacement* of p is defined and denoted as follows:

$$e(p) = p^c - p'_c(p)$$



Figure 20-7

That is, the genus of a pitch is the ordered pair in which the first element in the pair is the morph of the pitch and the second element is the displacement of the pitch. The genus of a pitch identifies the diatonic pitch name of the pitch without specifying its octave. Figure 20-8 shows some notes with their corresponding genera. Given a genus

$$q_1 = \langle m_1, e_1 \rangle$$

then the function $m(q_1)$ returns the morph of q_1 ,

$$m(q_1) = m_1$$

the function $e(q_1)$ returns the displacement of q_1 ,

$$e(q_1) = e_1$$

and the function $c(q_1)$ returns the chroma of q_1 ,

$$c(q_1) = (7((2((m_1 + 2) \bmod 7)) \bmod 7) + 8 + e_1) \bmod 12$$

The *universal set of genera* is the set that contains all and only genera. It is an infinite set and it is defined and denoted as follows:

$$\underline{q}^u = \{ \langle m, e \rangle : m \in \underline{m}^u, e \in \mathbb{Z} \}$$

An object \underline{q} is a *genus set* if and only if it is a subset of the universal set of genera.

Given two genera,

$$q_1 = \langle m_1, e_1 \rangle \quad q_2 = \langle m_2, e_2 \rangle$$

then q_1 is defined to be *greater than* q_2 (denoted $q_1 > q_2$) if and only if one of the following two conditions is satisfied:

1. $m_1 > m_2$
2. $(m_1 = m_2) \wedge (e_1 > e_2)$

A genus is *less than* another genus if and only if it is neither equal to it nor greater than it.

The figure displays two musical staves in 7/4 time. The top staff is in treble clef and contains a sequence of notes with vector labels: $\langle 6, 2 \rangle$ (marked with an asterisk), $\langle 2, -1 \rangle$, and $\langle 0, -1 \rangle$. The bottom staff is in bass clef and contains a sequence of notes with vector labels: $\langle 6, -7 \rangle$, $\langle 4, 1 \rangle$, and $\langle 2, 0 \rangle$. Below the bottom staff, there is a vertical staff with a single note labeled $\langle 0, 0 \rangle$.

Figure 20-8

capital letter part of the note in Figure 21-2 is defined to be C, then the capital letter parts of other notes can be found by comparison with this standard in the usual fashion.

The numerical subscript of the A.S.A. pitch name of a note depends on the capital letter part in that it must be the same as the numerical subscript of the closest C below it. The numerical subscript of the note in Figure 21-2 is defined to be one. The closest C flat above the note in Figure 21-2 would have a numerical subscript of 2 whereas the closest B sharp above would have a numerical subscript of 1. This example shows that the numerical subscript is not directly dependent upon pitch height. The pitch name numerical subscript of a note is determined by the vertical position of the note-head on the staff. The accidental of a note's A.S.A. pitch name is determined by explicit

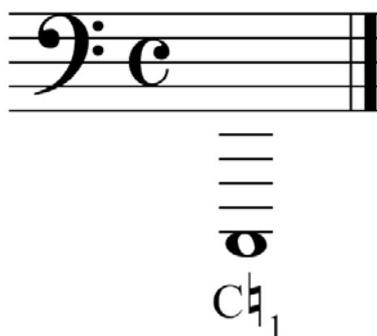


Figure 21-2 $C\sharp_1$ in the A.S.A. system

symbols placed before the note and either in the same bar as the note or in the key signature that operates in that bar.

Before defining how the pitch of a note is derived from its A.S.A. pitch name, I shall take this opportunity to define two more concepts—that of a *genus name* and that of a *pitch name*. An object is a *genus name* if and only if it is an ordered pair in which the first member of the pair is a member of the set $\{A,B,C,D,E,F,G\}$ and the second member of the pair is a member of the set $\{\dots\flat\flat, \flat, \sharp, \times, \sharp, \dots\}$. Given the A.S.A. pitch name of a note, then the genus name of the note is the ordered pair in which the first member is the capital letter part of the A.S.A. pitch name and the second member of the pair is the accidental of the pitch name. The following is a list of the genus names of the notes in Figure 21-1 and demonstrates the notation that will be used:

$$\begin{array}{ll} \text{gn}(G\times_4) = \langle G, \times \rangle & \text{gn}(A\flat_4) = \langle A, \flat \rangle \\ \text{gn}(C\flat_5) = \langle C, \flat \rangle & \text{gn}(G\flat\flat\flat\flat\flat_2) = \langle G, \flat\flat\flat\flat\flat \rangle \\ \text{gn}(C\sharp_3) = \langle C, \sharp \rangle & \text{gn}(E\sharp_3) = \langle E, \sharp \rangle \end{array}$$

The function $\text{gn}(x)$ can take as argument a genus, a pitch, a pitch name (see below), an A.S.A. pitch name or a note and returns the unique genus name associated with the argument.

An object is a *pitch name* (as opposed to an A.S.A. pitch name) if and only if it is an ordered pair in which the first member is a genus name and the second member is an integer. Given the A.S.A. pitch name of a note, then the pitch name of the note is the ordered pair in which the first member is the genus name of the note and the second

member is the numerical subscript of the A.S.A. pitch name. The following is a list of the pitch names of the notes in Figure 21-1 and shows the notation that will be used:

$$\begin{aligned} \text{pn}(G\sharp_4) &= \langle\langle G, \sharp \rangle, 4\rangle & \text{pn}(A\flat_4) &= \langle\langle A, \flat \rangle, 4\rangle \\ \text{pn}(C\flat_5) &= \langle\langle C, \flat \rangle, 5\rangle & \text{pn}(G\text{b}^{\text{b}^{\text{b}^{\text{b}^{\text{b}^{\text{b}}}}}}_2) &= \langle\langle G, \text{b}^{\text{b}^{\text{b}^{\text{b}^{\text{b}^{\text{b}}}}}} \rangle, 2\rangle \\ \text{pn}(C\sharp_3) &= \langle\langle C, \sharp \rangle, 3\rangle & \text{pn}(E\sharp_3) &= \langle\langle E, \sharp \rangle, 3\rangle \end{aligned}$$

The function $\text{pn}(x)$ can take as argument a pitch, an A.S.A pitch name or a note and returns the unique pitch name associated with the argument.

Having found the pitch name of a note, it is then possible to find the pitch of the note. In order to find the pitch of a note, it is necessary to find its chromatic and morphetic pitches. Given a note whose pitch name is

$$n = \langle\langle l, a \rangle, s \rangle$$

then the pitch of this note

$$p = \langle p_c(p), p_m(p) \rangle$$

can be found using the following algorithm:

1. Find $m(p)$, the morph of p , from l , the capital letter part of n , using the following table which shows the value of $m(p)$ for any value of l :

l	A	B	C	D	E	F	G
$m(p)$	0	1	2	3	4	5	6

2. Find $e(p)$, the displacement of p , from a , the accidental of n , using the following table which shows the value of $e(p)$ for each value of a :

a	...	bb	\flat	\flat	\sharp	\times	$\#\times$...	
$e(p)$...	-3	-2	-1	0	1	2	3	...

3. Find $o_m(p)$ from s and l using the following two rules:

$$l \in \{A, B\} \Rightarrow o_m(p) = s$$

$$l \in \{C, D, E, F, G\} \Rightarrow o_m(p) = s - 1$$

4. Find $p_m(p)$ which by definition is given by the equation

$$p_m(p) = 7 o_m(p) + m(p)$$

5. Find $c(p)$ from $m(p)$ and $e(p)$. This can be done using the fact that

$$c(p) = (7x + 8 + e(p)) \bmod 12$$

where x is the least non-negative integer for which

$$m(p) = (4x + 5) \bmod 7$$

This fact implies that if $m'(p)$ is defined as follows:

$$m'(p) = (m(p) + 2) \bmod 7$$

then

$$x = (2 m'(p)) \bmod 7$$

Therefore

$$c(p) = (7((2 m'(p)) \bmod 7) + 8 + e(p)) \bmod 12$$

and so

$$c(p) = \left(7 \left(\left(2 \left((m(p) + 2) \bmod 7 \right) \right) \bmod 7 \right) + 8 + e(p) \right) \bmod 12$$

The reader can confirm that this function gives the correct value of $c(p)$ for each value of $m(p)$ when $e(p) = 0$. These are given in the following table:

$m(p)$	0	1	2	3	4	5	6
$c(p)$	0	2	3	5	7	8	10

6. Find $o_c(p)$ from $c(p)$, $e(p)$ and $o_m(p)$ using the following equation:

$$o_c(p) = o_m(p) + \text{int} \left(\frac{\left((c(p) - e(p)) \bmod 12 \right) + e(p)}{12} \right)$$

7. Find $p_c(p)$ which by definition is given by the equation

$$p_c(p) = 12 o_c(p) + c(p)$$

8. Find $p(n) = p = \langle p_c(p), p_m(p) \rangle$ directly from the values of $p_c(p)$ and $p_m(p)$ found above.

This process of deriving the pitch of a note from its pitch name can also be reversed. That is, given a note whose pitch is

$$p = \langle p_c(p), p_m(p) \rangle$$

then the pitch-name of this note

$$n = \langle \langle l, a \rangle, s \rangle$$

can be found using the following algorithm:

1. Find $m(p)$ which by definition is given by the equation

$$m(p) = p_m(p) \bmod 7$$

2. Find l from $m(p)$ using the following table which shows the value of l for each value of $m(p)$:

$m(p)$	0	1	2	3	4	5	6
l	A	B	C	D	E	F	G

3. Find $o_m(p)$ which by definition is given by the equation

$$o_m(p) = p_m(p) \text{div } 7$$

4. Find s from $m(p)$ and $o_m(p)$ using the following two rules:

$$m(p) \in \{0, 1\} \Rightarrow s = o_m(p)$$

$$m(p) \in \{2, 3, 4, 5, 6\} \Rightarrow s = o_m(p) + 1$$

5. Find $e(p)$ which by definition is given by the following equation:

$$e(p) = p^c - p'_c(p)$$

where

$$p'_c(p) = 12 o_m(p) + c'(p)$$

and where

$$c'(p) = \left(7 \left(\left(2 \left((m(p) + 2) \bmod 7 \right) \right) \bmod 7 \right) + 8 \right) \bmod 12$$

6. Find a , the accidental of n , from $e(p)$ using the following table which shows the value of a for each value of $e(p)$:

$e(p)$...	-3	-2	-1	0	1	2	3	...
a	...	bb	b	b	q	#	x	##	...

7. Find $n = \langle \langle l, a \rangle, s \rangle$ directly from the values of l , a and s already found.

It is also possible to derive the genus of a note directly from its genus name. Given a note whose genus name is

$$g = \langle l, a \rangle$$

then the genus of this note

$$q = \langle m(q), e(q) \rangle$$

can be found using the following algorithm:

1. Find $m(q)$ from l using the following table which shows the value of $m(q)$ for each value of l :

l	A	B	C	D	E	F	G
$m(q)$	0	1	2	3	4	5	6

2. Find $e(q)$, from a , using the following table which shows the value of $e(q)$ for each value of a :

a	...	bb	b	b	q	#	x	##	...
$e(q)$...	-3	-2	-1	0	1	2	3	...

Given the genus of a note

$$q = \langle m(q), e(q) \rangle$$

then the genus name of the note

$$g = \langle l, a \rangle$$

can, of course, be found using the following algorithm:

1. Find l from $m(q)$ using the following table which shows the value of l for each value of $m(q)$:

$m(q)$	0	1	2	3	4	5	6
l	A	B	C	D	E	F	G

2. Find a from $e(q)$ using the following table which shows the value of a for each value of $e(q)$:

$e(q)$...	-3	-2	-1	0	1	2	3	...
a	...	bb	b	b	q	#	x	##	...

22 Pitch intervals

An object i is a *pitch interval* if and only if it is an ordered pair of integers

$$i = \langle i_c(i), i_m(i) \rangle$$

where the first element is the *chromatic interval* of i and the second element is the *morphic interval* of i .

Given two pitches,

$$p_1 = \langle p_1^c, p_1^m \rangle \quad p_2 = \langle p_2^c, p_2^m \rangle$$

then two pitch intervals can be defined in terms of them: that from p_1 to p_2 ; and that from p_2 to p_1 . The pitch interval from p_1 to p_2 is defined and denoted as follows:

$$i(p_1, p_2) = \langle p_2^c - p_1^c, p_2^m - p_1^m \rangle$$

and the pitch interval from p_2 to p_1 is defined and denoted as follows:

$$i(p_2, p_1) = \langle p_1^c - p_2^c, p_1^m - p_2^m \rangle$$

In the expression $i(p_1, p_2)$, p_1 is called the *object pitch* of the pitch interval and p_2 is

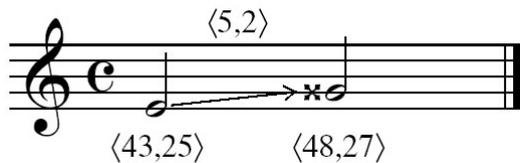


Figure 22-1

called the *image pitch* of the pitch interval. For example, the pitch interval from the E natural to the G double sharp in Figure 22-1 is

$$i(\langle 43, 25 \rangle, \langle 48, 27 \rangle) = \langle 5, 2 \rangle$$

which corresponds to a rising augmented third. The E natural is the object pitch and the G double sharp is the image pitch.

Pitch intervals can be directly related to the traditional terminology for musical intervals. For example, a rising major third is an increase of 4 in chromatic pitch and an increase of 2 in morphic pitch and is thus equivalent to the pitch interval $\langle 4, 2 \rangle$. Similarly, a falling perfect fifth corresponds to the pitch interval $\langle -7, -4 \rangle$, a rising major seventh corresponds to $\langle 11, 6 \rangle$, a rising diminished second corresponds to $\langle -1, 1 \rangle$ and so on.

The *universal set of pitch intervals* is the set that contains all and only pitch intervals. It is an infinite set and it is defined and denoted as follows:

$$\underline{i}^u = \{ \langle i^c, i^m \rangle : i^c, i^m \in \mathbb{Z} \}$$

An object is a *pitch interval set* if and only if it is a subset of the universal set of pitch intervals.

Given two pitch intervals

$$i_1 = \langle i_1^c, i_1^m \rangle \quad i_2 = \langle i_2^c, i_2^m \rangle$$

then i_1 is defined to be *greater than* i_2 (denoted $i_1 > i_2$) if and only if one of the following two conditions is satisfied:

1. $i_1^c > i_2^c$
2. $(i_1^c = i_2^c) \wedge (i_1^m > i_2^m)$

A pitch interval is *less than* another pitch interval if and only if it is neither equal to it nor greater than it.

Given a pitch interval, $i = i(p_1, p_2) = \langle i^c, i^m \rangle$ and given also the following defined quantities:

$$p_1 = \langle p_1^c, p_1^m \rangle$$

$$p_2 = \langle p_2^c, p_2^m \rangle$$

$$c_1 = p_1^c \bmod 12$$

$$c_2 = p_2^c \bmod 12$$

$$o_1^c = p_1^c \div 12$$

$$o_2^c = p_2^c \div 12$$

$$m_1 = p_1^m \bmod 7$$

$$m_2 = p_2^m \bmod 7$$

$$o_1^m = p_1^m \div 7$$

$$o_2^m = p_2^m \div 7$$

$$c_1' = (7((2((m_1 + 2) \bmod 7)) \bmod 7) + 8) \bmod 12$$

$$c_2' = (7((2((m_2 + 2) \bmod 7)) \bmod 7) + 8) \bmod 12$$

$$p_1^{c'} = 12o_1^m + c_1'$$

$$p_2^{c'} = 12o_2^m + c_2'$$

$$e_1 = p_1^c - p_1^{c'}$$

$$e_2 = p_2^c - p_2^{c'}$$

$$q_1 = \langle m_1, e_1 \rangle$$

$$q_2 = \langle m_2, e_2 \rangle$$

$$g_1 = \text{gn}(p_1) = \langle l_1, a_1 \rangle$$

$$g_2 = \text{gn}(p_2) = \langle l_2, a_2 \rangle$$

$$n_1 = \text{pn}(p_1) = \langle g_1, s_1 \rangle$$

$$n_2 = \text{pn}(p_2) = \langle g_2, s_2 \rangle$$

then it is useful to be able to express:

1. p_2 in terms of p_1 and i ;
2. m_2 in terms of m_1 and i ;
3. c_2 in terms of c_1 and i ;
4. q_2 in terms of q_1 and i ;
5. g_2 in terms of g_1 and i ; and
6. n_2 in terms of n_1 and i .

p_2 can be expressed in terms of p_1 and i as follows:

$$p_2 = \langle p_1^c + i^c, p_1^m + i^m \rangle$$

m_2 can be expressed in terms of m_1 and i as follows:

$$\begin{aligned}
m_2 &= p_2^m \bmod 7 \\
&= (p_1^m + i^m) \bmod 7 \\
&= (7o_1^m + m_1 + i^m) \bmod 7 \\
&= (m_1 + i^m) \bmod 7
\end{aligned}$$

c_2 can be expressed in terms of c_1 and i as follows:

$$\begin{aligned}
c_2 &= p_2^c \bmod 12 \\
&= (p_1^c + i^c) \bmod 12 \\
&= (12o_1^c + c_1 + i^c) \bmod 12 \\
&= (c_1 + i^c) \bmod 12
\end{aligned}$$

Finding q_2 given only q_1 and i involves finding m_2 and e_2 given only m_1 , e_1 , i^c and i^m . m_2 can be found from the equation $m_2 = (m_1 + i^m) \bmod 7$ derived above. e_2 can be found using an equation which can be derived as follows:

$$\begin{aligned}
e_2 &= p_2^c - p_2^{c'} \\
&= p_1^c + i^c - p_2^{c'} \\
&= p_1^c + i^c - 12o_2^m - c_2' \\
&= p_1^c + i^c - 12(p_2^m \text{ div } 7) - c_2' \\
&= p_1^c + i^c - 12((p_1^m + i^m) \text{ div } 7) - c_2' \\
&= p_1^c + i^c - 12((7o_1^m + m_1 + i^m) \text{ div } 7) - c_2' \\
&= p_1^c + i^c - 12(o_1^m + ((m_1 + i^m) \text{ div } 7)) - c_2'
\end{aligned}$$

But

$$\begin{aligned}
e_1 &= p_1^c - p_1^{c'} \\
\Rightarrow p_1^c &= e_1 + p_1^{c'} \\
&= e_1 + 12o_1^m + c_1'
\end{aligned}$$

Therefore,

$$\begin{aligned}
e_2 &= p_1^c + i^c - 12(o_1^m + ((m_1 + i^m) \text{ div } 7)) - c_2' \\
\Rightarrow e_2 &= e_1 + 12o_1^m + c_1' + i^c - 12o_1^m - 12((m_1 + i^m) \text{ div } 7) - c_2' \\
&= e_1 + c_1' + i^c - 12((m_1 + i^m) \text{ div } 7) - c_2'
\end{aligned}$$

where

$$c_1' = (7((2((m_1 + 2) \bmod 7)) \bmod 7) + 8) \bmod 12$$

as defined above and

$$\begin{aligned} c'_2 &= \left(7\left(\left(2\left(\left(m_2 + 2\right) \bmod 7\right)\right) \bmod 7\right) + 8\right) \bmod 12 \\ &= \left(7\left(\left(2\left(\left(\left(m_1 + i^m\right) \bmod 7\right) + 2\right) \bmod 7\right)\right) \bmod 7\right) + 8\right) \bmod 12 \end{aligned}$$

Finding g_2 (the genus name of p_2) given only g_1 (the genus name of p_1) and i can be achieved using the following algorithm:

1. Find q_1 using the algorithm defined in chapter 21 above for finding the genus of a pitch from its genus name.
2. Find q_2 from q_1 and i using the procedure derived earlier in this section.
3. Find g_2 from q_2 using the algorithm defined in chapter 21 above for finding the genus name of a pitch from its genus.

Finding n_2 (the pitch name of p_2) given only n_1 (the pitch name of p_1) and i involves finding g_2 and s_2 (the numerical subscript of the pitch name of p_2) using only g_1 , s_1 (the numerical subscript of the pitch name of p_1) and i . This can be achieved using the following algorithm:

1. Find p_1 from n_1 using the algorithm defined in chapter 21 above for finding the pitch of a note from its pitch name.
2. Find p_2 from p_1 and i using the equation $p_2 = \langle p_1^c + i^c, p_1^m + i^m \rangle$ given above.
3. Find n_2 from p_2 using the algorithm defined in chapter 21 above for finding the pitch name of a note from its pitch.

It is now possible to introduce the *transposition function* $\text{tran}(x,i)$ which takes two arguments of which the second must be a pitch interval and the first must be one of the following:

pitch	pitch set
chroma	chroma set
morph	morph set
genus	genus set

Given p_1 , c_1 , m_1 , q_1 , etc. as defined above then:

$$\text{tran}(p_1, i) =_{\text{df}} \langle p_1^c + i^c, p_1^m + i^m \rangle$$

$$\text{tran}(c_1, i) =_{\text{df}} (c_1 + i^c) \bmod 12$$

$$\text{tran}(m_1, i) =_{\text{df}} (m_1 + i^m) \bmod 7$$

$$\text{tran}(q_1, i) =_{\text{df}} \langle \text{tran}(m_1, i), e_2 \rangle$$

where e_2 is as defined above.

23 Pitch, chroma, morph and genus relations

An object R^p is a *pitch relation* if and only if it is an ordered pair

$$R^p = \langle \underline{i}(R^p), \underline{p}(R^p) \rangle$$

where the first element of the pair is a pitch interval set called the *pitch relation interval set* and the second element in the pair—the *pitch relation pitch set*—is the pitch set to which the pitch relation applies.

Pitch relations are used to construct logical propositions. Given two pitches p_1 and p_2 and a pitch relation $R^p = \langle \underline{i}(R^p), \underline{p}(R^p) \rangle$ then the expression

$$p_1 R^p p_2$$

is the logical proposition ‘Pitch p_1 is related to pitch p_2 by pitch relation R^p .’ This proposition is defined to be true if and only if both p_1 and p_2 are members of $\underline{p}(R^p)$ and the interval $i(p_1, p_2)$ is a member of $\underline{i}(R^p)$. That is:

$$\text{truth}(p_1 R^p p_2) =_{\text{df}} \text{truth}(\left(i(p_1, p_2) \in \underline{i}(R^p) \right) \wedge \left(p_1, p_2 \in \underline{p}(R^p) \right))$$

For example, given the pitches of a C major triad in root position on middle C: $p_1 = \langle 39, 23 \rangle$ (middle C), $p_2 = \langle 43, 25 \rangle$ (E, major third above middle C), $p_3 = \langle 46, 27 \rangle$ (G, perfect fifth above middle C), and given the pitch relation, $R^p = \langle \{ \langle 4, 2 \rangle, \langle 3, 2 \rangle \}, \underline{p}^u \rangle$ then

$$\text{truth}(p_1 R^p p_2) = \text{True}$$

$$\text{truth}(p_2 R^p p_3) = \text{True}$$

$$\text{truth}(p_2 R^p p_1) = \text{False}$$

$$\text{truth}(p_3 R^p p_2) = \text{False}$$

$$\text{truth}(p_1 R^p p_3) = \text{False}$$

$$\text{truth}(p_3 R^p p_1) = \text{False}$$

An object R^c is a *chroma relation* if and only if it is an ordered pair

$$R^c = \langle \underline{i}(R^c), \underline{c}(R^c) \rangle$$

where the first element of the pair is a pitch interval set called the *chroma relation interval set* and the second element in the pair—the *chroma relation chroma set*—is the chroma set to which the chroma relation applies.

Given two chromae c_1 and c_2 and a chroma relation $R^c = \langle \underline{i}(R^c), \underline{c}(R^c) \rangle$ then the expression

$$c_1 R^c c_2$$

is the logical proposition ‘Chroma c_1 is related to chroma c_2 by chroma relation R^c .’ The truth-value of this proposition can be determined from the following definition:

$$\text{truth}(c_1 R^c c_2) =_{\text{df}} \text{truth} \left(\exists p_1, p_2 \left(\begin{array}{l} (p_1, p_2 \in \underline{p}^u) \wedge \\ (\underline{i}(p_1, p_2) \in \underline{i}(R^c)) \wedge \\ (c_1, c_2 \in \underline{c}(R^c)) \wedge \\ (c_1 = \underline{c}(p_1)) \wedge \\ (c_2 = \underline{c}(p_2)) \end{array} \right) \right)$$

An object R^m is a *morph relation* if and only if it is an ordered pair

$$R^m = \langle \underline{i}(R^m), \underline{m}(R^m) \rangle$$

where the first element of the pair is a pitch interval set called the *morph relation interval set* and the second element in the pair—the *morph relation morph set*—is the morph set to which the morph relation applies.

Given two morphs m_1 and m_2 and a morph relation $R^m = \langle \underline{i}(R^m), \underline{m}(R^m) \rangle$ then the expression

$$m_1 R^m m_2$$

is the logical proposition ‘Morph m_1 is related to morph m_2 by morph relation R^m .’ The truth-value of this proposition can be determined from the following definition:

$$\text{truth}(m_1 R^m m_2) =_{\text{df}} \text{truth} \left(\exists p_1, p_2 \left(\begin{array}{l} (p_1, p_2 \in \underline{p}^u) \wedge \\ (\underline{i}(p_1, p_2) \in \underline{i}(R^m)) \wedge \\ (m_1, m_2 \in \underline{m}(R^m)) \wedge \\ (m_1 = \underline{m}(p_1)) \wedge \\ (m_2 = \underline{m}(p_2)) \end{array} \right) \right)$$

An object R^q is a *genus relation* if and only if it is an ordered pair

$$R^q = \langle \underline{i}(R^q), \underline{q}(R^q) \rangle$$

where the first element of the pair is a pitch interval set called the *genus relation interval set* and the second element in the pair—the *genus relation genus set*—is the genus set to which the genus relation applies.

Given two genera q_1 and q_2 and a genus relation $R^q = \langle \underline{i}(R^q), \underline{q}(R^q) \rangle$ then the expression

$$q_1 R^q q_2$$

is the logical proposition ‘Genus q_1 is related to genus q_2 by genus relation R^q .’ The truth-value of this proposition can be determined from the following definition:

$$\text{truth}(q_1 R^q q_2) =_{\text{df}} \text{truth} \left(\exists p_1, p_2 \left(\begin{array}{l} (p_1, p_2 \in \underline{p}^u) \wedge \\ (\mathbf{i}(p_1, p_2) \in \mathbf{i}(R^q)) \wedge \\ (q_1, q_2 \in \underline{q}(R^q)) \wedge \\ (q_1 = \mathbf{q}(p_1)) \wedge \\ (q_2 = \mathbf{q}(p_2)) \end{array} \right) \right)$$

24 Transposition and inversion of pitch sets

24.1 Transposition of pitch sets

Given two pitch sets,

$$\underline{p}_1 = \{p_{1,1}, p_{1,2}, \dots, p_{1,j}, \dots, p_{1,|\underline{p}_1|}\} \quad \text{and} \quad \underline{p}_2 = \{p_{2,1}, p_{2,2}, \dots, p_{2,k}, \dots, p_{2,|\underline{p}_2|}\}$$

then \underline{p}_1 and \underline{p}_2 are *transpositionally equivalent* if and only if $|\underline{p}_1| = |\underline{p}_2|$ and there exists a single pitch interval i such that for all $p_{1,j} \in \underline{p}_1$ there exists $p_{2,k} \in \underline{p}_2$ such that

$$i(p_{1,j}, p_{2,k}) = i$$

The *universal set of pitch sets* \underline{P}^u is defined to be the set that contains all and only pitch sets. \underline{P}^u is an infinite set. The universal set of pitch sets is exhaustively and exclusively partitioned into *transpositional equivalence classes of pitch sets*. The transpositional equivalence class to which any given pitch set \underline{p} belongs is the set that contains all and only those pitch sets that are transpositionally equivalent to \underline{p} . The transpositional equivalence class to which a pitch set \underline{p} belongs will be denoted $\underline{P}_{\text{tran}}(\underline{p})$.

Let

$$\underline{p}_1 = \{p_{a,1}, p_{a,2}, \dots, p_{a,j}, \dots, p_{a,|\underline{p}_1|}\} \quad \text{and} \quad \underline{p}_2 = \{p_{b,1}, p_{b,2}, \dots, p_{b,k}, \dots, p_{b,|\underline{p}_2|}\}$$

and let also

$$\underline{\theta}_a = \langle p_{a,1}, p_{a,2}, \dots, p_{a,j}, \dots, p_{a,|\underline{p}_1|} \rangle \quad \text{and} \quad \underline{\theta}_b = \langle p_{b,1}, p_{b,2}, \dots, p_{b,j}, \dots, p_{b,|\underline{p}_2|} \rangle$$

be the two ordered pitch sets that satisfy the following conditions:

1. $\underline{p}_1 = \bigcup_{\underline{\theta}_a} \{p_{a,j}\}$ and $\underline{p}_2 = \bigcup_{\underline{\theta}_b} \{p_{b,j}\}$;
2. for all $p_{a,j}, p_{a,j+1} \in \underline{\theta}_a$ it is true that $p_{a,j} < p_{a,j+1}$;
3. for all $p_{b,j}, p_{b,j+1} \in \underline{\theta}_b$ it is true that $p_{b,j} < p_{b,j+1}$.

\underline{p}_1 is defined to be *less than* \underline{p}_2 , denoted $\underline{p}_1 < \underline{p}_2$, if and only if at least one of the following conditions is satisfied:

1. $|\underline{p}_1| < |\underline{p}_2|$
2. $|\underline{p}_1| = |\underline{p}_2|$ and there exists some number n such that $p_{a,n} < p_{b,n}$ and $p_{a,j} = p_{b,j}$ for all $j < n$.

Each transpositional equivalence class $\underline{P}^{\text{tran}}$ contains one pitch set which is less than every other pitch set in $\underline{P}^{\text{tran}}$. This pitch set is called the *transpositionally prime pitch set* of $\underline{P}^{\text{tran}}$ and is denoted $\underline{p}_{\text{tran}}(\underline{P}^{\text{tran}})$. The transpositionally prime pitch set of a pitch set \underline{p} is defined and denoted as follows:

$$\underline{p}_{\text{tran}}(\underline{p}) =_{\text{df}} \underline{p}_{\text{tran}}(\underline{P}_{\text{tran}}(\underline{p}))$$

Given a pitch set,

$$\underline{p} = \{p_1, p_2, \dots, p_k, \dots, p_{|\underline{p}|}\}$$

then the value returned by the transposition function introduced in chapter 22 above when this function is given a pitch set as its first argument is defined as follows:

$$\text{tran}(\underline{p}, i) =_{\text{df}} \bigcup_{k=1}^{|\underline{p}|} \{\text{tran}(p_k, i)\}$$

24.2 Inversion of pitch sets

Given a pitch set,

$$\underline{p} = \{p_1, p_2, \dots, p_j, \dots, p_{|\underline{p}|}\}$$

then the *lowest pitch* in \underline{p} , denoted $p_{\min}(\underline{p})$, is the pitch in \underline{p} that is less than every other pitch in \underline{p} ; and the inversion of \underline{p} is denoted and defined by the following equation:

$$\text{inv}(\underline{p}) = \bigcup_{j=1}^{|\underline{p}|} \{\text{tran}(p_{\min}(\underline{p}), i(p_j, p_{\min}(\underline{p})))\}$$

Two pitch sets \underline{p}_1 and \underline{p}_2 are *inversionally equivalent* if and only if there exists a pitch interval i such that

$$(\underline{p}_2 = \text{tran}(\text{inv}(\underline{p}_1), i)) \vee (\underline{p}_2 = \text{tran}(\underline{p}_1, i))$$

This implies that the universal set of pitch sets P^u is exhaustively and exclusively partitioned into *inversional equivalence classes of pitch sets*. The inversional equivalence class to which any given pitch set \underline{p} belongs is the set that contains all and only those pitch sets that are inversionally equivalent to \underline{p} . The inversional equivalence class to which a pitch set \underline{p} belongs will be denoted $\underline{P}_{\text{inv}}(\underline{p})$.

Each inversional equivalence class $\underline{P}^{\text{inv}}$ contains one pitch set that is less than every other pitch set in $\underline{P}^{\text{inv}}$. This pitch set is called the *inversionally prime pitch set* of $\underline{P}^{\text{inv}}$ and is denoted $\underline{p}_{\text{inv}}(\underline{P}^{\text{inv}})$. The inversionally prime pitch set of a pitch set \underline{p} is defined and denoted as follows:

$$\underline{p}_{\text{inv}}(\underline{p}) =_{\text{df}} \underline{p}_{\text{inv}}(\underline{P}_{\text{inv}}(\underline{p}))$$

25 Representing chroma sets using set numbers

The *universal set of chroma sets* \underline{C}^u is the set that contains all and only chroma sets. The universal set of chroma sets contains 12 members therefore the cardinality of the universal set of chroma sets is $2^{12} = 4096$.

Given a chroma set,

$$\underline{c} = \{c_1, c_2, \dots, c_k, \dots, c_{|\underline{c}|}\}$$

then the *set number* of \underline{c} , denoted $\mu(\underline{c})$, is defined as follows:

$$\mu(\underline{c}) = \sum_{k=1}^{|\underline{c}|} 2^{c_k} - 1$$

No two distinct chroma sets have the same set number and every non-negative integer less than 4096 is the set number of a chroma set. The content of a chroma set can be derived algorithmically from its set number. Specific chroma sets can therefore be denoted by writing the set number as a superscript as follows:

$$\underline{c} = \underline{c}^{\mu(\underline{c})}$$

For example, if $\underline{c}_1 = \{4, 6, 10\}$ then

$$\begin{aligned} \mu(\underline{c}_1) &= \sum_{k=1}^{|\underline{c}_1|} 2^{c_k} \\ &= 2^4 + 2^6 + 2^{10} \\ &= 16 + 64 + 1024 \\ &= \underline{1104} \end{aligned}$$

therefore

$$\underline{c}_1 = \{4, 6, 10\} = \underline{c}^{1104}$$

Note in particular that $\underline{c}^0 = \emptyset$, that $\underline{c}^{4095} = \underline{c}^u$ and that therefore

$$\underline{C}^u = \{\underline{c}^0, \underline{c}^1, \dots, \underline{c}^{4095}\}$$

Given a chroma set \underline{c}^k such that $k = \mu(\underline{c}^k)$ then the content of \underline{c}^k can be derived from its set number k using the following algorithm (in which expressions of the form ' $r := k$ ' should be read 'Make r equal to k '⁴⁴⁹):

1. $r := k$
2. $\underline{c} := \emptyset$
3. If $r > 0$ then go to step 4 else output \underline{c} which gives the content of \underline{c}^k .
4. $\underline{c} := \underline{c} \cup \{\text{int}(\log_2 r)\}$
5. $r := r - 2^{\text{int}(\log_2 r)}$

⁴⁴⁹ See Borowski and Borwein 1989, 35, entry for 'assignment', sense 2.

6. Go to step 3.

For example, the content of the set \underline{c}^{1104} would be derived as follows:

1. $r := 1104$

2. $\underline{c} := \emptyset$

3. $r = 1104$ therefore $r > 0$ therefore $\underline{c} := \underline{c} \cup \{\text{int}(\log_2 1104)\} = \emptyset \cup \{10\} = \{10\}$

4. $r := 1104 - 2^{10} = 1104 - 1024 = 80$

5. $r = 80$ therefore $r > 0$ therefore $\underline{c} := \underline{c} \cup \{\text{int}(\log_2 80)\} = \{10\} \cup \{6\} = \{10, 6\}$

6. $r := 80 - 2^6 = 80 - 64 = 16$

7. $r = 16$ therefore $r > 0$ therefore $\underline{c} := \underline{c} \cup \{\text{int}(\log_2 16)\} = \{10, 6\} \cup \{4\} = \{10, 6, 4\}$

8. $r := 16 - 2^4 = 16 - 16 = 0$

9. $r = 0$ therefore $\underline{c}^{1104} = \{10, 6, 4\}.3$

26 Representing morph sets using set numbers

The *universal set of morph sets* \underline{M}^u is the set that contains all and only morph sets. The universal set of morphs contains 7 members therefore the cardinality of the universal set of morph sets is $2^7 = 128$.

Given a morph set,

$$\underline{m} = \{m_1, m_2, \dots, m_k, \dots, m_{|\underline{m}|}\}$$

then the *set number* of \underline{m} , denoted $\mu(\underline{m})$, is defined as follows:

$$\mu(\underline{m}) = \sum_{k=1}^{|\underline{m}|} 2^{m_k} 4$$

No two distinct morph sets have the same set number and every non-negative integer less than 128 is the set number of a morph set. Specific morph sets can therefore be denoted by writing the set number as a superscript as follows:

$$\underline{m} = \underline{m}^{\mu(\underline{m})}$$

For example, if $\underline{m}_1 = \{2,3,5\}$ then

$$\begin{aligned} \mu(\underline{m}_1) &= \sum_{k=1}^{|\underline{m}_1|} 2^{m_k} \\ &= 2^2 + 2^3 + 2^5 \\ &= 4 + 8 + 32 \\ &= \underline{44} \end{aligned}$$

therefore

$$\underline{m}_1 = \{2,3,5\} = \underline{m}^{44}$$

Note in particular that $\underline{m}^0 = \emptyset$, that $\underline{m}^{127} = \underline{m}^u$ and that therefore

$$\underline{M}^u = \{\underline{m}^0, \underline{m}^1 \dots \underline{m}^{127}\}$$

The content of a morph set can be derived from its set number using the algorithm described above for determining the content of a chroma set from its set number.

27 Transposition of chroma sets

Given two chroma sets,

$$\underline{c}_1 = \{c_{1,1}, c_{1,2}, \dots, c_{1,j}, \dots, c_{1,|\underline{c}_1|}\} \quad \text{and} \quad \underline{c}_2 = \{c_{2,1}, c_{2,2}, \dots, c_{2,k}, \dots, c_{2,|\underline{c}_2|}\}$$

then \underline{c}_1 and \underline{c}_2 are *transpositionally equivalent* if and only if $|\underline{c}_1| = |\underline{c}_2|$ and there exists a single pitch interval $i = \langle i^c, i^m \rangle$ such that for all $c_{1,j} \in \underline{c}_1$ there exists $c_{2,k} \in \underline{c}_2$ such that

$$c_{2,k} = (c_{1,j} + i^c) \bmod 12$$

The universal set of chroma sets \underline{C}^u is exhaustively and exclusively partitioned into *transpositional equivalence classes of chroma sets*. The transpositional equivalence class to which any given chroma set \underline{c} belongs is the set that contains all and only those chroma sets that are transpositionally equivalent to \underline{c} . The transpositional equivalence class to which a chroma set \underline{c} belongs will be denoted $\underline{C}_{\text{tran}}(\underline{c})$.

The chroma set with the lowest set number in a transpositional equivalence class of chroma sets $\underline{C}^{\text{tran}}$ is called the *transpositionally prime chroma set* of $\underline{C}^{\text{tran}}$ and is denoted $\underline{c}_{\text{tran}}(\underline{C}^{\text{tran}})$. The transpositionally prime chroma set of a chroma set \underline{c} is defined and denoted as follows:

$$\underline{c}_{\text{tran}}(\underline{c}) =_{\text{df}} \underline{c}_{\text{tran}}(\underline{C}_{\text{tran}}(\underline{c}))$$

Given a chroma set,

$$\underline{c} = \{c_1, c_2, \dots, c_k, \dots, c_{|\underline{c}|}\}$$

then the value returned by the transposition function introduced in chapter 22 above when this function is given a chroma set as its first argument is defined as follows:

$$\text{tran}(\underline{c}, i) =_{\text{df}} \bigcup_{k=1}^{|\underline{c}|} \{\text{tran}(c_k, i)\}$$

However, given a chroma set \underline{c} whose set number is k (i.e. $\underline{c} = \underline{c}^k$) then it can be shown that

$$\mu(\text{tran}(\underline{c}^k, i)) = \left((2^{i_c(i) \bmod 12} \times k) \bmod 12 \right) + \left((2^{i_c(i) \bmod 12} \times k) \text{div} 12 \right)$$

In general, it is computationally more efficient to transpose a chroma set via its set number using this equation than it is to apply the transposition function to each member of the chroma set individually. Given a chroma set \underline{c}^k , the set number of the transpositionally prime chroma set of \underline{c}^k can be calculated directly from its set number k . It is first necessary to find the set $\underline{\mu}_{\text{tran}}(\underline{c}^k)$ which is the set that contains all and only set numbers of chroma sets transpositionally equivalent to \underline{c}^k . This can be found using the following equation:

$$\underline{\mu}_{\text{tran}}(\underline{c}^k) = \bigcup_{n=0}^{11} \left\{ \left((2^n k) \bmod 12 \right) + \left((2^n k) \text{div} 12 \right) \right\}$$

The set number of $\underline{c}_{\text{tran}}(\underline{c}^k)$, denoted $\mu_{\text{tran}}(\underline{c}^k)$, is the least member of $\underline{\mu}_{\text{tran}}(\underline{c}^k)$. That is,

$$\mu(\underline{c}_{\text{tran}}(\underline{c}^k)) = \mu_{\text{tran}}(\underline{c}^k) = \min(\underline{\mu}_{\text{tran}}(\underline{c}^k))$$

28 Inversion of chroma sets

The *inversion* of a chroma set $\underline{c} = \{c_1, c_2, \dots, c_k, \dots, c_{|\underline{c}|}\}$ is defined and denoted as follows:

$$\text{inv}(\underline{c}) = \bigcup_{j=1}^{|\underline{c}|} \{(-c_j) \bmod 12\}$$

Given a chroma set \underline{c} , then the set number of $\text{inv}(\underline{c})$ can be determined directly from the set number of \underline{c} using the following equation:

$$\mu(\text{inv}(\underline{c})) = \text{rev}(12, \mu(\underline{c}))$$

Note that this equation uses the ‘binary reverse’ function defined in section 19.4 above.

Two chroma sets \underline{c}_1 and \underline{c}_2 are *inversionally equivalent* if and only if there exists a pitch interval i such that

$$\underline{c}_2 = \text{tran}(\text{inv}(\underline{c}_1), i)$$

This implies that the universal set of chroma sets \underline{C}^u is exhaustively and exclusively partitioned into *inversional equivalence classes of chroma sets*. The inversional equivalence class to which any given chroma set \underline{c} belongs is the set that contains all and only those chroma sets that are inversionally equivalent to \underline{c} . The inversional equivalence class to which a chroma set \underline{c} belongs will be denoted $\underline{C}_{\text{inv}}(\underline{c})$.

The chroma set with the lowest set number in an inversional equivalence class of chroma sets $\underline{C}^{\text{inv}}$ is called the *inversionally prime chroma set* of $\underline{C}^{\text{inv}}$ and is denoted $\underline{c}_{\text{inv}}(\underline{C}^{\text{inv}})$. The inversionally prime chroma set of a chroma set \underline{c} is defined and denoted as follows:

$$\underline{c}_{\text{inv}}(\underline{c}) =_{\text{df}} \underline{c}_{\text{inv}}(\underline{C}_{\text{inv}}(\underline{c}))$$

The set number of $\underline{c}_{\text{inv}}(\underline{c})$, denoted $\mu_{\text{inv}}(\underline{c})$, is defined as follows:

$$\mu(\underline{c}_{\text{inv}}(\underline{c})) = \mu_{\text{inv}}(\underline{c}) = \min(\underline{\mu}_{\text{tran}}(\underline{c}) \cup \underline{\mu}_{\text{tran}}(\text{inv}(\underline{c})))$$

29 Transposition of morph sets

Given two morph sets,

$$\underline{m}_1 = \{m_{1,1}, m_{1,2}, \dots, m_{1,j}, \dots, m_{1,|\underline{m}_1}|\} \quad \text{and} \quad \underline{m}_2 = \{m_{2,1}, m_{2,2}, \dots, m_{2,k}, \dots, m_{2,|\underline{m}_2}|\}$$

then \underline{m}_1 and \underline{m}_2 are *transpositionally equivalent* if and only if $|\underline{m}_1| = |\underline{m}_2|$ and there exists a single pitch interval $i = \langle i^c, i^m \rangle$ such that for all $m_{1,j} \in \underline{m}_1$ there exists $m_{2,k} \in \underline{m}_2$ such that

$$m_{2,k} = (m_{1,j} + i^m) \bmod 7$$

The universal set of morph sets \underline{M}^u is exhaustively and exclusively partitioned into *transpositional equivalence classes of morph sets*. The transpositional equivalence class to which any given morph set \underline{m} belongs is the set that contains all and only those morph sets that are transpositionally equivalent to \underline{m} . The transpositional equivalence class to which a morph set \underline{m} belongs will be denoted $\underline{M}_{\text{tran}}(\underline{m})$.

The morph set with the lowest set number in a transpositional equivalence class of morph sets $\underline{M}^{\text{tran}}$ is called the *transpositionally prime morph set* of $\underline{M}^{\text{tran}}$ and is denoted $\underline{m}_{\text{tran}}(\underline{M}^{\text{tran}})$. The transpositionally prime morph set of a morph set \underline{m} is defined and denoted as follows:

$$\underline{m}_{\text{tran}}(\underline{m}) =_{\text{df}} \underline{m}_{\text{tran}}(\underline{M}_{\text{tran}}(\underline{m}))$$

Given a morph set,

$$\underline{m} = \{m_1, m_2, \dots, m_k, \dots, m_{|\underline{m}|}\}$$

then the value returned by the transposition function introduced in chapter 22 above when this function is given a morph set as its first argument is defined as follows:

$$\text{tran}(\underline{m}, i) =_{\text{df}} \bigcup_{k=1}^{|\underline{m}|} \{\text{tran}(m_k, i)\}$$

However, given a morph set \underline{m} whose set number is k (i.e. $\underline{m} = \underline{m}^k$) then it can be shown that

$$\mu(\text{tran}(\underline{m}^k, i)) = ((2^{i_m(i) \bmod 7} \times k) \bmod 7) + ((2^{i_m(i) \bmod 7} \times k) \text{div } 7)$$

In general, it is computationally more efficient to transpose a morph set via its set number using this equation than it is to apply the transposition function to each member of the morph set individually. Given a morph set \underline{m}^k , the set number of the transpositionally prime morph set of \underline{m}^k can be calculated directly from its set number k . It is first necessary to find the set $\underline{\mu}_{\text{tran}}(\underline{m}^k)$ which is the set that contains all and only set numbers of morph sets transpositionally equivalent to \underline{m}^k . This can be found using the following equation:

$$\underline{\mu}_{\text{tran}}(\underline{m}^k) = \bigcup_{n=0}^6 \{((2^n k) \bmod 7) + ((2^n k) \text{div } 7)\}$$

The set number of $\underline{m}_{\text{tran}}(\underline{m}^k)$, denoted $\mu_{\text{tran}}(\underline{m}^k)$, is the least member of $\underline{\mu}_{\text{tran}}(\underline{m}^k)$. That is,

$$\mu(\underline{m}_{\text{tran}}(\underline{m}^k)) = \mu_{\text{tran}}(\underline{m}^k) = \min(\underline{\mu}_{\text{tran}}(\underline{m}^k))$$

30 Inversion of morph sets

The *inversion* of a morph set $\underline{m} = \{m_1, m_2, \dots, m_k, \dots, m_{|\underline{m}|}\}$ is defined and denoted as follows:

$$\text{inv}(\underline{m}) = \bigcup_{j=1}^{|\underline{m}|} \{(-m_j) \bmod 7\}$$

Given a morph set \underline{m} , then the set number of $\text{inv}(\underline{m})$ can be determined directly from the set number of \underline{m} using the following equation:

$$\mu(\text{inv}(\underline{m})) = \text{rev}(7, \mu(\underline{m}))$$

Two morph sets \underline{m}_1 and \underline{m}_2 are *inversionally equivalent* if and only if there exists a pitch interval i such that

$$\underline{m}_2 = \text{tran}(\text{inv}(\underline{m}_1), i)$$

This implies that the universal set of morph sets \underline{M}^u is exhaustively and exclusively partitioned into *inversional equivalence classes of morph sets*. The inversional equivalence class to which any given morph set \underline{m} belongs is the set that contains all and only those morph sets that are inversionally equivalent to \underline{m} . The inversional equivalence class to which a morph set \underline{m} belongs will be denoted $\underline{M}_{\text{inv}}(\underline{m})$.

The morph set with the lowest set number in an inversional equivalence class of morph sets $\underline{M}^{\text{inv}}$ is called the *inversionally prime morph set* of $\underline{M}^{\text{inv}}$ and is denoted $\underline{m}_{\text{inv}}(\underline{M}^{\text{inv}})$. The inversionally prime morph set of a morph set \underline{m} is defined and denoted as follows:

$$\underline{m}_{\text{inv}}(\underline{m}) =_{\text{df}} \underline{m}_{\text{inv}}(\underline{M}_{\text{inv}}(\underline{m}))$$

The set number of $\underline{m}_{\text{inv}}(\underline{m})$, denoted $\mu_{\text{inv}}(\underline{m})$, is defined as follows:

$$\mu(\underline{m}_{\text{inv}}(\underline{m})) = \mu_{\text{inv}}(\underline{m}) = \min(\mu_{\text{tran}}(\underline{m}) \cup \mu_{\text{tran}}(\text{inv}(\underline{m})))$$

31 Transposition and inversion of genus sets

31.1 Transposition of genus sets

Given two genus sets,

$$\underline{q}_1 = \{q_{1,1}, q_{1,2}, \dots, q_{1,j}, \dots, q_{1,|\underline{q}_1}|\} \quad \text{and} \quad \underline{q}_2 = \{q_{2,1}, q_{2,2}, \dots, q_{2,k}, \dots, q_{2,|\underline{q}_2}|\}$$

then \underline{q}_1 and \underline{q}_2 are *transpositionally equivalent* if and only if the following two conditions are satisfied:

1. $|\underline{q}_1| = |\underline{q}_2|$
2. there exists a single pitch interval i such that for all $q_{1,j} \in \underline{q}_1$ there exists a genus $q_{2,k} \in \underline{q}_2$ such that $q_{2,k} = \text{tran}(q_{1,j}, i)$.

The *universal set of genus sets* \underline{Q}^u is defined to be the set that contains all and only genus sets. \underline{Q}^u is an infinite set. The universal set of genus sets is exhaustively and exclusively partitioned into *transpositional equivalence classes of genus sets*. The transpositional equivalence class to which any given genus set \underline{q} belongs is the set that contains all and only those genus sets that are transpositionally equivalent to \underline{q} . The transpositional equivalence class to which a genus set \underline{q} belongs will be denoted $\underline{Q}_{\text{tran}}(\underline{q})$.

Let

$$\underline{q}_1 = \{q_{1,1}, q_{1,2}, \dots, q_{1,j}, \dots, q_{1,|\underline{q}_1}|\} \quad \text{and} \quad \underline{q}_2 = \{q_{2,1}, q_{2,2}, \dots, q_{2,k}, \dots, q_{2,|\underline{q}_2}|\}$$

and let also

$$\underline{\theta}_a = \langle q_{a,1}, q_{a,2}, \dots, q_{a,j}, \dots, q_{a,|\underline{q}_1}|\rangle \quad \text{and} \quad \underline{\theta}_b = \langle q_{b,1}, q_{b,2}, \dots, q_{b,j}, \dots, q_{b,|\underline{q}_2}|\rangle$$

be the two ordered genus sets that satisfy the following conditions:

1. $\underline{q}_1 = \bigcup_{\underline{\theta}_a} \{q_{a,j}\}$ and $\underline{q}_2 = \bigcup_{\underline{\theta}_b} \{q_{b,j}\}$;
2. for all $q_{a,j}, q_{a,j+1} \in \underline{\theta}_a$ it is true that $q_{a,j} < q_{a,j+1}$;
3. for all $q_{b,j}, q_{b,j+1} \in \underline{\theta}_b$ it is true that $q_{b,j} < q_{b,j+1}$.

\underline{q}_1 is defined to be *less than* \underline{q}_2 , denoted $\underline{q}_1 < \underline{q}_2$, if and only if one of the following conditions is satisfied:

1. $|\underline{q}_1| < |\underline{q}_2|$
2. $|\underline{q}_1| = |\underline{q}_2|$ and there exists some number n such that $q_{a,n} < q_{b,n}$ and $q_{a,j} = q_{b,j}$ for all $j < n$.

Each transpositional equivalence class $\underline{Q}^{\text{tran}}$ contains one genus set which is less than every other member of $\underline{Q}^{\text{tran}}$. This genus set is called the *transpositionally prime*

genus set of Q^{tran} and is denoted $\underline{q}_{\text{tran}}(\underline{Q}^{\text{tran}})$. The transpositionally prime genus set of a genus set \underline{q} is defined and denoted as follows:

$$\underline{q}_{\text{tran}}(\underline{q}) =_{\text{df}} \underline{q}_{\text{tran}}(\underline{Q}_{\text{tran}}(\underline{q}))$$

Given a genus set,

$$\underline{q} = \{q_1, q_2, \dots, q_k, \dots, q_{|\underline{q}|}\}$$

then the value returned by the transposition function introduced in chapter 22 above when this function is given a genus set as its first argument is defined as follows:

$$\text{tran}(\underline{q}, i) =_{\text{df}} \bigcup_{k=1}^{|\underline{q}|} \{\text{tran}(q_k, i)\}$$

31.2 Inversion of genus sets

Given a genus $q = \langle m, e \rangle$, and given that

$$p_0(q) = \langle p_{c,0}(q), p_{m,0}(q) \rangle \text{ and } p_1(q) = \langle p_{c,1}(q), p_{m,1}(q) \rangle$$

where

$$p_{c,0}(q) = e + \left(\left(7 \left(\left(2 \left((m+2) \bmod 7 \right) \bmod 7 \right) + 8 \right) \bmod 12 \right) \right) \text{ and } p_{m,0}(q) = m$$

$$p_{c,1}(q) = p_{c,0}(q) + 12 \text{ and } p_{m,1}(q) = m + 7$$

then for any pair of genera, q_1 and q_2 , the function $i(q_1, q_2)$ returns the pitch interval that satisfies the following two conditions:

1. $i(q_1, q_2) \in \{i(p_0(q_1), p_0(q_2)), i(p_0(q_1), p_1(q_2))\}$
2. $i_c(i(q_1, q_2)) \geq 0$

Given a genus set,

$$\underline{q} = \{q_1, q_2, \dots, q_k, \dots, q_{|\underline{q}|}\}$$

the *inversion* of \underline{q} is denoted and defined as follows

$$\text{inv}(\underline{q}) =_{\text{df}} \bigcup_{j=1}^{|\underline{q}|} \{\text{tran}(q_{\min}(\underline{q}), i(q_j, q_{\min}(\underline{q})))\}$$

where $q_{\min}(\underline{q})$ is the *lowest genus* in \underline{q} which is defined to be the genus in \underline{q} that is less than every other genus in \underline{q} .

Two genus sets \underline{q}_1 and \underline{q}_2 are *inversionally equivalent* if and only if there exists a pitch interval i such that

$$(\underline{q}_2 = \text{tran}(\text{inv}(\underline{q}_1), i)) \vee (\underline{q}_2 = \text{tran}(\underline{q}_1, i))$$

This implies that the universal set of genus sets Q^u is exhaustively and exclusively partitioned into *inversional equivalence classes of genus sets*. The inversional

equivalence class to which any given genus set \underline{q} belongs is the set that contains all and only those genus sets that are inversionally equivalent to \underline{q} . The inversional equivalence class to which a genus set \underline{q} belongs will be denoted $\underline{Q}_{\text{inv}}(\underline{q})$.

Each inversional equivalence class $\underline{Q}^{\text{inv}}$ contains one genus set that is less than every other genus set in $\underline{Q}^{\text{inv}}$. This genus set is called the *inversionally prime genus set* of $\underline{Q}^{\text{inv}}$ and is denoted $\underline{q}_{\text{inv}}(\underline{Q}^{\text{inv}})$. The inversionally prime genus set of a genus set \underline{q} is defined and denoted as follows:

$$\underline{q}_{\text{inv}}(\underline{q}) =_{\text{df}} \underline{q}_{\text{inv}}(\underline{Q}_{\text{inv}}(\underline{q}))$$

32 The concept of a digraph

Pitch, chroma, morph and genus relations can be conveniently represented using *digraphs*. A *digraph* is an abstract mathematical object that can be visually represented as a set of points called *vertices*, connected by directed line segments called *arcs*.⁴⁵⁰ Figure 32-1 shows a digraph.

An object d is a *digraph* if and only if it is an ordered pair as follows,

$$d = \langle \underline{v}, \underline{a} \rangle$$

where the first element in the pair is the *vertex set* of d and the second element is the *arc set* of d . The function $\underline{v}(d)$ returns the vertex set of d and the function $\underline{a}(d)$ returns the arc set of d . The vertex set of a digraph is the set that contains all and only vertices in the digraph. For example, the vertex set of the digraph in Figure 32-1 is

$$\{v_1, v_2, v_3, v_4, v_5, v_6\}$$

The arc set of a digraph is the set of all and only arcs in the digraph. For example, the arc set of the digraph in Figure 32-1 is

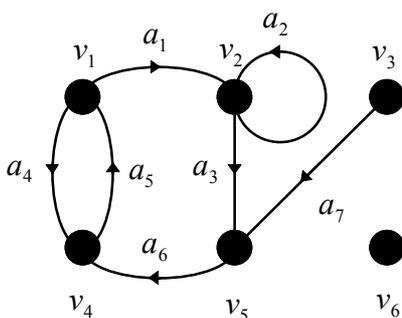


Figure 32-1

$$\{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$$

An object a is an *arc* if and only if it is an ordered pair as follows

$$a = \langle v_i(a), v_t(a) \rangle$$

where $v_i(a)$ and $v_t(a)$ are vertices in the vertex set of a specified digraph. $v_i(a)$ is the *initial vertex* of a and $v_t(a)$ is the *terminal vertex* of a . $v_i(a)$ and $v_t(a)$ taken together are the *endvertices* of a , denoted $\underline{v}(a)$. An arc is represented in its digraph by a directed line segment drawn *from* its initial vertex *to* its terminal vertex. For example, in Figure 32-1,

$$a_4 = \langle v_1, v_4 \rangle \Rightarrow \begin{cases} v_i(a_4) = v_1 \\ v_t(a_4) = v_4 \\ \underline{v}(a_4) = \{v_1, v_4\} \end{cases}$$

⁴⁵⁰ For more detailed introductions to graph theory see Wilson 1979 or Ore 1962.

Note that, in general, given two vertices, v_1 and v_2 , then

$$(v_1 \neq v_2) \Rightarrow (\langle v_1, v_2 \rangle \neq \langle v_2, v_1 \rangle)$$

For example, in Figure 32-1,

$$\left. \begin{array}{l} a_4 = \langle v_1, v_4 \rangle \\ a_5 = \langle v_4, v_1 \rangle \end{array} \right\} \Rightarrow a_4 \neq a_5$$

The number of arcs in a digraph for which a given vertex v is the *initial* vertex is called the *out-degree* of v and it is denoted $\rho_i(v)$ where the ‘i’ subscript stands for ‘initial.’ For example, the out-degree of vertex v_1 in Figure 32-1 is 2. The number of arcs in a digraph for which a given vertex v is the *terminal* vertex is called the *in-degree* of v and it is denoted $\rho_t(v)$ where the ‘t’ subscript stands for ‘terminal.’ For example, the in-degree of vertex v_1 in Figure 32-1 is 1.

Beginning at any vertex in a digraph, it is possible to ‘move’ from vertex to vertex, ‘travelling’ along arcs in the directions indicated by the arrowheads. For example, beginning at vertex v_4 in Figure 32-1, one might first traverse arc a_5 to get to v_1 , then go along a_1 to v_2 and from there to v_5 by way of a_3 . Such a ‘journey’ along the arcs of a digraph is called a *walk*. Borowski and Borwein define a walk to be ‘an alternating sequence of edges and vertices in a graph.’⁴⁵¹ In the context of this thesis, an object $\underline{\alpha}$ is a *walk* in a specified digraph if and only if it is an ordered set of arcs

$$\underline{\alpha} = \langle a_1, a_2, \dots, a_k, \dots, a_{|\underline{\alpha}|} \rangle$$

such that $v_i(a_{k+1}) = v_t(a_k)$ for all $1 \leq k < |\underline{\alpha}|$. For example, the walk in Figure 32-1 in which one travels from v_4 to v_1 to v_2 to v_5 would be denoted $\langle a_5, a_1, a_3 \rangle$.

It proves convenient to allow for walks to be denoted in an alternative manner in which the vertices passed through are simply written in the order in which they are encountered. For example, in Figure 32-1,

$$\langle a_5, a_1, a_3 \rangle = v_4 v_1 v_2 v_5$$

It must always be remembered, however, that a walk is an ordered set of *arcs* and *not* an ordered set of *vertices*.

The *length* of a walk $\underline{\alpha}$ is equal to the cardinality of $\underline{\alpha}$ —that is, the length of a walk is the number of arcs in the walk. The length of a walk $\underline{\alpha}$ is denoted $|\underline{\alpha}|$. For example, if $\underline{\alpha} = \langle a_5, a_1, a_3 \rangle$ then $|\underline{\alpha}| = 3$.

The first arc in a walk $\underline{\alpha}$ is called the *initial arc* of $\underline{\alpha}$ and it is denoted $a_i(\underline{\alpha})$. For example, if $\underline{\alpha} = \langle a_5, a_1, a_3 \rangle$ then $a_i(\underline{\alpha}) = a_5$. The initial vertex of the initial arc of a walk $\underline{\alpha}$ is called the *initial vertex* of $\underline{\alpha}$ and it is denoted $v_i(\underline{\alpha})$. For example, if $\underline{\alpha} = \langle a_5, a_1, a_3 \rangle$ in Figure 32-1 then $v_i(\underline{\alpha}) = v_4$.

If a walk $\underline{\alpha}$ is finite then the last arc in $\underline{\alpha}$ is called the *terminal arc* of $\underline{\alpha}$ and it is denoted $a_t(\underline{\alpha})$. For example, if $\underline{\alpha} = \langle a_5, a_1, a_3 \rangle$ then $a_t(\underline{\alpha}) = a_3$. The terminal vertex of the terminal arc of a walk $\underline{\alpha}$ is called the *terminal vertex* of $\underline{\alpha}$ and it is denoted $v_t(\underline{\alpha})$.

⁴⁵¹ Borowski and Borwein 1989, 631, entry for ‘walk’.

For example, if $\underline{\alpha} = \langle a_5, a_1, a_3 \rangle$ in Figure 32-1 then $v_t(\underline{\alpha}) = v_5$. A vertex is an *endvertex* of a walk $\underline{\alpha}$ if and only if it is either the initial or terminal vertex of $\underline{\alpha}$.

A vertex is an *inner vertex* of a walk if it is an endvertex of any arc in the walk other than the initial and terminal arcs in the walk. The second and penultimate vertices in a walk are therefore inner vertices by virtue of being the initial vertex of the second arc and the terminal vertex of the penultimate arc respectively. If the vertex which is the initial vertex of a walk is passed through at some other point in the walk, then it is also an inner vertex. For example, in Figure 32-1, the vertex v_4 is both an endvertex and an inner vertex of the walk $\langle a_5, a_1, a_3, a_6, a_5 \rangle$.

The *walk set* of a walk $\underline{\alpha}$ is the set that contains all and only those vertices that are endvertices of arcs in $\underline{\alpha}$. The walk set of a walk $\underline{\alpha}$ is denoted $\underline{v}(\underline{\alpha})$. For example, in Figure 32-1, the walk set of the walk $\langle a_5, a_1, a_3, a_6, a_5 \rangle$ is $\{v_1, v_2, v_4, v_5\}$.

A walk is a *closed walk* if and only if its initial vertex is the same as its terminal vertex. For example, in Figure 32-1, the walk $\langle a_5, a_1, a_3, a_6, a_5, a_4 \rangle$ is a closed walk.

A walk is a *trail* if and only if no two arcs in the walk are identical. For example, in Figure 32-1, the walk $\langle a_5, a_1, a_3, a_6, a_5, a_4 \rangle$ is *not* a trail, but the walk $\langle a_5, a_1, a_2, a_3 \rangle$ is a trail.

A trail is a *closed trail* if and only if its initial vertex is the same as its terminal vertex. For example, in Figure 32-1, the walk $\langle a_5, a_1, a_2, a_3, a_6 \rangle$ is a closed trail.

A trail $\underline{\alpha}$ is a *path* if and only if it satisfies one of the following conditions:

1. $|\underline{v}(\underline{\alpha})| = |\underline{\alpha}| + 1$ (i.e. it does not pass through any vertex more than once);
2. $|\underline{v}(\underline{\alpha})| = |\underline{\alpha}|$ and $v_i(\underline{\alpha}) = v_t(\underline{\alpha})$ (i.e. it does not pass through any vertex more than once except the initial vertex which is the same as the terminal vertex).

In Figure 32-1, the trail $\langle a_5, a_1, a_2, a_3, a_6 \rangle$ is *not* a path because it passes twice through the inner vertex v_2 . However, the walks $\langle a_7, a_6, a_5, a_1 \rangle$ and $\langle a_6, a_5, a_1, a_3 \rangle$ are both paths.

Finally, a path is a *circuit* if and only if its initial vertex is the same as its terminal vertex. That is, a walk $\underline{\alpha}$ is a circuit if and only if $|\underline{v}(\underline{\alpha})| = |\underline{\alpha}|$ and $v_i(\underline{\alpha}) = v_t(\underline{\alpha})$. In Figure 32-1 there are seven circuits as follows:

$$\begin{aligned} &\langle a_2 \rangle \\ &\langle a_4, a_5 \rangle \quad \langle a_5, a_4 \rangle \\ &\langle a_6, a_5, a_1, a_3 \rangle \quad \langle a_5, a_1, a_3, a_6 \rangle \quad \langle a_1, a_3, a_6, a_5 \rangle \quad \langle a_3, a_6, a_5, a_1 \rangle \end{aligned}$$

33 Pitch relation digraphs

Digraphs can be used to represent pitch, chroma, morph and genus relations. Given the pitch relation,

$$R^p = \langle \underline{i}(R^p), \underline{p}(R^p) \rangle$$

then the digraph that represents this relation is denoted

$$d(R^p) = \langle \underline{v}(d(R^p)), \underline{a}(d(R^p)) \rangle$$

and must satisfy the following conditions:

1. For every pitch $p \in \underline{p}(R^p)$ there must exist one and only one vertex $v \in \underline{v}(d(R^p))$ that represents p . The function $v(p)$ returns the vertex in $\underline{v}(d(R^p))$ that represents p .
2. For every vertex $v \in \underline{v}(d(R^p))$ there must exist one and only one pitch $p \in \underline{p}(R^p)$ that is represented by v . The function $p(v)$ returns the pitch in $\underline{p}(R^p)$ that is represented by vertex v .
3. Given any pair of pitches, $p_1, p_2 \in \underline{p}(R^p)$, then the arc $\langle v(p_1), v(p_2) \rangle$ will be a member of $\underline{a}(d(R^p))$ if and only if $\text{truth}(p_1 R^p p_2) = \text{True}$. That is,

$$(\text{truth}(p_1 R^p p_2) = \text{True}) \Leftrightarrow (\langle v(p_1), v(p_2) \rangle \in \underline{a}(d(R^p)))$$

Given an arc $a = \langle v(p_1), v(p_2) \rangle$, then the *arc interval* of a , denoted $i(a)$ is defined as follows:

$$a = \langle v(p_1), v(p_2) \rangle \Rightarrow i(a) = i(p_1, p_2)$$

A digraph is a *pitch relation digraph* if and only if it represents a pitch relation. It proves convenient to allow the vertices, arcs and walks in a pitch relation digraph to be denoted in an alternative manner in which vertices are replaced with the pitches that they represent. For example, an arc $\langle v(p_1), v(p_2) \rangle$ can be written $\langle p_1, p_2 \rangle$. That is,

$$\langle p_1, p_2 \rangle =_{\text{df}} \langle v(p_1), v(p_2) \rangle$$

Similarly,

$$a \in \underline{a}(d(R^p)) \Rightarrow \begin{cases} p_i(a) =_{\text{df}} v_i(a) \\ p_t(a) =_{\text{df}} v_t(a) \\ p(a) =_{\text{df}} \underline{v}(a) \end{cases}$$

Also, given that $\underline{\alpha}$ is a walk in a pitch relation digraph, such that

$$\underline{\alpha} = \langle a_1, a_1, \dots, a_k, \dots \rangle = v_1 v_2 \dots v_k \dots$$

then

$$v_k = v(p_k) \Rightarrow \underline{\alpha} =_{\text{df}} p_1 p_2 \dots p_k \dots$$

and

$$p_i(\underline{\alpha}) =_{\text{df}} v_i(\underline{\alpha})$$

$$p_t(\underline{\alpha}) =_{\text{df}} v_t(\underline{\alpha})$$

$$\underline{p}(\underline{\alpha}) =_{\text{df}} \underline{v}(\underline{\alpha})$$

Finally, given a set of walks in a specified pitch relation digraph,

$$\underline{A}^p = \{\underline{\alpha}_1, \underline{\alpha}_2, \dots, \underline{\alpha}_k, \dots, \underline{\alpha}_{|\underline{A}^p|}\}$$

then the function $\underline{P}(\underline{A}^p)$ returns the set that contains all and only those pitch sets that are walk sets of walks in \underline{A}^p . That is,

$$\underline{P}(\underline{A}^p) =_{\text{df}} \bigcup_{k=1}^{|\underline{A}^p|} \{\underline{p}(\underline{\alpha}_k)\}$$

I shall not explicitly define the concepts of chroma, morph and genus relation digraphs. Suffice it to say that these concepts are defined on a strict analogy with that of a pitch relation digraph.

34 The thirds relations

A pitch, chroma, morph or genus relation $R = \langle \underline{i}(R), x \rangle$ is defined to be a *thirds relation* if and only if it satisfies the following conditions:

1. $\underline{i}(R)$ —the relation interval set of R —must be equal to $\{\langle 4,2 \rangle, \langle 3,2 \rangle\}$. That is, the relation interval set of R must contain only those pitch intervals that correspond to a rising major third or a rising minor third.
2. x , the second element of R , must be equal to \underline{p}^u if R is a pitch relation, \underline{c}^u if R is a chroma relation, \underline{m}^u if R is a morph relation and \underline{q}^u if R is a genus relation.

The only relations that satisfy these conditions are the pitch relation $\langle \{\langle 4,2 \rangle, \langle 3,2 \rangle\}, \underline{p}^u \rangle$, the chroma relation $\langle \{\langle 4,2 \rangle, \langle 3,2 \rangle\}, \underline{c}^u \rangle$, the morph relation $\langle \{\langle 4,2 \rangle, \langle 3,2 \rangle\}, \underline{m}^u \rangle$ and the genus relation $\langle \{\langle 4,2 \rangle, \langle 3,2 \rangle\}, \underline{q}^u \rangle$. From this point forwards, let

$$R_{\text{thirds}}^p = \langle \{\langle 4,2 \rangle, \langle 3,2 \rangle\}, \underline{p}^u \rangle \quad R_{\text{thirds}}^c = \langle \{\langle 4,2 \rangle, \langle 3,2 \rangle\}, \underline{c}^u \rangle$$

$$R_{\text{thirds}}^m = \langle \{\langle 4,2 \rangle, \langle 3,2 \rangle\}, \underline{m}^u \rangle \quad R_{\text{thirds}}^q = \langle \{\langle 4,2 \rangle, \langle 3,2 \rangle\}, \underline{q}^u \rangle$$

A digraph d is defined to be a *thirds relation digraph* if and only if it represents a thirds relation. There are therefore only four thirds relation digraphs as follows:

$$d(R_{\text{thirds}}^p) \quad d(R_{\text{thirds}}^c) \quad d(R_{\text{thirds}}^m) \quad d(R_{\text{thirds}}^q)$$

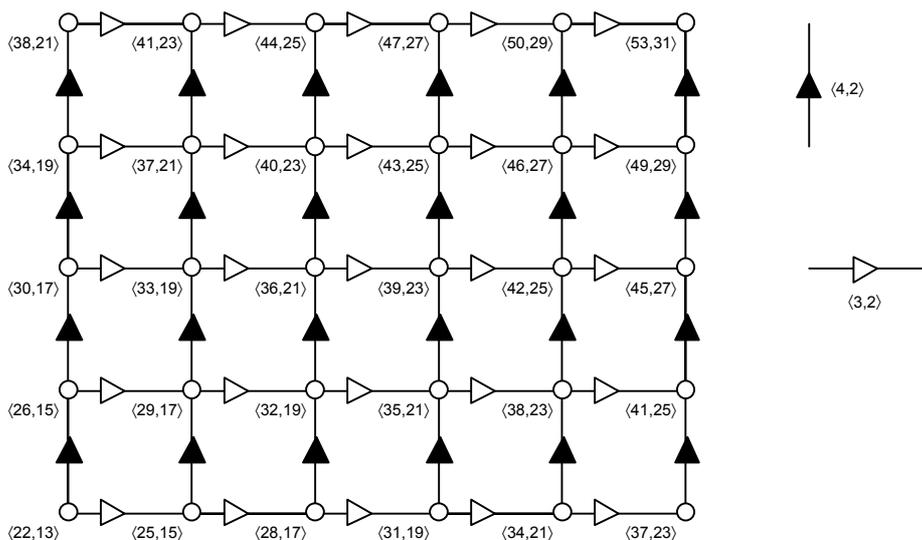


Figure 34-1

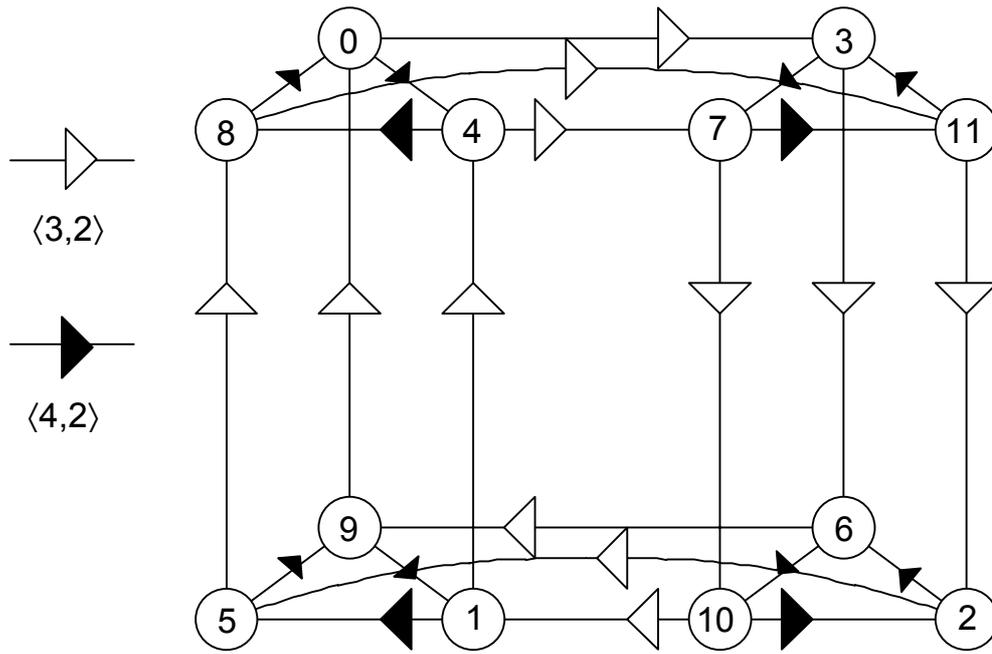


Figure 34-2

Figure 34-1 shows part of $d(R_{\text{thirds}}^p)$, Figure 34-2 shows $d(R_{\text{thirds}}^c)$, Figure 34-3 shows $d(R_{\text{thirds}}^m)$ and Figure 34-4 shows part of $d(R_{\text{thirds}}^q)$. The digraphs $d(R_{\text{thirds}}^p)$ and

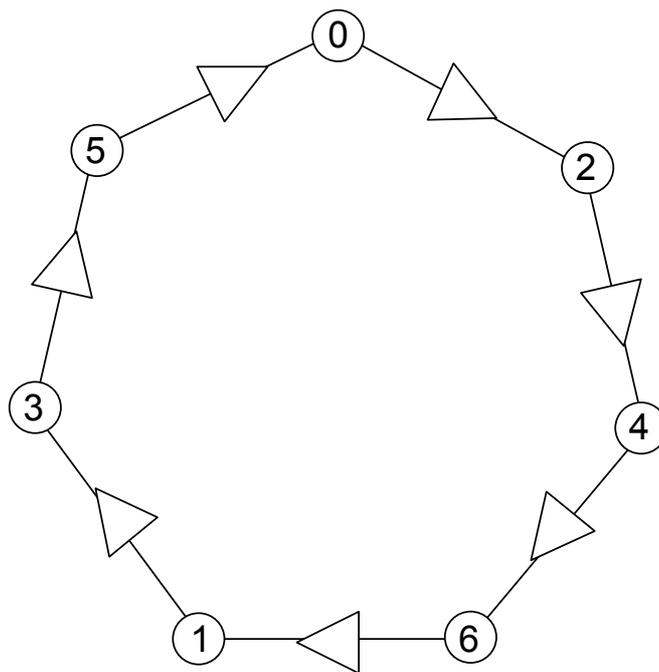


Figure 34-3

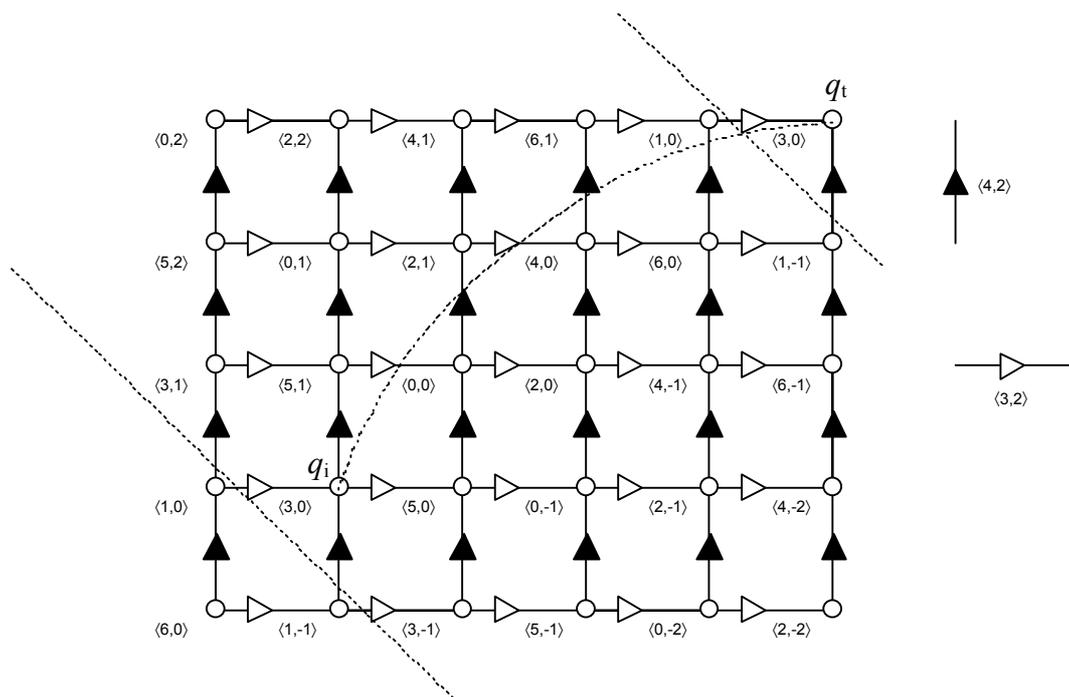


Figure 34-4

$d(R_{\text{thirds}}^q)$ are infinite—they each have an infinite vertex set corresponding respectively to the infinite universal set of pitches and the infinite universal set of genera.

All the morphetic pitches in Figure 34-1 are odd integers. This shows that $d(R_{\text{thirds}}^p)$ is a *disconnected* digraph. It has two infinite connected component digraphs—one containing those pitches whose morphetic pitches are even and the other containing those pitches whose morphetic pitches are odd. Each of these component digraphs can be embedded in a plane and is thus a *planar digraph*. $d(R_{\text{thirds}}^c)$, shown in Figure 34-2, is a finite connected digraph that can be embedded in the surface of a torus. $d(R_{\text{thirds}}^m)$ shown in Figure 34-3, is a finite, cyclic, planar digraph. $d(R_{\text{thirds}}^q)$ shown in Figure 34-4, is an infinite, connected digraph that can be embedded in the surface of a cylinder—in Figure 34-4, if the plane of the page is curved so that the dotted lines become coincident, then the cylinder can be constructed. $d(R_{\text{thirds}}^q)$ is closely related to Schoenberg’s ‘chart of the regions.’⁴⁵²

Longuet-Higgins (1979, 315-319) has discussed informally how the pitch relations

$$\langle \{ \langle 4,2 \rangle, \langle 7,4 \rangle, \langle 12,7 \rangle \}, \underline{p}^u \rangle \quad \text{and} \quad \langle \{ \langle 4,2 \rangle, \langle 7,4 \rangle \}, \underline{q}^u \rangle$$

can be used to represent major and minor chords and ‘extended keys’ in a way which may give insight into why these particular pitch collections seem to be employed as they are in tonal music. In particular, he points out that each 3×4 rectangular region in

⁴⁵² Schoenberg 1969, 20.

what he calls ‘harmonic space’ contains the ‘pitches’ in an ‘extended key.’ Longuet-Higgins’ ‘harmonic space’ corresponds approximately to the genus relation digraph $d(\langle\langle\{4,2\},\{7,4\}\rangle\rangle, \underline{q}^u)$ and his class of ‘extended keys’ corresponds to the transpositional equivalence class of genus name sets that contains the set:

$$\{\langle A, \natural \rangle, \langle E, \natural \rangle, \langle B, \natural \rangle, \langle F, \sharp \rangle, \langle F, \natural \rangle, \langle C, \natural \rangle, \langle G, \natural \rangle, \langle D, \natural \rangle, \langle D, \flat \rangle, \langle A, \flat \rangle, \langle E, \flat \rangle, \langle B, \flat \rangle\}$$

This particular genus name set is that associated with the extended key of C. Longuet-Higgins also points out that major and minor chords correspond to ‘L-shaped patterns’ in his ‘harmonic space.’⁴⁵³

Concepts approximately equivalent to $d(R_{\text{thirds}}^c)$, and the ways in which these concepts can be employed in the study of tonal musical structure have been discussed more or less formally by a number of writers including Balzano (1980), Shepard (1982), Krumhansl (1985) and Mazzola et al. (1989). For example, Balzano (1980) points out that the chroma sets associated with the major scales and the descending melodic minor scales—that is, the ‘diatonic’ chroma sets that form the transpositional equivalence class whose prime chroma set is

$$\underline{c}^{1387} = \{0,1,3,5,6,8,10\}$$

—are represented by ‘compact, “space-filling,”’ regions in the graphic representation of the direct-product group $C_3 \times C_4$.⁴⁵⁴ However, the chroma sets associated with the *harmonic* minor scales and the *ascending* melodic minor scales are not revealed to have any special group-theoretical properties when viewed from the perspective suggested by Balzano.

The chroma sets associated with the harmonic minor scales form the transpositional equivalence class of chroma sets whose prime chroma set is

$$\underline{c}^{859} = \{0,1,3,4,6,8,9\}$$

The chroma sets associated with the ascending melodic minor scales form the transpositional equivalence class of chroma sets whose prime chroma set is

$$\underline{c}^{1371} = \{0,1,3,4,6,8,10\}$$

Let the *universal set of historic scale-type chroma sets* $\underline{C}^{\text{hist}}$ be defined to be the set that contains all and only those chroma sets that are associated with major scales, harmonic minor scales, ascending melodic minor scales or descending melodic minor scales. $\underline{C}^{\text{hist}}$ is therefore defined as follows:

$$\underline{C}^{\text{hist}} =_{\text{df}} \underline{C}_{\text{tran}}(\underline{c}^{1387}) \cup \underline{C}_{\text{tran}}(\underline{c}^{859}) \cup \underline{C}_{\text{tran}}(\underline{c}^{1371})$$

I shall now show that considerations of symmetry and musical practice suggest that a fourth transpositional equivalence class of chroma sets should be added to the universal set of historical scale-type chroma sets to obtain what I shall call the *universal set of theoretical scale-type chroma sets*.

It would seem that the notion of constructing scales using disjunct and conjunct combinations of tetrachords can be traced back to classical Greek culture and that it was

⁴⁵³ Longuet-Higgins 1979, 316–7.

⁴⁵⁴ Balzano 1980, 72–73.

1. Major scale

0 - 2 - 4 - 5 (+2) 0 - 2 - 4 - 5
 E.G. C - D - E - F G - A - B - C

2. Ascending melodic minor scale

0 - 2 - 3 - 5 (+2) 0 - 2 - 4 - 5
 E.G. C - D - E^b - F G - A - B - C

3. Descending melodic minor scale

0 - 2 - 3 - 5 (+2) 0 - 1 - 3 - 5
 E.G. C - D - E^b - F G - A^b - B^b - C

4. Harmonic minor scale

0 - 2 - 3 - 5 (+2) 0 - 1 - 4 - 5
 E.G. C - D - E^b - F G - A^b - B - C

5. Possible combinations of tetrachords

<i>Lower</i>	<i>Upper</i>
L1 0-2-4-5	U1 0-2-4-5
L2 0-2-3-5	U2 0-1-4-5
	U3 0-1-3-5

	<i>Chroma set</i>	<i>Transpositionally prime chroma set</i>	<i>Name of set type</i>
L1 + U1:	{0,2,4,5,7,9,11}	{0,1,3,5,6,8,10}	DIATONIC
L1 + U2:	{0,2,4,5,7,8,11}	{0,1,3,5,6,8,9}	HARMONIC MAJOR
L1 + U3:	{0,2,4,5,7,8,10}	{0,1,3,4,6,8,10}	ASCENDING MELODIC MINOR
L2 + U1:	{0,2,3,5,7,9,11}	{0,1,3,4,6,8,10}	ASCENDING MELODIC MINOR
L2 + U2:	{0,2,3,5,7,8,11}	{0,1,3,4,6,8,9}	HARMONIC MINOR
L2 + U3:	{0,2,3,5,7,8,10}	{0,1,3,5,6,8,10}	DIATONIC

Figure 34-5

adopted by medieval theorists such as Boethius (ca. 475-520) and Alcuin (735-804). The idea of viewing scales as being constructed from tetrachords is therefore by no means new.

A major scale, as shown under heading 1 in Figure 34-5, can be interpreted as the disjunct combination of two {0,2,4,5}-type tetrachords, the lower one starting on the tonic and the upper one starting on the dominant. Similarly, the melodic ascending form of the minor scale as shown under heading 2 in Figure 34-5, can be interpreted as the disjunct combination of a lower {0,2,3,5}-type tetrachord starting on the tonic and an upper {0,2,4,5}-type tetrachord starting on the dominant. The melodic descending form of the minor scale, shown under heading 3 in Figure 34-5, can be interpreted as the disjunct combination of a lower {0,2,3,5}-type tetrachord starting on the tonic and an upper {0,1,3,5}-type tetrachord starting on the dominant. Finally, the harmonic form of the minor scale, as shown under heading 4 in Figure 34-5, can be interpreted as a

disjunct combination of a lower {0,2,3,5}-type tetrachord and an upper {0,1,4,5}-type tetrachord.

These interpretations suggest that in tonal scales, there are two different types of ‘allowable’ lower tetrachord and three different types of ‘allowable’ upper tetrachord. Moreover, the names that are given to the scales support this interpretation—if the lower tetrachord is {0,2,4,5}-type, then the scale is called ‘major’ and if it is {0,2,3,5}-type, it is called ‘minor.’ If the upper tetrachord is {0,2,4,5}-type (or {0,1,3,5}-type), the scale is called ‘melodic,’ and if it is {0,1,4,5}-type, the scale is called ‘harmonic.’

There are six ways in which these tetrachords can be disjunctly combined as shown under heading 5 in Figure 34-5. It can be seen from Figure 34-5 that the combination of a lower ‘major’ tetrachord with an upper ‘harmonic’ tetrachord (emboldened) gives rise to a new type of scale that I think one would have to call a ‘harmonic major’ scale. As far as I am aware, no tonal theorist has ever employed the notion of a harmonic major scale in his or her attempts to explain the pitch structure of tonal pieces. The class of chroma sets associated with the harmonic major scales is the transpositional equivalence class whose prime chroma set is

$$\underline{c}^{875} = \{0,1,3,5,6,8,9\}$$

Therefore, it can be said that at least with respect to their tetrachordal structure, the class of scales whose associated chroma sets are members of $\underline{C}^{\text{hist}}$ have no special properties that are not also possessed by scales whose associated chroma sets are members of $\underline{C}_{\text{tran}}(\underline{c}^{875})$. This suggests that it might prove useful to define a set

$$\underline{C}^{\text{theo}} = \underline{C}^{\text{hist}} \cup \underline{C}_{\text{tran}}(\underline{c}^{875})$$

I call $\underline{C}^{\text{theo}}$ the *universal set of theoretical scale-type chroma sets* and I call the set $\underline{C}_{\text{tran}}(\underline{c}^{875})$, the set of harmonic major scale-type chroma sets. Viewed from the perspective of tetrachords, $\underline{C}^{\text{theo}}$ is the set that contains all and only those chroma sets that are associated with scales that can be formed by disjunct combination of a {0,2,3,5} or {0,2,4,5}-type lower tetrachord with either a {0,1,4,5}, a {0,2,4,5} or a {0,1,3,5}-type upper tetrachord.

The possibility that $\underline{C}^{\text{theo}}$ may prove to be more useful in the description of tonal music than $\underline{C}^{\text{hist}}$ might also be suggested by the fact that $\underline{C}^{\text{theo}}$ is closed under the operation of chroma set inversion whereas $\underline{C}^{\text{hist}}$ is not.

The inversion of a diatonic-type chroma set, i.e. a set transpositionally equivalent to \underline{c}^{1387} , is another chroma set transpositionally equivalent to \underline{c}^{1387} . For example, the inversion of the chroma set associated with a C major scale

$$\{3,5,7,8,10,0,2\}$$

is the chroma set associated with the D major scale

$$\{5,7,9,10,0,2,4\}$$

In other words,

$$\underline{C}_{\text{tran}}(\underline{c}^{1387}) = \underline{C}_{\text{inv}}(\underline{c}^{1387})$$

Similarly, the inversion of an ascending melodic minor scale-type chroma set, i.e. a set transpositionally equivalent to \underline{c}^{1371} , returns another chroma set transpositionally

equivalent to \underline{c}^{1371} . For example, the inversion of the chroma set associated with an ascending B flat melodic minor scale

$$\{1,3,4,6,8,10,0\}$$

is the chroma set associated with the ascending F sharp melodic minor scale

$$\{9,11,0,2,4,6,8\}$$

Therefore

$$\underline{C}_{\text{tran}}(\underline{c}^{1371}) = \underline{C}_{\text{inv}}(\underline{c}^{1371})$$

However, the inversion of a harmonic minor scale-type chroma set, i.e. a set transpositionally equivalent to \underline{c}^{859} , returns a set that is a member of the transpositional equivalence class of chroma sets whose prime chroma set is

$$\underline{c}^{875} = \{0,1,3,5,6,8,9\}$$

That is, the inversion of a harmonic minor scale-type chroma set is a *harmonic major* scale-type chroma set. For example, the inversion of the chroma set associated with a C sharp harmonic minor scale

$$\{4,6,7,9,11,0,3\}$$

is the set

$$\{1,3,5,6,8,9,0\}$$

which is that associated with the B flat harmonic major scale:

$$\{\langle B, \flat \rangle, \langle C, \sharp \rangle, \langle D, \sharp \rangle, \langle E, \flat \rangle, \langle F, \sharp \rangle, \langle G, \flat \rangle, \langle A, \sharp \rangle\},$$

Therefore

$$\underline{C}_{\text{inv}}(\underline{c}^{859}) = \underline{C}_{\text{tran}}(\underline{c}^{859}) \cup \underline{C}_{\text{tran}}(\underline{c}^{875})$$

which implies that $\underline{C}_{\text{inv}}(\underline{c}^{859}) \not\subset \underline{C}^{\text{hist}}$ and therefore that $\underline{C}^{\text{hist}}$ is not closed under the operation of inversion. But the set

$$\underline{C}^{\text{theo}} = \underline{C}^{\text{hist}} \cup \underline{C}_{\text{tran}}(\underline{c}^{875})$$

is closed under transposition and inversion and can be obtained from the universal set of historical chroma sets by unifying it with the transpositional equivalence class that



Figure 34-6

contains the chroma sets associated with the harmonic major scales.

That the notion of a harmonic major scale might actually prove useful in the description of tonal music is suggested by the fact that such a scale contains all and only those pitches that are employed in a iv-V-I progression—that is, in a so-called ‘tierce de Picardie.’ Such a progression often occurs at the conclusion of pieces in the minor mode that employ a minor subdominant triad to prepare a final cadence onto the major tonic triad. For example, the set of pitches used in the cadential figure shown in Figure 34-6 form a harmonic major scale. This figure is taken from the conclusion of Bach’s chorale, ‘Zeuch ein zu deinen Toren’ (BWV 28/6, no.23 in Bach 1990).

The foregoing discussion has shown that with respect to their tetrachordal structure, the chroma sets in $\underline{C}^{\text{hist}}$ have no special properties that are not also possessed by chroma sets in $\underline{C}_{\text{tran}}(\underline{c}^{875})$. Furthermore, whereas $\underline{C}^{\text{hist}}$ is not closed under the operation of inversion, the set

$$\underline{C}^{\text{theo}} = \underline{C}^{\text{hist}} \cup \underline{C}_{\text{tran}}(\underline{c}^{875})$$

is closed under inversion. I have also shown that the harmonic major scales associated with $\underline{C}_{\text{tran}}(\underline{c}^{875})$ might prove useful in the description of tonal music.

In the next section I shall show that a certain highly suggestive graph-theoretical property is possessed by all and only those walks in $d(R_{\text{thirds}}^q)$ whose walk sets are the genus sets associated with the diatonic, harmonic minor, melodic minor and harmonic major scales.

35 Circuits in the thirds relation digraphs

I shall define a genus set to be a *diatonic genus set* if and only if it is transpositionally equivalent to

$$\underline{q}_1 = \{\langle 2,0 \rangle, \langle 3,0 \rangle, \langle 4,0 \rangle, \langle 5,0 \rangle, \langle 6,0 \rangle, \langle 0,0 \rangle, \langle 1,0 \rangle\}$$

\underline{q}_1 contains all and only the genera in a C major scale. The transpositional equivalence class of genus sets that contains \underline{q}_1 will be denoted $\underline{Q}^{\text{diat}}$.

A genus set is a *harmonic minor scale genus set* if and only if it is transpositionally equivalent to

$$\underline{q}_2 = \{\langle 2,0 \rangle, \langle 3,0 \rangle, \langle 4,-1 \rangle, \langle 5,0 \rangle, \langle 6,0 \rangle, \langle 0,-1 \rangle, \langle 1,0 \rangle\}$$

\underline{q}_2 contains all and only the genera in a C harmonic minor scale. The transpositional equivalence class of genus sets that contains \underline{q}_2 will be denoted $\underline{Q}^{\text{hami}}$.

A genus set is a *melodic minor scale genus set* if and only if it is transpositionally equivalent to

$$\underline{q}_3 = \{\langle 2,0 \rangle, \langle 3,0 \rangle, \langle 4,-1 \rangle, \langle 5,0 \rangle, \langle 6,0 \rangle, \langle 0,0 \rangle, \langle 1,0 \rangle\}$$

\underline{q}_3 contains all and only the genera in an ascending C melodic minor scale. The transpositional equivalence class of genus sets that contains \underline{q}_3 will be denoted $\underline{Q}^{\text{memi}}$.

A genus set is a *harmonic major scale genus set* if and only if it is transpositionally equivalent to

$$\underline{q}_4 = \{\langle 2,0 \rangle, \langle 3,0 \rangle, \langle 4,0 \rangle, \langle 5,0 \rangle, \langle 6,0 \rangle, \langle 0,-1 \rangle, \langle 1,0 \rangle\}$$

\underline{q}_4 contains all and only the genera in a C harmonic major scale. The transpositional equivalence class of genus sets that contains \underline{q}_4 will be denoted $\underline{Q}^{\text{hama}}$.

A genus set is defined to be a *tonal scale genus set* if and only if it is a member of

$$\underline{Q}^{\text{scale}} = \underline{Q}^{\text{diat}} \cup \underline{Q}^{\text{hami}} \cup \underline{Q}^{\text{memi}} \cup \underline{Q}^{\text{hama}} \quad (\text{Eq.1})$$

I shall call $\underline{Q}^{\text{scale}}$ the *universal set of tonal scale genus sets*.

Let $\underline{\alpha}_{\text{thirds}}^{\text{q}}$ be any finite walk whatsoever in the digraph $d(R_{\text{thirds}}^{\text{q}})$,

$$\underline{\alpha}_{\text{thirds}}^{\text{q}} = q_1 q_2 \dots q_k \dots q_n$$

and let

$$\underline{\alpha}_{\text{c}}(\underline{\alpha}_{\text{thirds}}^{\text{q}}) = c_1 c_2 \dots c_k \dots c_n$$

be the walk in $d(R_{\text{thirds}}^{\text{c}})$ such that $c_k = c(q_k)$ for all q_k in $\underline{\alpha}_{\text{thirds}}^{\text{q}}$. Similarly, let

$$\underline{\alpha}_{\text{m}}(\underline{\alpha}_{\text{thirds}}^{\text{q}}) = m_1 m_2 \dots m_k \dots m_n$$

be the walk in $d(R_{\text{thirds}}^{\text{m}})$ such that $m_k = m(q_k)$ for all q_k in $\underline{\alpha}_{\text{thirds}}^{\text{q}}$. For example if

$$\underline{\alpha}_{\text{thirds}}^{\text{q}} = \langle 5,0 \rangle \langle 0,0 \rangle \langle 2,0 \rangle \langle 4,-1 \rangle$$

then

$$\underline{\alpha}_m(\underline{\alpha}_{\text{thirds}}^q) = 5\ 0\ 2\ 4 \quad \text{and} \quad \underline{\alpha}_c(\underline{\alpha}_{\text{thirds}}^q) = 8\ 0\ 3\ 6$$

Let $\underline{A}_{\text{thirds}}^{q,\gamma}$ be the set that contains all and only those walks $\underline{\alpha}_{\text{thirds}}^q$ such that $\underline{\alpha}_{\text{thirds}}^q$, $\underline{\alpha}_c(\underline{\alpha}_{\text{thirds}}^q)$ and $\underline{\alpha}_m(\underline{\alpha}_{\text{thirds}}^q)$ are *circuits*.

Given a set of walks in a specified genus relation digraph,

$$\underline{A}^q = \{ \underline{\alpha}_1, \underline{\alpha}_2, \dots, \underline{\alpha}_k, \dots, \underline{\alpha}_{|\underline{A}^q|} \}$$

then the function $\underline{Q}(\underline{A}^q)$ returns the set that contains all and only those genus sets that are walk sets of walks in \underline{A}^q . That is,

$$\underline{Q}(\underline{A}^q) =_{\text{df}} \bigcup_{k=1}^{|\underline{A}^q|} \{ \underline{q}(\underline{\alpha}_k) \}$$

where $\underline{q}(\underline{\alpha}_k)$ is the walk set of $\underline{\alpha}_k$.

The set

$$\underline{Q}(\underline{A}_{\text{thirds}}^{q,\gamma})$$

therefore contains all and only those genus sets that are walk sets of walks in $\underline{A}_{\text{thirds}}^{q,\gamma}$. I shall now prove that

$$\underline{Q}(\underline{A}_{\text{thirds}}^{q,\gamma}) = \underline{Q}^{\text{scale}}$$

It is immediately obvious from Figure 34-3 that all circuits in $d(R_{\text{thirds}}^m)$ are of length 7. Therefore

$$(\underline{\alpha}_{\text{thirds}}^q \in \underline{A}_{\text{thirds}}^{q,\gamma}) \Rightarrow (|\underline{\alpha}_m(\underline{\alpha}_{\text{thirds}}^q)| = 7) \Rightarrow \begin{cases} |\underline{\alpha}_{\text{thirds}}^q| = 7 \\ |\underline{\alpha}_c(\underline{\alpha}_{\text{thirds}}^q)| = 7 \end{cases}$$

Also, it is clear from Figure 34-4 that any circuit of length 7 in $d(R_{\text{thirds}}^q)$ must contain exactly 3 arcs with arc interval $\langle 4,2 \rangle$ and exactly 4 arcs with arc interval $\langle 3,2 \rangle$. Consider, for example, those circuits that begin and end on genus $\langle 3,0 \rangle$ —that is, those walks that traverse the digraph between the two nodes q_i and q_t joined by a curved dotted line in Figure 34-4. These two nodes lie at the diagonally opposite corners of a rectangular region of the graph that is 3 arcs ‘high’ and 4 arcs ‘wide.’ Since every arc in Figure 34-4 is directed either up the page if its arc interval is $\langle 4,2 \rangle$ or to the right if its arc interval is $\langle 3,2 \rangle$, it is clear that any walk of length 7 that begins at q_i and ends at q_t will involve passing along three arcs with arc interval $\langle 4,2 \rangle$ and 4 arcs with arc interval $\langle 3,2 \rangle$.⁴⁵⁵

Let d be some specified pitch, chroma, morph or genus relation digraph and let

⁴⁵⁵ The fact that any circuit of length 7 in the thirds genus relation digraph must contain 3 arcs with arc interval $\langle 4,2 \rangle$ and 4 arcs with arc interval $\langle 3,2 \rangle$ can be proved algebraically. But to do so would require burdening the reader with a considerable quantity of tedious mathematical detail. In any case, the simple graphical proof given here is perfectly sound.

$$\underline{\alpha}_1 = \langle a_{1,1}, a_{1,2}, \dots, a_{1,j}, \dots, a_{1,|\underline{\alpha}_1|} \rangle \quad \underline{\alpha}_2 = \langle a_{2,1}, a_{2,2}, \dots, a_{2,k}, \dots, a_{2,|\underline{\alpha}_2|} \rangle$$

be any two walks in d . Remembering that $i(a)$ denotes the *arc interval* of a , then $\underline{\alpha}_1$ and $\underline{\alpha}_2$ are defined to be *transpositionally equivalent* if and only if $|\underline{\alpha}_1| = |\underline{\alpha}_2|$ and

$$i(a_{1,j}) = i(a_{2,j})$$

for all j such that $1 \leq j \leq |\underline{\alpha}_1|$. The *transpositional equivalence class* of walks to which any given walk $\underline{\alpha}_1$ belongs is the set that contains all and only those walks that are transpositionally equivalent to $\underline{\alpha}_1$ and it is denoted $\underline{A}_{\text{tran}}(\underline{\alpha}_1)$. For example, in Figure 34-4, the walks

$$\langle 5,0 \rangle \langle 0,0 \rangle \langle 2,0 \rangle \langle 4,-1 \rangle \quad \text{and} \quad \langle 2,0 \rangle \langle 4,0 \rangle \langle 6,0 \rangle \langle 1,-1 \rangle$$

are transpositionally equivalent.

$\underline{\alpha}_1$ and $\underline{\alpha}_2$ are defined to be *cyclically equivalent* if and only if $|\underline{\alpha}_1| = |\underline{\alpha}_2|$, $v_i(\underline{\alpha}_1) = v_i(\underline{\alpha}_2)$, $v_t(\underline{\alpha}_1) = v_t(\underline{\alpha}_2)$ and there exists some non-negative integer n such that for all $1 \leq j \leq |\underline{\alpha}_1|$,

$$a_{1,j} = a_{2,(j \bmod |\underline{\alpha}_1|) + n}$$

The *cyclic equivalence class* of walks to which any given closed walk $\underline{\alpha}_1$ belongs is the set that contains all and only those walks that are cyclically equivalent to $\underline{\alpha}_1$ and it is denoted $\underline{A}_{\text{cyc}}(\underline{\alpha}_1)$. For example, in Figure 34-4, the walks

$$\langle 3,0 \rangle \langle 5,0 \rangle \langle 0,0 \rangle \langle 2,0 \rangle \langle 4,0 \rangle \langle 6,0 \rangle \langle 1,0 \rangle \langle 3,0 \rangle \quad \text{and} \quad \langle 1,0 \rangle \langle 3,0 \rangle \langle 5,0 \rangle \langle 0,0 \rangle \langle 2,0 \rangle \langle 4,0 \rangle \langle 6,0 \rangle \langle 1,0 \rangle$$

are cyclically equivalent.

$\underline{\alpha}_1$ and $\underline{\alpha}_2$ are defined to be *cyclo-transpositionally equivalent* if and only if $|\underline{\alpha}_1| = |\underline{\alpha}_2|$, $v_i(\underline{\alpha}_1) = v_i(\underline{\alpha}_2)$, $v_t(\underline{\alpha}_1) = v_t(\underline{\alpha}_2)$ and there exists some non-negative integer n such that for all $1 \leq j \leq |\underline{\alpha}_1|$,

$$i(a_{1,j}) = i(a_{2,(j \bmod |\underline{\alpha}_1|) + n})$$

The *cyclo-transpositional equivalence class* of walks to which any given closed walk $\underline{\alpha}_1$ belongs is the set that contains all and only those walks that are cyclo-transpositionally equivalent to $\underline{\alpha}_1$ and it is denoted $\underline{A}_{\text{cytr}}(\underline{\alpha}_1)$. For example, in Figure 34-4, the walks

$$\langle 3,0 \rangle \langle 5,0 \rangle \langle 0,0 \rangle \langle 2,0 \rangle \langle 4,0 \rangle \langle 6,0 \rangle \langle 1,0 \rangle \langle 3,0 \rangle \quad \text{and} \quad \langle 6,0 \rangle \langle 1,-1 \rangle \langle 3,-1 \rangle \langle 5,0 \rangle \langle 0,-1 \rangle \langle 2,0 \rangle \langle 4,-1 \rangle \langle 6,0 \rangle$$

are cyclo-transpositionally equivalent.

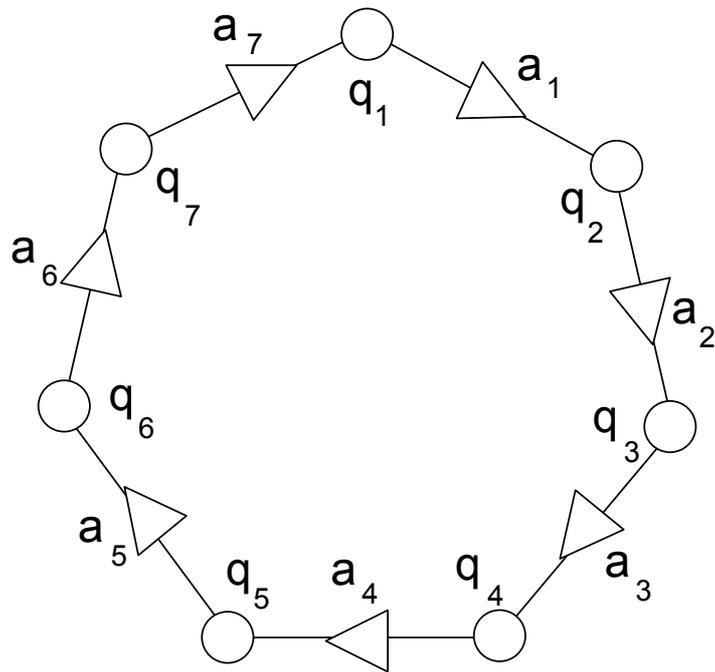


Figure 35-1

Figure 35-1 represents a circuit of length 7 in $d(R_{\text{thirds}}^q)$. As has been shown, exactly three of the seven arcs a_1 – a_7 must have an arc interval of $\langle 4,2 \rangle$ and the other

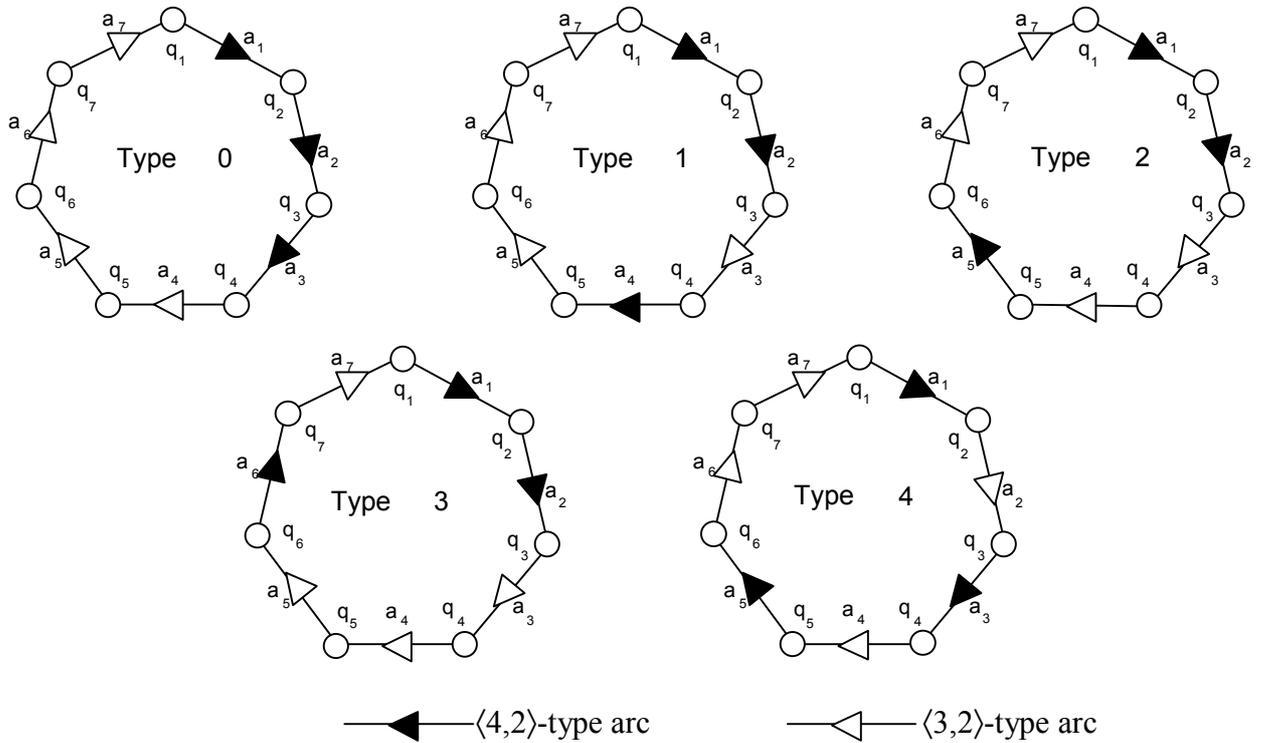


Figure 35-2

four arcs must have an arc interval of $\langle 3,2 \rangle$. The three $\langle 4,2 \rangle$ -arcs may occur consecutively, or they may be arranged so that no more than two $\langle 4,2 \rangle$ -arcs occur consecutively; or they may be arranged so that *no* $\langle 4,2 \rangle$ -arcs occur consecutively.

The ring labelled Type 0 in Figure 35-2 represents a circuit of length 7 in $d(R_{\text{thirds}}^q)$ in which the three $\langle 4,2 \rangle$ -type arcs occur consecutively. In fact, because q_1 in this diagram can be any genus whatsoever, the ring labelled Type 0 represents the cyclo-transpositional equivalence class of circuits of length 7 in $d(R_{\text{thirds}}^q)$ in which the three $\langle 4,2 \rangle$ -type arcs occur consecutively. This cyclo-transpositional equivalence class will be denoted $\underline{A}_0^{q,\text{cytr}}$. The circuit

$$\langle 3,0 \rangle \langle 5,1 \rangle \langle 0,1 \rangle \langle 2,2 \rangle \langle 4,1 \rangle \langle 6,1 \rangle \langle 1,0 \rangle \langle 3,0 \rangle$$

is an example of a circuit in $\underline{A}_0^{q,\text{cytr}}$ (see Figure 34-4).

If the three $\langle 4,2 \rangle$ -type arcs in a circuit of length 7 in $d(R_{\text{thirds}}^q)$ are arranged so that exactly two of them occur consecutively, then these two arcs may be followed only by 1, 2 or 3 consecutive $\langle 3,2 \rangle$ -type arcs. The ring labelled Type 1 in Figure 35-2 represents the cyclo-transpositional equivalence class of circuits of length 7 in $d(R_{\text{thirds}}^q)$ in which two of the three $\langle 4,2 \rangle$ -type arcs occur consecutively and are separated from the next $\langle 4,2 \rangle$ -type arc by a single $\langle 3,2 \rangle$ -type arc. The ring labelled Type 2 represents those circuits in which the pair of consecutive $\langle 4,2 \rangle$ -type arcs are followed by *two* consecutive $\langle 3,2 \rangle$ -type arcs, and the ring labelled Type 3 represents those circuits in which the pair of $\langle 4,2 \rangle$ -type arcs are followed by *three* consecutive $\langle 3,2 \rangle$ -type arcs. The cyclo-transpositional equivalence classes represented by the rings labelled Type 1, Type 2 and Type 3 will be denoted $\underline{A}_1^{q,\text{cytr}}$, $\underline{A}_2^{q,\text{cytr}}$ and $\underline{A}_3^{q,\text{cytr}}$, respectively. Finally, the ring labelled Type 4 represents the cyclo-transpositional equivalence class $\underline{A}_4^{q,\text{cytr}}$ which contains all and only circuits of length 7 in $d(R_{\text{thirds}}^q)$ in which no $\langle 4,2 \rangle$ -type arcs occur consecutively. The set that contains all and only circuits of length 7 in $d(R_{\text{thirds}}^q)$ is therefore

$$\underline{A}_{\text{thirds}}^{q,\text{circ},7} = \underline{A}_0^{q,\text{cytr}} \cup \underline{A}_1^{q,\text{cytr}} \cup \underline{A}_2^{q,\text{cytr}} \cup \underline{A}_3^{q,\text{cytr}} \cup \underline{A}_4^{q,\text{cytr}} \quad (\text{Eq.2})$$

As defined above, $\underline{A}_{\text{thirds}}^{q,\gamma}$ is the set that contains all and only those walks $\underline{\alpha}_{\text{thirds}}^q$ in $d(R_{\text{thirds}}^q)$ such that $\underline{\alpha}_{\text{thirds}}^q$, $\underline{\alpha}_c(\underline{\alpha}_{\text{thirds}}^q)$ and $\underline{\alpha}_m(\underline{\alpha}_{\text{thirds}}^q)$ are circuits. As has already been shown,

$$(\underline{\alpha}_{\text{thirds}}^q \in \underline{A}_{\text{thirds}}^{q,\gamma}) \Rightarrow (|\underline{\alpha}_{\text{thirds}}^q| = 7)$$

therefore

$$\underline{A}_{\text{thirds}}^{q,\gamma} \subseteq \underline{A}_{\text{thirds}}^{q,\text{circ},7} \quad (\text{Eq.3})$$

If $\underline{\alpha}_{\text{thirds}}^q$ is a member of $\underline{A}_0^{q,\text{cytr}}$ then it will be cyclo-transpositionally equivalent to

$$\langle 3,0 \rangle \langle 5,1 \rangle \langle 0,1 \rangle \langle 2,2 \rangle \langle 4,1 \rangle \langle 6,1 \rangle \langle 1,0 \rangle \langle 3,0 \rangle$$

and so $\underline{\alpha}_c(\underline{\alpha}_{\text{thirds}}^q)$ will be cyclo-transpositionally equivalent to the following walk in $d(R_{\text{thirds}}^c)$:

$$5 \ 9 \ 1 \ 5 \ 8 \ 11 \ 2 \ 5$$

But this walk is not a circuit because it passes three times through the chroma 5. Therefore,

$$\begin{aligned} (\underline{\alpha}_{\text{thirds}}^q \in \underline{A}_0^{q,\text{cytr}}) &\Rightarrow (\underline{\alpha}_{\text{thirds}}^q \notin \underline{A}_{\text{thirds}}^{q,\gamma}) \\ &\Rightarrow (\underline{A}_0^{q,\text{cytr}} \not\subseteq \underline{A}_{\text{thirds}}^{q,\gamma}) \end{aligned} \quad (\text{Eq.4})$$

If $\underline{\alpha}_{\text{thirds}}^q$ is a member of $\underline{A}_1^{q,\text{cytr}}$ then it will be cyclo-transpositionally equivalent to

$$\underline{\alpha}_1^q = \langle 0, -1 \rangle \langle 2, 0 \rangle \langle 4, 0 \rangle \langle 6, 0 \rangle \langle 1, 0 \rangle \langle 3, 0 \rangle \langle 5, 0 \rangle \langle 0, -1 \rangle$$

and so $\underline{\alpha}_c(\underline{\alpha}_{\text{thirds}}^q)$ will be cyclo-transpositionally equivalent to

$$11 \ 3 \ 7 \ 10 \ 2 \ 5 \ 8 \ 11$$

which is a circuit. Therefore,

$$\begin{aligned} (\underline{\alpha}_{\text{thirds}}^q \in \underline{A}_1^{q,\text{cytr}}) &\Rightarrow (\underline{\alpha}_{\text{thirds}}^q \in \underline{A}_{\text{thirds}}^{q,\gamma}) \\ &\Rightarrow (\underline{A}_1^{q,\text{cytr}} \subseteq \underline{A}_{\text{thirds}}^{q,\gamma}) \end{aligned} \quad (\text{Eq.5})$$

The walk set of

$$\underline{\alpha}_1^q = \langle 0, -1 \rangle \langle 2, 0 \rangle \langle 4, 0 \rangle \langle 6, 0 \rangle \langle 1, 0 \rangle \langle 3, 0 \rangle \langle 5, 0 \rangle \langle 0, -1 \rangle$$

is the set

$$\underline{q}_4 = \{ \langle 2, 0 \rangle, \langle 3, 0 \rangle, \langle 4, 0 \rangle, \langle 5, 0 \rangle, \langle 6, 0 \rangle, \langle 0, -1 \rangle, \langle 1, 0 \rangle \}$$

which, as mentioned above, is the genus set that contains all and only those genera in a C harmonic major scale. The set that contains all and only those genus sets that are walk sets of circuits in $\underline{A}_1^{q,\text{cytr}}$ is therefore equal to the set of all and only harmonic major scale genus sets. That is,

$$\underline{Q}(\underline{A}_1^{q,\text{cytr}}) = \underline{Q}^{\text{hama}} \quad (\text{Eq.6})$$

If $\underline{\alpha}_{\text{thirds}}^q$ is a member of $\underline{A}_2^{q,\text{cytr}}$ then it will be cyclo-transpositionally equivalent to

$$\underline{\alpha}_2^q = \langle 4, -1 \rangle \langle 6, 0 \rangle \langle 1, 0 \rangle \langle 3, 0 \rangle \langle 5, 0 \rangle \langle 0, 0 \rangle \langle 2, 0 \rangle \langle 4, -1 \rangle$$

and so $\underline{\alpha}_c(\underline{\alpha}_{\text{thirds}}^q)$ will be cyclo-transpositionally equivalent to

$$6 \ 10 \ 2 \ 5 \ 8 \ 0 \ 3 \ 6$$

which is a circuit. Therefore,

$$\begin{aligned} (\underline{\alpha}_{\text{thirds}}^q \in \underline{A}_2^{q,\text{cytr}}) &\Rightarrow (\underline{\alpha}_{\text{thirds}}^q \in \underline{A}_{\text{thirds}}^{q,\gamma}) \\ &\Rightarrow (\underline{A}_2^{q,\text{cytr}} \subseteq \underline{A}_{\text{thirds}}^{q,\gamma}) \end{aligned} \quad (\text{Eq.7})$$

The walk set of

$$\underline{\alpha}_2^q = \langle 4, -1 \rangle \langle 6, 0 \rangle \langle 1, 0 \rangle \langle 3, 0 \rangle \langle 5, 0 \rangle \langle 0, 0 \rangle \langle 2, 0 \rangle \langle 4, -1 \rangle$$

is the set

$$\underline{q}_3 = \{ \langle 2, 0 \rangle, \langle 3, 0 \rangle, \langle 4, -1 \rangle, \langle 5, 0 \rangle, \langle 6, 0 \rangle, \langle 0, 0 \rangle, \langle 1, 0 \rangle \}$$

which, as mentioned above, is the genus set that contains all and only those genera in a C melodic minor scale. The set that contains all and only those genus sets that are walk sets of circuits in $\underline{A}_2^{q, \text{cytr}}$ is therefore equal to the set of all and only melodic minor scale genus sets. That is,

$$\underline{Q}(\underline{A}_2^{q, \text{cytr}}) = \underline{Q}^{\text{memi}} \quad (\text{Eq.8})$$

If $\underline{\alpha}_{\text{thirds}}^q$ is a member of $\underline{A}_3^{q, \text{cytr}}$ then it will be cyclo-transpositionally equivalent to

$$\underline{\alpha}_3^q = \langle 4, -1 \rangle \langle 6, 0 \rangle \langle 1, 0 \rangle \langle 3, 0 \rangle \langle 5, 0 \rangle \langle 0, -1 \rangle \langle 2, 0 \rangle \langle 4, -1 \rangle$$

and so $\underline{\alpha}_c(\underline{\alpha}_{\text{thirds}}^q)$ will be cyclo-transpositionally equivalent to

$$6 \quad 10 \quad 2 \quad 5 \quad 8 \quad 11 \quad 3 \quad 6$$

which is a circuit. Therefore,

$$\begin{aligned} (\underline{\alpha}_{\text{thirds}}^q \in \underline{A}_3^{q, \text{cytr}}) &\Rightarrow (\underline{\alpha}_{\text{thirds}}^q \in \underline{A}_{\text{thirds}}^{q, \gamma}) \\ &\Rightarrow (\underline{A}_3^{q, \text{cytr}} \subseteq \underline{A}_{\text{thirds}}^{q, \gamma}) \end{aligned} \quad (\text{Eq.9})$$

The walk set of

$$\underline{\alpha}_3^q = \langle 4, -1 \rangle \langle 6, 0 \rangle \langle 1, 0 \rangle \langle 3, 0 \rangle \langle 5, 0 \rangle \langle 0, -1 \rangle \langle 2, 0 \rangle \langle 4, -1 \rangle$$

is the set

$$\underline{q}_2 = \{ \langle 2, 0 \rangle, \langle 3, 0 \rangle, \langle 4, -1 \rangle, \langle 5, 0 \rangle, \langle 6, 0 \rangle, \langle 0, -1 \rangle, \langle 1, 0 \rangle \}$$

which, as mentioned above, is the genus set that contains all and only those genera in a C harmonic minor scale. The set that contains all and only those genus sets that are walk sets of circuits in $\underline{A}_3^{q, \text{cytr}}$ is therefore equal to the set of all and only harmonic minor scale genus sets. That is,

$$\underline{Q}(\underline{A}_3^{q, \text{cytr}}) = \underline{Q}^{\text{hami}} \quad (\text{Eq.10})$$

If $\underline{\alpha}_{\text{thirds}}^q$ is a member of $\underline{A}_4^{q, \text{cytr}}$ then it will be cyclo-transpositionally equivalent to

$$\underline{\alpha}_4^q = \langle 5, 0 \rangle \langle 0, 0 \rangle \langle 2, 0 \rangle \langle 4, 0 \rangle \langle 6, 0 \rangle \langle 1, 0 \rangle \langle 3, 0 \rangle \langle 5, 0 \rangle$$

and so $\underline{\alpha}_c(\underline{\alpha}_{\text{thirds}}^q)$ will be cyclo-transpositionally equivalent to

$$8 \quad 0 \quad 3 \quad 7 \quad 10 \quad 2 \quad 5 \quad 8$$

which is a circuit. Therefore,

$$\begin{aligned} (\underline{\alpha}_{\text{thirds}}^q \in \underline{A}_4^{q, \text{cytr}}) &\Rightarrow (\underline{\alpha}_{\text{thirds}}^q \in \underline{A}_{\text{thirds}}^{q, \gamma}) \\ &\Rightarrow (\underline{A}_4^{q, \text{cytr}} \subseteq \underline{A}_{\text{thirds}}^{q, \gamma}) \end{aligned} \quad (\text{Eq.11})$$

The walk set of

$$\underline{\alpha}_4^q = \langle 5,0 \rangle \langle 0,0 \rangle \langle 2,0 \rangle \langle 4,0 \rangle \langle 6,0 \rangle \langle 1,0 \rangle \langle 3,0 \rangle \langle 5,0 \rangle$$

is the set

$$\underline{q}_1 = \{ \langle 2,0 \rangle, \langle 3,0 \rangle, \langle 4,0 \rangle, \langle 5,0 \rangle, \langle 6,0 \rangle, \langle 0,0 \rangle, \langle 1,0 \rangle \}$$

which, as mentioned above, is the genus set that contains all and only those genera in a diatonic C major scale. The set that contains all and only those genus sets that are walk sets of circuits in $\underline{A}_4^{q,\text{cytr}}$ is therefore equal to the set of all and only diatonic scale genus sets. That is,

$$\underline{Q}(\underline{A}_4^{q,\text{cytr}}) = \underline{Q}^{\text{diat}} \quad (\text{Eq.12})$$

The set of all and only circuits of length 7 in $d(R_{\text{thirds}}^q)$ was shown above to be

$$\underline{A}_{\text{thirds}}^{q,\text{circ},7} = \underline{A}_0^{q,\text{cytr}} \cup \underline{A}_1^{q,\text{cytr}} \cup \underline{A}_2^{q,\text{cytr}} \cup \underline{A}_3^{q,\text{cytr}} \cup \underline{A}_4^{q,\text{cytr}} \quad (\text{Eq.2})$$

Therefore,

$$\left. \begin{array}{l} \underline{A}_0^{q,\text{cytr}} \not\subseteq \underline{A}_{\text{thirds}}^{q,\gamma} \quad (\text{Eq.4}) \\ \underline{A}_1^{q,\text{cytr}} \subseteq \underline{A}_{\text{thirds}}^{q,\gamma} \quad (\text{Eq.5}) \\ \underline{A}_2^{q,\text{cytr}} \subseteq \underline{A}_{\text{thirds}}^{q,\gamma} \quad (\text{Eq.7}) \\ \underline{A}_3^{q,\text{cytr}} \subseteq \underline{A}_{\text{thirds}}^{q,\gamma} \quad (\text{Eq.9}) \\ \underline{A}_4^{q,\text{cytr}} \subseteq \underline{A}_{\text{thirds}}^{q,\gamma} \quad (\text{Eq.11}) \\ \underline{A}_{\text{thirds}}^{q,\gamma} \subseteq \underline{A}_{\text{thirds}}^{q,\text{circ},7} \quad (\text{Eq.3}) \end{array} \right\} \Rightarrow \underline{A}_{\text{thirds}}^{q,\gamma} = \underline{A}_1^{q,\text{cytr}} \cup \underline{A}_2^{q,\text{cytr}} \cup \underline{A}_3^{q,\text{cytr}} \cup \underline{A}_4^{q,\text{cytr}}$$

This, in turn, implies that

$$\underline{Q}(\underline{A}_{\text{thirds}}^{q,\gamma}) = \underline{Q}(\underline{A}_1^{q,\text{cytr}}) \cup \underline{Q}(\underline{A}_2^{q,\text{cytr}}) \cup \underline{Q}(\underline{A}_3^{q,\text{cytr}}) \cup \underline{Q}(\underline{A}_4^{q,\text{cytr}}) \quad (\text{Eq.13})$$

But it was shown above that

$$\underline{Q}(\underline{A}_1^{q,\text{cytr}}) = \underline{Q}^{\text{hama}} \quad (\text{Eq.6})$$

$$\underline{Q}(\underline{A}_2^{q,\text{cytr}}) = \underline{Q}^{\text{memi}} \quad (\text{Eq.8})$$

$$\underline{Q}(\underline{A}_3^{q,\text{cytr}}) = \underline{Q}^{\text{hami}} \quad (\text{Eq.10})$$

$$\underline{Q}(\underline{A}_4^{q,\text{cytr}}) = \underline{Q}^{\text{diat}} \quad (\text{Eq.12})$$

Therefore, it follows from Eq.1 and Eq.13 that

$$\begin{aligned} \underline{Q}(\underline{A}_{\text{thirds}}^{q,\gamma}) &= \underline{Q}^{\text{hama}} \cup \underline{Q}^{\text{memi}} \cup \underline{Q}^{\text{hami}} \cup \underline{Q}^{\text{diat}} \\ &= \underline{Q}^{\text{scale}} \quad (\text{Q.E.D.}) \end{aligned}$$

36 Introduction to an example of an algorithmic style theory

I shall devote the remainder of this thesis to describing some of the work that I have done towards the development of an algorithmic style theory intended to embody the knowledge required for composition of all and only those pieces in the style of a particular subset of Bach's chorale harmonizations. The theory system of this algorithmic style theory will be denoted T_{Bach} .

The universal set of scores of T_{Bach} , denoted $\underline{s}_u(T_{\text{Bach}})$, is the universal set of Standard Notation scores as defined in section 3.3 above.

A score is defined to be a member of the corpus kernel of the theory $\underline{s}_k(T_{\text{Bach}})$ if and only if it occurs in a copy of Klaus Schubert's edition of Bach's *371 Four-Part Chorales* (Bach 1990) and satisfies either conditions A1–A4 or conditions B1–B4 where conditions A1–A4 are as follows:

- A1. The score has a time signature of 3/4 throughout.
- A2. Either:
 - i. the score begins with either a complete bar or an incomplete bar consisting of a one-crotchet anacrusis and has 16 complete bars including the final bar; or
 - ii. the score begins with an incomplete bar consisting of a one-crotchet anacrusis, has 15 complete bars and ends with an incomplete bar of two crotchets' duration.
- A3. The score has four phrases, the end of each being marked by a fermata symbol.
- A4. The score contains no repeat marks.

Conditions B1–B4 are as follows:

- B1. The score has a time-signature of **C** or **♩** throughout.
- B2. Either:
 - i. the score begins with either a complete bar or an incomplete bar consisting of a one-crotchet anacrusis and has 8 complete bars including the final bar; or
 - ii. the score begins with an incomplete bar consisting of a one-crotchet anacrusis, has 7 complete bars and ends with an incomplete bar of three crotchets' duration.
- B3. The score has four phrases, the end of each being marked by a fermata symbol.
- B4. The score contains no repeat marks.

Scores that satisfy conditions A1–A4 are prosodically closely related to scores that satisfy conditions B1–B4 in that any verse stanza whose metric structure is such that it could appropriately be set to a score satisfying conditions A1–A4 could also be set to a score satisfying conditions B1–B4. For example, Bach composed at least two settings of the text 'Ach Gott, wie manches Herzeleid' of which one, BWV 3/6 (no.156

12	‘Puer Natus in Bethlehem’	BWV 65/2
53	‘Das neugeborne Kindelein’	BWV 122/6
93	‘Wach auf, mein Herz’	BWV 194/12
164	‘Herr Gott, dich loben alle wir’	BWV 326
176	‘Erstanden ist der heil’ge Christ’	BWV 306
188	‘Ich dank dir schon durch deinen Sohn’	BWV 349
207	‘Des heil’gen Geistes reiche Gnad’	BWV 295
217	‘Ach Gott, wie manches Herzeleid’	BWV 153/9
257	‘Nun laßt uns Gott, dem Herren’	BWV 194/12 ⁴⁵⁶
334	‘Für deinen Thron tret ich hiermit’	BWV 327 ⁴⁵⁷

Figure 36-1

in Bach 1990), satisfies conditions B1–B4 while the other, BWV 153/9 (no.217 in Bach 1990), satisfies conditions A1–A4.

The table in Figure 36-1 lists all and only those scores in Bach 1990 that satisfy conditions A1–A4 and the table in Figure 36-2 lists all and only those scores in Bach 1990 that satisfy conditions B1–B4.

The corpus of the theory system, $\underline{s}_c(T_{\text{Bach}})$, which by definition contains all and only those scores that are notationally equivalent to at least one of the scores in $\underline{s}_k(T_{\text{Bach}})$, is therefore equal to the union of 29 notational equivalence classes.

The acceptability algorithm of T_{Bach} , $\alpha(T_{\text{Bach}})$, is defined to be equal to any acceptability algorithm that satisfies the specification given in section 3.8 above.

The style of T_{Bach} , $\underline{s}_s(T_{\text{Bach}})$, is therefore the set of scores that contains all and only those scores s such that s is a member of the corpus $\underline{s}_c(T_{\text{Bach}})$ or s is determined by the acceptability algorithm $\alpha(T_{\text{Bach}})$ to be a member of the universal set of acceptable scores $\underline{s}_a(T_{\text{Bach}})$.

The representation algorithm of T_{Bach} , $\rho(T_{\text{Bach}})$, will be defined in chapter 38 below and the composing algorithm, $\gamma(T_{\text{Bach}})$, will be discussed in chapter 39 below. This composing algorithm has been fully implemented in a working computer program called IOTA, written in Lisp and tested on a Macintosh computer. The derivation algorithm of T_{Bach} , $\delta(T_{\text{Bach}})$, will also be described in chapter 39. It is essentially the same as the composing algorithm and has also been computationally implemented in IOTA.

The algorithmic style theory associated with the theory system T_{Bach} , is the hypothesis that

$$\underline{s}_w(T_{\text{Bach}}) = \underline{s}_s(T_{\text{Bach}})$$

⁴⁵⁶ 257 is the same as 93 except that 257 contains trills (which aren’t represented by the representation algorithm used in this theory) and the final bar of 93 is complete whereas the final bar of 257 is only two crotchets’ long.

⁴⁵⁷ 334 is closely related to 164–164 is essentially the same except it has a bass line ‘divided’ into quavers.

6	‘Christus, der ist mein Leben’	BWV 281
46	‘Vom Himmel hoch, da komm ich her’	BWV 248/9
72	‘Erhalt uns, Herr, bei deinem Wort’	BWV 6/6
136	‘Herr Jesu Christ, dich zu uns wend’	BWV 332
154	‘Der du bist drei in Einigkeit’	BWV 293
156	‘Ach Gott, wie manches Herzeleid’	BWV 3/6
157	‘Wo Gott zum Haus nicht gibt sein’ Gunst’	BWV 438
170	‘Nun komm, der Heiden Heiland’	BWV 62/6
180	‘Als Jesus Christus in der Nacht’	BWV 265
185	‘Nun freut euch, Gottes Kinder all’	BWV 387
187	‘Komm, Gott Schöpfer, heiliger Geist’	BWV 370
189	‘Herr Jesu Christ, wahr Mensch und Gott’	BWV 336
224	‘Das walt Gott Vater und Gott Sohn’	BWV 290
236	‘O Jesu, du mein Bräutigam’	BWV 335 ⁴⁵⁸
240	‘Nun sich der Tag geendet hat’	BWV 396
245	‘Christe, der du bist Tag und Licht’	BWV 274
247	‘Wenn wir in höchsten Nöten sein’	BWV 432
295	‘Herr Jesu Christ, mein’s Lebens Licht’	BWV 335 ⁴⁵⁹
344	‘Vom Himmel hoch, da komm ich her’	BWV 248/23

Figure 36-2

where $\underline{s}_w(T_{\text{Bach}})$ is the universal set of well-formed scores—that is, the set of all and only those scores that are mapped by the representation algorithm $\rho(T_{\text{Bach}})$ onto representations that are members of the universal output set of the composing algorithm $\gamma(T_{\text{Bach}})$.

Unfortunately, this algorithmic style theory has been refuted. The universal set of well-formed scores $\underline{s}_w(T_{\text{Bach}})$ has been shown to contain 22 of the 29 notational equivalence classes of scores in the corpus $\underline{s}_c(T_{\text{Bach}})$ but has also been shown to contain unacceptable scores. That is, the composing algorithm $\gamma(T_{\text{Bach}})$ has been shown to overgenerate and has not yet been shown to contain the corpus. I shall suggest below some ways in which it might be possible to modify the composing algorithm to produce a new algorithmic style theory that is not so easily refuted.

No parsing algorithm will be defined. I was able to produce a well-formed derivation for one score from each notational equivalence class in the corpus so far tested merely by inspection and study of the scores in the corpus. Of course, if I had not been able to find a well-formed derivation for any member of the corpus, then I would have had to define a parsing algorithm. But in the event, the composing algorithm of T_{Bach} was shown to overgenerate even before any extensive testing for undergeneration had been carried out.

Also, no score algorithm will be defined for T_{Bach} . I made a brief attempt to produce a computer program that generates physical Standard Notation scores automatically from representations of the type generated by $\rho(T_{\text{Bach}})$ and this attempt was sufficient to convince me that a representation of the type generated by the

⁴⁵⁸ 236 differs only in minor detail from 295.

⁴⁵⁹ See footnote 458.

representation algorithm and the composing algorithm of T_{Bach} is complete enough for a Standard Notation score to be algorithmically derived from it. However, in this attempt to produce a computational implementation of a score algorithm, I encountered a number of purely technical programming problems connected with printing graphics and rendering graphics on the computer screen and I could not afford the time at that stage in my research which would have been required to solve these problems.

37 Introduction to the representation algorithm of T_{Bach}

37.1 Introduction

The definition of the representation algorithm of an algorithmic style theory must be sufficiently complete and precise for there to be no doubt as to whether or not any given representation is the correct representation of any given score. As discussed in section 6.1 above, this suggests that a representation algorithm should be implementable as a working computer system that generates an appropriate computer data-structure automatically from a scanned image of a physical score. Unfortunately, the problem of generating computational representations from scanned images of musical scores is still unsolved⁴⁶⁰ and the development of a complete solution was deemed beyond the scope of this project. The definition given below is therefore not sufficient to serve as a detailed specification for a computationally implementable representation algorithm. However, it *is* intended to be sufficiently precise and complete for there to be one and only one correct representation for each Standard Notation score and for it to be possible in every case for a human to obtain this representation directly from the score using only the definition given below and ‘without any exercise of intelligence.’⁴⁶¹

An object is defined to be a representation in the theory system T_{Bach} if and only if it is derivable from a Standard Notation score using the definition given below. This definition therefore generatively specifies the universal set of representations of T_{Bach} , $\underline{r}_u(T_{\text{Bach}})$. $\underline{r}_u(T_{\text{Bach}})$ was defined so that the representations that it contains are particularly appropriate for the corpus of chorale harmonizations, $\underline{s}_c(T_{\text{Bach}})$. However, I believe that the definition given below could profitably be employed with only minor modification in almost any algorithmic style theory intended to account for a tonal musical style. In particular, I have made some effort to ensure that the representations in $\underline{r}_u(T_{\text{Bach}})$ can be used not only to represent the two-stave, four-voice, keyboard scores in the corpus of T_{Bach} but also scores for any combination of instruments. Also, although the scores in $\underline{s}_c(T_{\text{Bach}})$ are all in either 3/4-time or common-time by definition, the representations in $\underline{r}_u(T_{\text{Bach}})$ can be used to represent scores in which the metric structure—which may take any conceivable value—changes with any frequency whatsoever.

37.2 The concepts of a location, a segment, a segmentation and a structure

37.2.1 Location

A score implies a sequence of actions that should be taken by a performer in order to produce a performance of the piece of music represented by the score. If the score indicates that several actions are to be carried out simultaneously (for example, several keyboard keys are to be pressed simultaneously) then the *locations* of the representations in the score of these actions are identical. Given two actions A_1 and A_2 then if the score indicates that A_2 is to be carried out after A_1 , then the location of the

⁴⁶⁰ For discussions of the problems involved in automatically recognizing scores, see Alphonse et al. 1988, Carter and Bacon 1990, Clarke et al., 1990.

⁴⁶¹ Borowski and Borwein 1989, 13, entry for ‘algorithm’.

representation of A_2 in the score is defined to be *greater than* that of the representation in the score of action A_1 and the location of the representation of the action A_1 is defined to be *less than* that of A_2 .

A location is completely defined by giving the bar number of the bar in which the location occurs, called the *bar number*, and the fraction of that bar that precedes the location in question. This fraction is called the *bar fraction*.⁴⁶² A *location* is therefore defined to be an ordered pair

$$l = \langle b(l), \phi(l) \rangle$$

where the bar number $b(l)$ is a natural number, and the bar fraction, $\phi(l)$, is a non-negative rational number less than one. The first and last bars in a score are treated as complete even if they are written as incomplete. For all locations l in the first bar of a score,

$$b(l) = 1$$

For example, the location of the first notes in the score of the chorale ‘Christus, der ist mein Leben’ (BWV 281, no.6 in Bach 1990) is $\langle 1, \langle 3, 4 \rangle \rangle$ (not $\langle 0, \langle 3, 4 \rangle \rangle$.)

It will be seen later that it proves useful when representing metric structure to impose a further constraint on the representation of locations. This constraint is that the location bar fraction should be expressed in *least denominator form*. For any location l , it is therefore possible to speak of *the* (unique) *location bar fraction numerator*, $v(\phi(l))$, and *the* (unique) *location bar fraction denominator*, $\delta(\phi(l))$, where these values are understood to be the numerator and denominator when the location bar fraction is expressed in its least denominator form.

As stated above, given two distinct locations within the same piece, one location can be said to be *greater than* the other. Given two distinct locations, l_1 and l_2 , then

$$l_1 > l_2 \Leftrightarrow_{\text{df}} (b(l_1) + \phi(l_1)) > (b(l_2) + \phi(l_2))$$

This allows for the *location interval* between any two locations, $\Delta(l_1, l_2)$, to be defined. The location interval is itself an ordered pair in which the first member of the pair is a non-negative integer called the *bar interval* and the second member of the pair is a non-negative rational number less than one, called the *bar fraction interval*. Given two locations, l_1 and l_2 , such that $l_2 \geq l_1$ and given that

$$l_1 = \langle b_1, \langle v_1, \delta_1 \rangle \rangle \quad \text{and} \quad l_2 = \langle b_2, \langle v_2, \delta_2 \rangle \rangle$$

then finding the location interval between these two locations involves first finding the integer z defined as follows:

$$z = \text{int} \left(\frac{v_2}{\delta_2} - \frac{v_1}{\delta_1} \right)$$

The location interval between l_1 and l_2 is then:

$$\Delta(l_1, l_2) = \Delta(l_2, l_1) = \langle b_2 - b_1 + z, \langle \delta_1 v_2 - \delta_2 v_1 - z \delta_1 \delta_2, \delta_1 \delta_2 \rangle \rangle$$

where the bar fraction interval is expressed in least denominator form.

⁴⁶² Bars are numbered as though repeat marks were not present.

37.2.2 Segment

Any two distinct locations l_1 and l_2 where $l_1 < l_2$ can be used to define a *segment* \underline{l} which is the set of all locations l such that $l_1 \leq l < l_2$. l_1 would then be the *initial location* of \underline{l} , and l_2 would be the *terminal location* of \underline{l} . These are denoted as follows:

$$l_i(\underline{l}) = l_1 \text{ and } l_t(\underline{l}) = l_2$$

Note, therefore, that for any segment \underline{l} ,

$$l_i(\underline{l}) \notin \underline{l} \text{ and } l_t(\underline{l}) \in \underline{l}$$

The terminal and initial locations of a segment taken together are called the *boundary locations* of the segment. A segment can be represented as an ordered pair:

$$\underline{l} = \langle l_i(\underline{l}), l_t(\underline{l}) \rangle$$

Each representation has a *universal segment*,

$$\underline{l}^u = \langle l_i(\underline{l}^u), l_t(\underline{l}^u) \rangle$$

which is that segment that satisfies the following four conditions:

1. $l_i(\underline{l}^u) = \langle 1, \langle 0, 1 \rangle \rangle$;
2. if l is any location in the score in question then $l \in \underline{l}^u$;
3. $\phi(l_t(\underline{l}^u)) = \langle 0, 1 \rangle$;
4. $b(l_t(\underline{l}^u)) = n + 1$ where n is the bar number of the last bar in the score.

Given two segments \underline{l}_1 and \underline{l}_2 , then \underline{l}_2 is a *subsegment* of \underline{l}_1 (and \underline{l}_1 is a *supersegment* of \underline{l}_2) if and only if the initial location of \underline{l}_2 is greater than or equal to that of \underline{l}_1 and the terminal location of \underline{l}_2 is less than or equal to that of \underline{l}_1 . That is,

$$(\underline{l}_2 \subseteq \underline{l}_1) \Leftrightarrow_{\text{df}} (l_i(\underline{l}_2) \geq l_i(\underline{l}_1)) \wedge (l_t(\underline{l}_2) \leq l_t(\underline{l}_1))$$

Similarly, \underline{l}_2 is a *proper subsegment* of \underline{l}_1 (and \underline{l}_1 is a *proper supersegment* of \underline{l}_2) if and only if \underline{l}_2 is a subsegment of \underline{l}_1 and it is not equal to \underline{l}_1 . That is,

$$(\underline{l}_2 \subset \underline{l}_1) \Leftrightarrow_{\text{df}} (\underline{l}_2 \subseteq \underline{l}_1) \wedge (\underline{l}_2 \neq \underline{l}_1)$$

A segment can never equal the empty set since every segment contains its initial location.

The *duration* of a segment is defined and denoted

$$\Delta(\underline{l}) = \Delta(l_t(\underline{l}), l_i(\underline{l}))$$

That is, the duration of a segment is the location interval between its initial and terminal locations. The first member of a duration is called the *bar duration* and the second member of a duration is called the *bar fraction duration*.

37.2.3 Segmentation

An object is a *segmentation* if and only if it is an ordered set of segments,

$$\underline{A} = \langle \underline{l}_1, \underline{l}_2, \dots, \underline{l}_k, \dots, \underline{l}^t \rangle$$

such that:

1. $l_i(\underline{l}_{k+1}) = l_t(\underline{l}_k)$ for all $\underline{l}_k, \underline{l}_{k+1} \in \underline{A}$
2. $l_i(\underline{l}_1) = l_i(\underline{l}^u) = \langle 1, \langle 0, 1 \rangle \rangle$
3. $l_t(\underline{l}^t) = l_t(\underline{l}^u)$

The last segment in a segmentation \underline{l}^t is called the *terminal segment* of the segmentation and it can be denoted with a superscript ‘t’ as shown. The first segment in a segmentation is called the *initial segment* of the segmentation.

37.2.4 Structure

In the context of a representation in T_{Bach} , a *structure*

$$\underline{\nu} = \langle n_1, n_2, \dots, n_k, \dots, n^t \rangle$$

is an ordered set of *nodes*, a *node* being an ordered pair

$$n = \langle \underline{l}(n), \lambda(\underline{l}(n)) \rangle$$

in which the first member of the pair is a segment called the *node segment* of n and the second member of the pair is some specified function of $\underline{l}(n)$ called the *node attribute* of n . Note that

$$l_i(n) =_{\text{df}} l_i(\underline{l}(n)) \quad \text{and} \quad l_t(n) =_{\text{df}} l_t(\underline{l}(n)) \quad \text{and} \quad \lambda(n) =_{\text{df}} \lambda(\underline{l}(n))$$

An object is defined to be a *structure* (in the context of a representation in T_{Bach}),

$$\underline{\nu} = \langle n_1, n_2, \dots, n_k, \dots, n^t \rangle$$

if and only if it satisfies the following conditions:

1. $\lambda(n_k) \neq \lambda(n_{k+1})$ for all $n_k, n_{k+1} \in \underline{\nu}$;
2. $l_i(n_{k+1}) = l_t(n_k)$ for all $n_k, n_{k+1} \in \underline{\nu}$;
3. $l_i(n_1) = l_i(\underline{l}^u)$;
4. $l_t(n^t) = l_t(\underline{l}^u)$.

38 Definition of a representation in T_{Bach}

An object r is a *representation* in T_{Bach} if and only if it is an ordered triple as follows:

$$r = \langle \underline{v}_r(r), \underline{\Omega}(r), \underline{\Sigma}(r) \rangle$$

where $\underline{v}_r(r)$ is the *repeat structure* of r , $\underline{\Omega}(r)$ is the *voice set* of r and $\underline{\Sigma}(r)$ is the *staff set* of r . I shall now define these three elements of a representation and specify how each is to be derived from a score.

38.1 Repeat structure

The *repeat structure* of a representation r is denoted

$$\underline{v}_r(r) = \langle n_{r,1}(r), n_{r,2}(r), \dots, n_{r,k}(r), \dots, n_{r,t}(r) \rangle$$

It is a structure in which each node $n_{r,k}(r)$ is a *repeat node*. Each repeat node

$$n_{r,k}(r) = \langle \underline{l}_{r,k}(r), \underline{h}(\underline{l}_{r,k}(r)) \rangle$$

is a node in which the node segment $\underline{l}_{r,k}(r)$ is a *repeat node segment* and the node attribute $\underline{h}(\underline{l}_{r,k}(r))$ is the *repeat order set* of repeat node $n_{r,k}(r)$. Note that

$$\underline{h}_k(r) =_{\text{df}} \underline{h}(n_{r,k}(r)) =_{\text{df}} \underline{h}(\underline{l}_{r,k}(r))$$

The repeat structure of a representation can be derived from a score using the repeat marks and instructions such as ‘dal segno,’ ‘da capo’ and ‘Coda.’ The repeat structure gives the order in which each repeat node segment is to be performed. The repeat order set of a repeat node is an ordered set of natural numbers,

$$\underline{h}_k(r) = \langle h_{k,1}(r), h_{k,2}(r), \dots, h_{k,j}(r), \dots, h_{k,|\underline{h}_k(r)|}(r) \rangle$$

such that $h_{k,j}(r) < h_{k,j+1}(r)$ for all $1 \leq j < |\underline{h}_k(r)| - 1$. Each element $h_{k,j}(r)$ in a repeat order set is a *repeat order number*.

For example, the repeat structure of the chorale, ‘Dir, dir, Jehova will ich singen’ (BWV 299, no.209 in Bach 1990) is as follows:

$$\begin{aligned} &\langle \\ &\langle \langle \langle 1, \langle 0, 1 \rangle \rangle, \langle 1, \langle 2, 3 \rangle \rangle \rangle, \langle 1 \rangle \rangle, \\ &\langle \langle \langle 1, \langle 2, 3 \rangle \rangle, \langle 9, \langle 2, 3 \rangle \rangle \rangle, \langle 2, 3 \rangle \rangle, \\ &\langle \langle \langle 9, \langle 2, 3 \rangle \rangle, \langle 18, \langle 0, 1 \rangle \rangle \rangle, \langle 4 \rangle \rangle \\ &\rangle \end{aligned}$$

That is, to perform this chorale, one must first play repeat segment,

$$\langle \langle 1, \langle 0, 1 \rangle \rangle, \langle 1, \langle 2, 3 \rangle \rangle \rangle$$

then play repeat segment

$$\langle \langle 1, \langle 2, 3 \rangle \rangle, \langle 9, \langle 2, 3 \rangle \rangle \rangle$$

twice, and finally play repeat segment

$$\langle\langle 9, \langle 2, 3 \rangle \rangle, \langle 18, \langle 0, 1 \rangle \rangle\rangle$$

Note that in cases where there are first and second time endings, the first time ending must have its own repeat node. Note also that for any pair of consecutive repeat nodes, $n_{r,k}(r), n_{r,k+1}(r) \in \underline{v}_r(r)$, the following must *never* be true:

$$|\underline{h}_k(r)| = |\underline{h}_{k+1}(r)| = 1$$

The repeat structure of ‘Herr Gott, dich loben wir’ (BWV 328, no.205 in Bach 1990) provides a more complex example:

$$\begin{aligned} &\langle \\ &\langle\langle 1, \langle 0, 1 \rangle \rangle, \langle 5, \langle 3, 4 \rangle \rangle\rangle, && \langle 1 \rangle, \\ &\langle\langle 5, \langle 3, 4 \rangle \rangle, \langle 9, \langle 3, 4 \rangle \rangle\rangle, && \langle 2, 3, 4 \rangle, \\ &\langle\langle 9, \langle 3, 4 \rangle \rangle, \langle 11, \langle 3, 4 \rangle \rangle\rangle && \langle 5, 6 \rangle \\ &\langle\langle 11, \langle 3, 4 \rangle \rangle, \langle 14, \langle 3, 4 \rangle \rangle\rangle, && \langle 7 \rangle \\ &\langle\langle 14, \langle 3, 4 \rangle \rangle, \langle 18, \langle 3, 4 \rangle \rangle\rangle, && \langle 8, 9, 10, 11, 12, 13 \rangle \\ &\langle\langle 18, \langle 3, 4 \rangle \rangle, \langle 22, \langle 3, 4 \rangle \rangle\rangle, && \langle 14, 15, 16, 17, 18, 19 \rangle \\ &\langle\langle 22, \langle 3, 4 \rangle \rangle, \langle 38, \langle 3, 4 \rangle \rangle\rangle, && \langle 20 \rangle \\ &\langle\langle 38, \langle 3, 4 \rangle \rangle, \langle 42, \langle 3, 4 \rangle \rangle\rangle, && \langle 21, 22, 23 \rangle \\ &\langle\langle 42, \langle 3, 4 \rangle \rangle, \langle 50, \langle 0, 1 \rangle \rangle\rangle, && \langle 24 \rangle \\ &\rangle \end{aligned}$$

The *repeat segmentation* of a representation r can be derived from the repeat structure of r and is defined to be the segmentation,

$$\underline{\Delta}_r(r) = \langle \underline{1}_{r,1}(r), \underline{1}_{r,2}(r), \dots, \underline{1}_{r,k}(r), \dots, \underline{1}_{r,t}(r) \rangle$$

where $\underline{1}_{r,k}(r)$ is the node segment of repeat node $n_{r,k}(r)$.

38.2 Voice set

The *voice set* of a representation r in T_{Bach} is denoted

$$\underline{\Omega}(r) = \langle V_1(r), V_2(r), \dots, V_k(r), \dots, V_t(r) \rangle$$

It contains all and only *voices* $V_k(r)$ for a single score. The voice set is an ordered set in which the voices are ordered according to the following procedure. First, remembering that

$$(p_1 < p_2) \Leftrightarrow_{\text{df}} (p_c(p_1) < p_c(p_2)) \vee ((p_c(p_1) = p_c(p_2)) \wedge (p_m(p_1) < p_m(p_2)))$$

where p_1 and p_2 are any two pitches, the voices $V_k(r)$ are first sorted in descending order of first pitch. In general, this imposes only a *partial* ordering on the voices as more than one voice may begin with a given pitch. The resulting partially ordered set of voices is therefore sorted again in descending order of second pitch, then in descending order of third pitch and so on until a total ordering has been imposed on the voices. If such an ordering cannot be imposed then the voice set is one of the possible partial orderings.

38.2.1 Voice

An object is a *voice* in a representation r in the theory system T_{Bach} if and only if it is a quadruple as follows

<i>STAFF</i>	<i>STEM DIRECTION</i>	<i>VOICE</i>
UPPER	UPWARDS	SOPRANO
UPPER	DOWNWARDS	ALTO
LOWER	UPWARDS	TENOR
LOWER	DOWNWARDS	BASS

Figure 38-1

$$V_k(r) = \langle \underline{v}_{n,k}(r), \underline{v}_{m,k}(r), \underline{v}_{s,k}(r), \underline{\Delta}_{p,k}(r) \rangle$$

where $\underline{v}_{n,k}(r)$ is the *note structure* of voice $V_k(r)$, $\underline{v}_{m,k}(r)$ is the *metre structure* of $V_k(r)$, $\underline{v}_{s,k}(r)$ is the *staff structure* of $V_k(r)$ and $\underline{\Delta}_{p,k}(r)$ is the *phrase segmentation* of $V_k(r)$. Each of these four concepts will be defined in more detail below. It is only possible to derive the voices in a representation from a score if the individual voices are *explicitly* represented in the score. That is, it is possible to derive a representation of the type defined in this section from a Standard Notation score only if it is possible at each location in the score to determine algorithmically whether or not each voice has a pitch or a rest and, if it has a pitch, what pitch this is.

Each of the chorale scores in the corpus of T_{Bach} is constructed from four voices (soprano, alto, tenor and bass). The voice to which a note in such a score belongs can be determined unambiguously from the direction of the note-stem and the staff on which it is written (see Figure 38-1).

At locations in a score in $\underline{s}_c(T_{\text{Bach}})$ where there are simultaneous notes without stems, the voice to which each note belongs can be determined either from explicit voice-leading marks in the score (e.g. BWV 153/1, no.3 in Bach 1990, locations $\langle 7, \langle 1, 2 \rangle \rangle$ and $\langle 8, \langle 0, 1 \rangle \rangle$), or from the relative heights of the notes on the staves, the higher note on the upper staff being a soprano note, the lower an alto note and so on.

38.2.1.1 Note structure

The *note structure* of voice $V_k(r)$ in representation r in the theory system T_{Bach} is denoted

$$\underline{v}_{n,k}(r) = \langle n_{n,k,1}(r), n_{n,k,2}(r), \dots, n_{n,k,j}(r), \dots, n_{n,k,t}(r) \rangle$$

It is a structure in which each node $n_{n,k,j}(r)$ is a *note node*. Each note node,

$$n_{n,k,j}(r) = \langle \underline{l}_{n,k,j}(r), \underline{p}_n(\underline{l}_{n,k,j}(r)) \rangle$$

is a node in which the node segment $\underline{l}_{n,k,j}(r)$ is a *note node segment* and the node attribute $\underline{p}_n(\underline{l}_{n,k,j}(r))$ is a pitch set called the *note node pitch set* of note node $n_{n,k,j}(r)$.

The following abbreviations can be used:

$$\underline{p}_{n,k,j}(r) =_{\text{df}} \underline{p}_n(n_{n,k,j}(r)) =_{\text{df}} \underline{p}_n(\underline{l}_{n,k,j}(r))$$

Given that representation r is the representation of score s , then each note node $\underline{n}_{n,k,j}(r)$ will be associated with one and only one of the following combinations of symbols in voice $V_k(r)$ in score s :

1. an untied note;
2. a sequence of one or more consecutive rests, preceded by either a note or the beginning of the score, and followed by either a note or the end of the score;
3. a sequence of consecutive tied notes.

Also, for each instance of one of these three classes of symbol combination in a voice $V_k(r)$ in score s there must exist one and only one note node $\underline{n}_{n,k,j}(r)$ in representation r .

If the symbol combination associated with note node $\underline{n}_{n,k,j}(r)$ is an untied note x , then:

1. $l_i(\underline{n}_{n,k,j}(r))$ will be equal to the onset location of x ;
2. $l_t(\underline{n}_{n,k,j}(r))$ will be equal to the offset location of x ;
3. $p_n(\underline{n}_{n,k,j}(r))$ will be equal to a set containing only the pitch of x which can be derived from x via the A.S.A. pitch name of x as described in chapter 21 above.

If the symbol combination associated with note node $\underline{n}_{n,k,j}(r)$ is a sequence of one or more consecutive rests, preceded by either a note y_1 or the beginning of the score, and followed by either a note y_2 or the end of the score, then:

1. $l_i(\underline{n}_{n,k,j}(r))$ will be equal to the offset location of y_1 or $\langle 1, \langle 0, 1 \rangle \rangle$;
2. $l_t(\underline{n}_{n,k,j}(r))$ will be equal to the onset location of y_2 or $l_t(l^u)$;
3. $p_n(\underline{n}_{n,k,j}(r))$ will be equal to \emptyset , the empty set.

If the symbol combination associated with note node $\underline{n}_{n,k,j}(r)$ is a sequence w of consecutive tied notes whose first note is z_{start} and whose final note is z_{end} , then:

1. $l_i(\underline{n}_{n,k,j}(r))$ will be equal to the onset location of z_{start} ;
2. $l_t(\underline{n}_{n,k,j}(r))$ will be equal to the offset location of z_{end} ;
3. $p_n(\underline{n}_{n,k,j}(r))$ will be equal to a set that contains only the pitch of z_{start} (which, since w is a sequence of *tied* notes, will be equal to the pitch of every other note in w). The pitch of z_{start} can be derived from z_{start} via the A.S.A. pitch name of z_{start} as described in section chapter 21 above.

Figure 38-2 shows bars 5 and 6 of the chorale ‘Christus, der ist mein Leben’ (BWV 281, no.6 in Bach 1990). Given that r_6 is the representation of the complete score of this chorale in T_{Bach} , then the alto voice in this score would be represented by $V_2(r_6)$. The note structure for the alto voice in this chorale would therefore be denoted $\underline{v}_{n,2}(r_6)$. The initial location of the note node in $\underline{v}_{n,2}(r_6)$ which corresponds to the untied $C\sharp_5$ semiquaver in the alto voice at location $\langle 6, \langle 3, 8 \rangle \rangle$ would be $\langle 6, \langle 3, 8 \rangle \rangle$. The terminal location of this note node would be $\langle 6, \langle 7, 16 \rangle \rangle$ and the note node pitch set for this note node would be $\{\langle 51, 30 \rangle\}$. The complete note node corresponding to this note would be

$$n_{n,2,21}(r_6) = \langle \langle \langle 6, \langle 3, 8 \rangle \rangle, \langle 6, \langle 7, 16 \rangle \rangle \rangle, \{\langle 51, 30 \rangle\}$$

The note node that represents the rest in the alto voice at location $\langle 5, \langle 1, 2 \rangle \rangle$ is $n_{n,2,18}(r_6)$. The initial location of this note node is

$$l_i(\underline{n}_{n,2,18}(r_6)) = \langle 5, \langle 1, 2 \rangle \rangle$$

The terminal location is

$$l_t(\underline{n}_{n,2,18}(r_6)) = \langle 5, \langle 3, 4 \rangle \rangle$$

and the note node pitch set is

$$\underline{p}_n(\underline{n}_{n,2,18}(r_6)) = \emptyset$$

The note node in $\underline{v}_{n,2}(r_6)$ that represents the sequence of two tied $A\sharp_4$ s that begins at location $\langle 5, \langle 3, 4 \rangle \rangle$ would be $n_{n,2,19}(r_6)$. This note node would be as follows:

$$n_{n,2,19}(r_6) = \langle \langle \langle 5, \langle 3, 4 \rangle \rangle, \langle 6, \langle 1, 8 \rangle \rangle \rangle, \{\langle 48, 28 \rangle\}$$

For each note structure in a representation r there exists a unique *note segmentation* which can be derived directly from its corresponding note structure. Given that

$$\underline{n}_{n,k,j}(r) = \langle \underline{l}_{n,k,j}(r), \underline{p}_{n,k,j}(r) \rangle,$$

then the note segmentation in r that corresponds to the note structure

$$\underline{v}_{n,k}(r) = \langle \underline{n}_{n,k,1}(r), \underline{n}_{n,k,2}(r), \dots, \underline{n}_{n,k,j}(r), \dots, \underline{n}_{n,k,t}(r) \rangle$$



Figure 38-2

is

$$\underline{\Lambda}_{n,k}(r) = \langle \underline{1}_{n,k,1}(r), \underline{1}_{n,k,2}(r), \dots, \underline{1}_{n,k,j}(r), \dots, \underline{1}_{n,k,t}(r) \rangle$$

38.2.1.2 Metre structure

The *metre structure* of voice $V_k(r)$ in representation r in the theory system T_{Bach} is denoted

$$\underline{v}_{m,k}(r) = \langle n_{m,k,1}(r), n_{m,k,2}(r), \dots, n_{m,k,j}(r), \dots, n_{m,k,t}(r) \rangle$$

It is a structure in which each node $n_{m,k,j}(r)$ is a *metre node*. Each metre node,

$$n_{m,k,j}(r) = \langle \underline{1}_{m,k,j}(r), \underline{\mu}(\underline{1}_{m,k,j}(r)) \rangle$$

is a node in which the node segment $\underline{1}_{m,k,j}(r)$ is a *metre node segment*, and the node attribute $\underline{\mu}(\underline{1}_{m,k,j}(r))$ is an ordered set called the *metric length set* of metre node $n_{m,k,j}(r)$. The following abbreviations can be used:

$$\underline{\mu}_{k,j}(r) =_{\text{df}} \underline{\mu}(n_{m,k,j}(r)) =_{\text{df}} \underline{\mu}(\underline{1}_{m,k,j}(r))$$

Each metric length set is an ordered set of *metric lengths*:

$$\underline{\mu}_{k,j}(r) = \langle \mu_{k,j,1}(r), \mu_{k,j,2}(r), \dots, \mu_{k,j,i}(r), \dots, \mu_{k,j,t}(r) \rangle$$

And each metric length is an ordered pair as follows:

$$\mu_{k,j,i}(r) = \langle \lambda_{m,k,j,i}(r), \delta_{m,k,j,i}(r) \rangle$$

where $\lambda_{m,k,j,i}(r)$ is a prime number greater than one called the *metric length number* and $\delta_{m,k,j,i}(r)$, which is called the *metric length parity*, is a member of the set $\{0,1\}$. If the reader is finding the proliferation of indices at all confusing, then he or she should note that $\delta_{m,k,j,i}(r)$ is the metric length parity of metric length $\mu_{k,j,i}(r)$, in metric length set $\underline{\mu}_{k,j}(r)$, which is in turn the node attribute of metre node $n_{m,k,j}(r)$ which is the j th metre node in the metre structure $\underline{v}_{m,k}(r)$ which is the metre structure of voice $V_k(r)$.

The first beat of a bar is generally considered the metrically ‘strongest’ beat in the bar. In a bar in common time, the beat with location bar fraction $\langle 1,2 \rangle$ is generally considered the second strongest. The third and fourth strongest are the beats at $\langle 1,4 \rangle$ and $\langle 3,4 \rangle$ respectively, however, these two beats are sometimes considered to be of effectively equal strength.

In a representation in T_{Bach} , each voice $V_k(r)$ has its own metre structure $\underline{v}_{m,k}(r)$ which may differ from those of the other voices. This reflects the fact that, in general, it is quite possible for the pattern of weak and strong beats at a given location l in one voice within a piece to be different from the pattern of weak and strong beats in another voice at location l . This happens, for example, whenever one voice has three durationally equal notes in the same time period as that in which another voice has two or four durationally equal notes.

The function $\sigma_m(V_k(r), l)$ returns a natural number that indicates the *metric strength* of location l in voice $V_k(r)$ in representation r . For example, the metric

strength of any location that is the initial location of a bar is 1. The metric strength of the location $\langle 1,2 \rangle$ in a bar of common time is 2, that of $\langle 1,4 \rangle$, 3 and that of $\langle 3,4 \rangle$, 4. This function therefore attempts to represent the perceived relative metric strength of any location within a bar.

The function $v_m(V_k(r), l)$ returns a natural number that indicates the *metric level* of location l in voice $V_k(r)$ in representation r . For example, the metric level of any location whose bar fraction is $\langle 0,1 \rangle$ is 1 and the metric level of the location $\langle 1,2 \rangle$ in a bar of common time is 2. But in common time, the metric level of any location whose bar fraction is $\langle 1,4 \rangle$ is the *same* as that of any location whose bar fraction is $\langle 3,4 \rangle$ —the metric level of both is 3. The relative metric levels of locations within a particular voice therefore represent intuitions such as that the second and fourth crotchet in a bar in common time are in some sense metrically ‘equal.’ The function $v_m(V_k(r), l)$ attempts to explicate the sense in which such locations are perceived to be metrically ‘equal.’

The definition of $\sigma_m(V_k(r), l)$ implies that no two locations within the same bar may have the same metric strength. However, within any given bar, there is one location whose metric level is 1 and there are 2^{n-2} locations with metric level n for all n greater than 1.

Given a metric length set,

$$\underline{\mu} = \langle \mu_1, \mu_2, \dots, \mu_k, \dots, \mu^t \rangle$$

where $\mu_k = \langle \lambda_k^m, \delta_k^m \rangle$, then this set can be used to generate an infinite ordered set of triples as follows:

$$\underline{\zeta}(\underline{\mu}) = \langle \langle \phi_1, v_1^m, \sigma_1^m \rangle, \langle \phi_2, v_2^m, \sigma_2^m \rangle, \dots, \langle \phi_k, v_k^m, \sigma_k^m \rangle, \dots \rangle$$

where the first member of each triple ϕ_k is a location bar fraction and the second and third members of each triple, v_k^m and σ_k^m , are the metric level and metric strength respectively of any location whose bar fraction is ϕ_k in a bar whose metric length set is $\underline{\mu}$. Note that, in general, for any given metric length set $\underline{\mu}$, the set that contains all and only ϕ_k in the set

$$\underline{\zeta}(\underline{\mu}) = \langle \langle \phi_1, v_1^m, \sigma_1^m \rangle, \langle \phi_2, v_2^m, \sigma_2^m \rangle, \dots, \langle \phi_k, v_k^m, \sigma_k^m \rangle, \dots \rangle$$

is an infinite set but does *not* contain all non-negative rational numbers less than one.

Given a metric length set,

$$\underline{\mu} = \langle \mu_1, \mu_2, \dots, \mu_k, \dots, \mu^t \rangle$$

where $\mu_k = \langle \lambda_k^m, \delta_k^m \rangle$, then the triple $\langle \phi_1, v_1^m, \sigma_1^m \rangle$ in the ordered set,

$$\underline{\zeta}(\underline{\mu}) = \langle \langle \phi_1, v_1^m, \sigma_1^m \rangle, \langle \phi_2, v_2^m, \sigma_2^m \rangle, \dots, \langle \phi_k, v_k^m, \sigma_k^m \rangle, \dots \rangle$$

is defined to be $\langle \langle 0,1 \rangle, 1, 1 \rangle$ and the value of $\langle \phi_k, v_k^m, \sigma_k^m \rangle$ for all k greater than 1 can be found as follows:

v_k^m satisfies the following two conditions:

1. if $k = 1$ then $v_k^m = 1$;

2. if $k > 1$ then $v_k^m = 2 + \text{int}(\log_2(k-1))$.

σ_k^m satisfies the following two conditions:

1. if $k = 1$ then $\sigma_k^m = 1$;
2. if $k > 1$ then $\sigma_k^m = y_k + x_k + 1$,

where

$$y_k = \text{rev}(w_k, z_k) \quad w_k = \text{int}(\log_2(k-1)) \quad z_k = k - x_k - 1 \quad x_k = 2^{w_k}$$

ϕ_k satisfies the following three conditions:

1. if $k = 1$ then $\phi_k = \langle 0, 1 \rangle$
2. if $1 < k \leq \lfloor \underline{\mu} \rfloor + 1$ then $\phi_k = \phi_{x_k} + \frac{j_k(\psi_{y_k} - \phi_{x_k})}{\lambda_{k-1}^m}$
3. if $k > \lfloor \underline{\mu} \rfloor + 1$ then $\phi_k = \phi_{x_k} + \frac{\psi_{y_k} - \phi_{x_k}}{2}$

where

$$x_k = \frac{1}{2} \left(\frac{k-1}{f_k} + 1 \right) \quad \text{and} \quad y_k = x_{k+1}$$

and where

1. if $y_k = 1$ then $\psi_{y_k} = \phi_{y_k} + 1$, otherwise $\psi_{y_k} = \phi_{y_k}$;
2. $z_k = \text{int} \left(\frac{\lambda_{k-1}^m}{2} \right)$
3. if $z_k \bmod 2 = \delta_{k-1}^m$ then $j_k = z_k$ otherwise $j_k = z_k + 1$
4. $f_k = \text{rev} \left(\text{int}(\log_2(k-1)) + 1, 2^{\text{int}(\log_2(\text{rev}(\text{int}(\log_2(k-1))+1, k-1)))} \right)$

Given a metric length set,

$$\underline{\mu}_{k,j}(r) = \langle \mu_{k,j,1}(r), \mu_{k,j,2}(r), \dots, \mu_{k,j,i}(r), \dots, \mu_{k,j,t}(r) \rangle$$

the values of $v_m(V_k(r), l)$ and $\sigma_m(V_k(r), l)$ can be found for any location l such that there exists a triple,

$$\langle \phi, v^m, \sigma^m \rangle \in \underline{\zeta}(\underline{\mu}_{k,j}(r)) \quad \text{where} \quad \phi = \phi(l)$$

To find $v_m(V_k(r), l)$ and $\sigma_m(V_k(r), l)$, one therefore merely has to generate $\underline{\zeta}(\underline{\mu}_{k,j}(r))$ until one determines the triple that contains $\phi(l)$ as its first element.

The function

$$\underline{\mu}(V_k(r), l) = \langle \mu_1(V_k(r), l), \mu_2(V_k(r), l), \dots, \mu_j(V_k(r), l), \dots, \mu_t(V_k(r), l) \rangle$$



Figure 38-3

returns the *operational metric length set* of location l in voice $V_k(r)$. This function therefore returns a value—a metric length set—that provides a complete description of the complete pattern of beats in voice $V_k(r)$ within the bar that contains l . For a given voice $V_k(r)$, all locations within a single bar have the same operational metric length set. That is,

$$\underline{\mu}(V_k(r), l_1) = \underline{\mu}(V_k(r), l_2) \text{ for all } l_1, l_2 \text{ such that } b(l_1) = b(l_2)$$

This implies by virtue of the definition of a structure, that the bar fraction of any metre node segment boundary location will necessarily be $\langle 0, 1 \rangle$.

To determine the *operational metric length set* of location l in voice $V_k(r)$, it is first necessary to know the time signature that applies to voice $V_k(r)$ in the bar which contains l . Time signatures are generally ordered pairs and they are written on a staff with the first member of the pair above the second. These two numbers will be called the *time signature numerator* and *time signature denominator* respectively. Given the time signature that operates in the bar that contains l , it is possible to work out a metric length set that corresponds to this time signature. It is then necessary to examine the voice $V_k(r)$ in the bar that contains l to determine whether there are any explicitly notated deviations from the time signature. For example, the location may occur in the middle of a set of triplets. Such explicitly notated metric deviations in a bar for a given voice must be represented in the operational metric length set for that bar and that voice.

For example, Figure 38-3 is taken from bar 52 in Chopin's study, op.10, no.12.⁴⁶³ The time signature in operation in this bar is $\langle 4, 4 \rangle$ and the metric length set associated with such a time signature is $\langle \rangle$ —that is, the empty ordered set. This metric length set generates the bar fractions $\langle 5, 8 \rangle$ and $\langle 7, 8 \rangle$ as the location bar fractions of the sixth and eighth strongest beats in the bar respectively. However, in Figure 38-3 the third and fourth crotchets of the bar are divided into triplets so the sixth and eighth strongest beats in the bar are, respectively, $\langle 2, 3 \rangle$ and $\langle 11, 12 \rangle$. Consequently, the operational metric length set for this bar is not that associated with the time signature but rather:

$$\langle \langle 2, 1 \rangle, \langle 3, 0 \rangle, \langle 3, 0 \rangle \rangle$$

The derivation of metric length sets for situations in which there are explicit deviations from the time signature metric structure is rather complex to describe formally. However, the derivation of the metric length set associated with any time signature of the form,

$$\langle \text{time signature numerator}, \text{time signature denominator} \rangle$$

is relatively straight-forward and will now be described.

⁴⁶³ Chopin 1957.

In fact, it is only the *numerator* of the time signature that operates in a bar that affects the operational metric length set for that bar. Given that the numerator of the time signature for a given bar is t then the corresponding metric length set,

$$\underline{\mu} = \langle \langle \lambda_1, \delta_1 \rangle, \langle \lambda_2, \delta_2 \rangle, \dots, \langle \lambda_k, \delta_k \rangle, \dots, \langle \lambda^t, \delta^t \rangle \rangle$$

can be found as follows (assuming that there are no explicit metric deviations in the bar such as triplets):

1. Find k_{\max} , which is defined as follows:

$$k_{\max} = \text{int}\left(2^{(\ln(1/t)/\ln(2/3))+1}\right)$$

2. Find $\mu_1 = \langle \lambda_1, \delta_1 \rangle$ where λ_1 is the lowest prime factor of t and where, for all $1 \leq k \leq k_{\max}$, $\delta_k = 1$ if $\lambda_k = 2$ and $\delta_k = 0$ otherwise.
3. Find λ_k for all $2 \leq k \leq k_{\max}$ using the following equation:

$$\lambda_k = \text{lpf}\left(\max\left(\left\{2, t \prod_{n=1}^{u_k} w_{k,n}\right\}\right)\right)$$

where:

- i. $u_k = \text{int}(\log_2 k)$;
 - ii. $h_{k,n} = \left(\left(\dots\left((k - 2^{u_k}) \bmod 2^{u_k - 1}\right)\dots\right) \bmod 2^{u_k - (n-1)}\right) \text{div } 2^{u_k - n}$;
 - iii. $y_{k,n} = k \text{div } 2^{u_k - (n-1)}$;
 - iv. $w_{k,n} = \frac{j_{y_{k,n}}}{\lambda_{y_{k,n}}}$ if $h_{k,n} = 0$ and $w_{k,n} = 1 - \frac{j_{y_{k,n}}}{\lambda_{y_{k,n}}}$ if $h_{k,n} = 1$;
 - v. $j_k = z_k$ if $z_k \bmod 2 = \delta_k$ and $j_k = z_k + 1$ otherwise;
 - vi. $z_k = \text{int}\left(\frac{\lambda_k}{2}\right)$.
4. Find δ_k for all $2 \leq k \leq k_{\max}$ using the fact that $\delta_k = z_k \bmod 2$ if $j_k = z_k$ and $\delta_k \neq z_k \bmod 2$ if $j_k = z_k + 1$.
 5. Finally, it is necessary to truncate $\underline{\mu}$ so that all trailing metric lengths that are equal to $\langle 2, 1 \rangle$ are deleted.

This procedure only gives a default metric length set for a given time signature numerator t . Other metric length sets may operate in a bar with the given time signature numerator but these can only be determined from explicitly notated deviations such as special beaming, triplets and so on.

The metre structure of voice $V_k(r)$ in representation r in the theory system T_{Bach} ,

$$\underline{\mathbf{v}}_{m,k}(r) = \langle n_{m,k,1}(r), n_{m,k,2}(r), \dots, n_{m,k,j}(r), \dots, n_{m,k,t}(r) \rangle$$

must satisfy the following conditions in addition to those that must be satisfied by any structure:

1. for all $l_{m,k,j}(r)$ it must be true that $\underline{\mu}(V_k(r), l_1) = \underline{\mu}(V_k(r), l_2)$ for all $l_1, l_2 \in l_{m,k,j}(r)$;
2. for all $n_{m,k,j}(r), n_{m,k,j+1}(r)$, it must be true that $\underline{\mu}_{\underline{k},j}(r) \neq \underline{\mu}_{\underline{k},j+1}(r)$

For each metre structure in a representation r there exists a unique *metre segmentation* which can be derived directly from its corresponding metre structure. Given that

$$n_{m,k,j}(r) = \langle \underline{1}_{m,k,j}(r), \underline{\mu}(\underline{1}_{m,k,j}(r)) \rangle,$$

then the metre segmentation in r that corresponds to the metre structure

$$\underline{v}_{m,k}(r) = \langle n_{m,k,1}(r), n_{m,k,2}(r), \dots, n_{m,k,j}(r), \dots, n_{m,k,t}(r) \rangle$$

is

$$\underline{\Delta}_{m,k}(r) = \langle \underline{1}_{m,k,1}(r), \underline{1}_{m,k,2}(r), \dots, \underline{1}_{m,k,j}(r), \dots, \underline{1}_{m,k,t}(r) \rangle$$

38.2.1.3 Staff structure

The *staff structure* of voice $V_k(r)$ in representation r in the theory system T_{Bach} is denoted

$$\underline{v}_{s,k}(r) = \langle n_{s,k,1}(r), n_{s,k,2}(r), \dots, n_{s,k,j}(r), \dots, n_{s,k,t}(r) \rangle$$

It is a structure in which each node $n_{s,k,j}(r)$ is a *staff node*. Each staff node,

$$n_{s,k,j}(r) = \langle \underline{1}_{s,k,j}(r), \tau(\underline{1}_{s,k,j}(r)) \rangle$$

is an ordered pair in which the node segment $\underline{1}_{s,k,j}(r)$ is a *staff node segment* and the node attribute $\tau(\underline{1}_{s,k,j}(r))$ is another ordered pair,

$$\tau(\underline{1}_{s,k,j}(r)) = \langle \lambda_o(S(V_k(r), \underline{1}_{s,k,j}(r)), \underline{1}_{s,k,j}(r)), \delta_s(V_k(r), \underline{1}_{s,k,j}(r)) \rangle$$

called the *staff node attribute* of staff node $n_{s,k,j}(r)$. The following abbreviations can be used:

$$\tau_{k,j}(r) =_{\text{df}} \tau(n_{s,k,j}(r)) =_{\text{df}} \tau(\underline{1}_{s,k,j}(r))$$

The function $S(V_k(r), l)$ returns the *operational staff* of voice $V_k(r)$ at location l . The operational staff of $V_k(r)$ at location l is the *staff* (in the representational sense defined below) which represents the graphic staff in the score on which $V_k(r)$ is written at location l .

The function $S(V_k(r), \underline{l})$ returns the *operational staff* of voice $V_k(r)$ for segment \underline{l} which is the representational staff that represents the graphical staff in the score on which voice $V_k(r)$ is written for the whole duration of segment \underline{l} . The value of $S(V_k(r), \underline{l})$ is only defined if

$$S(V_k(r), l_1) = S(V_k(r), l_2) \text{ for all } l_1, l_2 \in \underline{l}$$

The function $\lambda_o(S, l)$ returns the *operational staff order* of staff S at location l which is the ordinal position of the staff on the page, counting down from the top of the system that contains the staff. If a staff is not present at the location in question then the staff order of the staff is 0. For example, if the staff is the top staff in a system, its staff order will be 1; if it is the second down, it will be 2 and so on.

The function $\underline{\lambda}_o(S, \underline{l})$ similarly returns the *operational staff order* of staff s for the entire duration of segment \underline{l} . $\underline{\lambda}_o(S, \underline{l})$ is only defined if

$$\underline{\lambda}_o(S, l_1) = \underline{\lambda}_o(S, l_2) \text{ for all } l_1, l_2 \in \underline{l}$$

The function $\delta_s(V_k(r), l)$ returns the *operational stem direction* of the notes in voice $V_k(r)$ at location l . $\delta_s(V_k(r), l)$ represents the direction in which stems are drawn on the notes in voice $V_k(r)$ at location l . The operational stem direction can have one of only three values: UP, DOWN and BOTH. The value of the operational stem direction is UP if all stems are drawn upwards, DOWN if all stems are drawn downwards and BOTH if stems are drawn both upwards and downwards in the given voice at location l .

The function $\delta_s(V_k(r), \underline{l})$ similarly returns the *operational stem direction* of voice $V_k(r)$ for the entire duration of segment \underline{l} . $\delta_s(V_k(r), \underline{l})$ is only defined if

$$\delta_s(V_k(r), l_1) = \delta_s(V_k(r), l_2) \text{ for all } l_1, l_2 \in \underline{l}$$

The *staff node attribute* of staff node $n_{s,k,j}(r)$,

$$\tau(\underline{l}_{s,k,j}(r)) = \langle \lambda_o(S(V_k(r), \underline{l}_{s,k,j}(r)), \underline{l}_{s,k,j}(r)), \delta_s(V_k(r), \underline{l}_{s,k,j}(r)) \rangle$$

is therefore an ordered pair in which the first member of the pair

$$\lambda_o(S(V_k(r), \underline{l}_{s,k,j}(r)), \underline{l}_{s,k,j}(r))$$

is the operational staff order of the operational staff of voice $V_k(r)$ for segment $\underline{l}_{s,k,j}(r)$; and the second element,

$$\delta_s(V_k(r), \underline{l}_{s,k,j}(r))$$

is the operational stem direction of voice $V_k(r)$ for segment $\underline{l}_{s,k,j}(r)$.

Given the staff structure of voice $V_k(r)$ in representation r in the theory system T_{Bach} ,

$$\underline{v}_{s,k}(r) = \langle n_{s,k,1}(r), n_{s,k,2}(r), \dots, n_{s,k,j}(r), \dots, n_{s,k,t}(r) \rangle$$

then it must be true that $\tau_{k,j}(r) \neq \tau_{k,j+1}(r)$ for all $n_{s,k,j}(r), n_{s,k,j+1}(r)$.

For each staff structure in a representation r there exists a unique *staff segmentation* which can be derived directly from its corresponding staff structure. Given that

$$n_{s,k,j}(r) = \langle \underline{l}_{s,k,j}(r), \tau(\underline{l}_{s,k,j}(r)) \rangle,$$

then the staff segmentation in r that corresponds to the staff structure

$$\underline{v}_{s,k}(r) = \langle n_{s,k,1}(r), n_{s,k,2}(r), \dots, n_{s,k,j}(r), \dots, n_{s,k,t}(r) \rangle$$

is

$$\underline{\Delta}_{s,k}(r) = \langle \underline{l}_{s,k,1}(r), \underline{l}_{s,k,2}(r), \dots, \underline{l}_{s,k,j}(r), \dots, \underline{l}_{s,k,t}(r) \rangle$$

38.2.1.4 Phrase segmentation

The *phrase segmentation* of voice $V_k(r)$ in representation r in the theory system T_{Bach} is a segmentation which is denoted

$$\underline{\Delta}_{p,k}(r) = \langle \underline{1}_{p,k,1}(r), \underline{1}_{p,k,2}(r), \dots, \underline{1}_{p,k,j}(r), \dots, \underline{1}_{p,k,t}(r) \rangle$$

Each phrase in a phrase segmentation is called a *phrase segment*. The phrase segmentation of a voice represents the way in which the voice is divided temporally into contiguous phrases. For it to be possible to derive a phrase segmentation for a score, there must be some *completely unambiguous* notation of phrases in operation in the score and the criteria by which one is determining the boundary locations of phrases must be stated explicitly. In the case of the scores in Bach 1990, the ends of phrases are marked unambiguously by fermata symbols. The phrase segmentation of a score in Bach 1990 is therefore defined to be fully determined by the fermata symbols in the score.

If a phrase is followed by a rest in all parts, as occurs on each occasion in chorale no.85 in Bach 1990 (BWV 45/7), then the phrase segment associated with the phrase in each voice is deemed to terminate *after* this succeeding rest unless a repeat mark intervenes between the fermata and the general rest, in which case the end of the phrase coincides with the repeat mark. For example, the terminal location of the first phrase segment for each voice of chorale no.85 is $\langle 3, \langle 3,4 \rangle \rangle$ but the terminal location of the third phrase for each voice in chorale no.86 (BWV) is $\langle 7, \langle 3,4 \rangle \rangle$.

For example, in chorale no.6 in Bach 1990 (BWV 281), the phrase segmentation for each voice is:

$$\begin{aligned} &\langle \\ &\langle \langle 1, \langle 0,1 \rangle \rangle, \langle 3, \langle 3,4 \rangle \rangle \rangle, \\ &\langle \langle 3, \langle 3,4 \rangle \rangle, \langle 5, \langle 3,4 \rangle \rangle \rangle, \\ &\langle \langle 5, \langle 3,4 \rangle \rangle, \langle 7, \langle 3,4 \rangle \rangle \rangle, \\ &\langle \langle 7, \langle 3,4 \rangle \rangle, \langle 10, \langle 0,1 \rangle \rangle \rangle \\ &\rangle \end{aligned}$$

38.2.1.4 Phrase segmentation

The *phrase segmentation* of voice $V_k(r)$ in representation r in the theory system T_{Bach} is a segmentation which is denoted

$$\underline{\Delta}_{p,k}(r) = \langle \underline{1}_{p,k,1}(r), \underline{1}_{p,k,2}(r), \dots, \underline{1}_{p,k,j}(r), \dots, \underline{1}_{p,k,t}(r) \rangle$$

Each phrase in a phrase segmentation is called a *phrase segment*. The phrase segmentation of a voice represents the way in which the voice is divided temporally into contiguous phrases. For it to be possible to derive a phrase segmentation for a score, there must be some *completely unambiguous* notation of phrases in operation in the score and the criteria by which one is determining the boundary locations of phrases must be stated explicitly. In the case of the scores in Bach 1990, the ends of phrases are marked unambiguously by fermata symbols. The phrase segmentation of a score in Bach 1990 is therefore defined to be fully determined by the fermata symbols in the score.

If a phrase is followed by a rest in all parts, as occurs on each occasion in chorale no.85 in Bach 1990 (BWV 45/7), then the phrase segment associated with the phrase in each voice is deemed to terminate *after* this succeeding rest unless a repeat mark intervenes between the fermata and the general rest, in which case the end of the phrase coincides with the repeat mark. For example, the terminal location of the first phrase segment for each voice of chorale no.85 is $\langle 3, \langle 3, 4 \rangle \rangle$ but the terminal location of the third phrase for each voice in chorale no.86 (BWV) is $\langle 7, \langle 3, 4 \rangle \rangle$.

For example, in chorale no.6 in Bach 1990 (BWV 281), the phrase segmentation for each voice is:

$$\langle \langle \langle 1, \langle 0, 1 \rangle \rangle, \langle 3, \langle 3, 4 \rangle \rangle \rangle, \langle \langle 3, \langle 3, 4 \rangle \rangle, \langle 5, \langle 3, 4 \rangle \rangle \rangle, \langle \langle 5, \langle 3, 4 \rangle \rangle, \langle 7, \langle 3, 4 \rangle \rangle \rangle, \langle \langle 7, \langle 3, 4 \rangle \rangle, \langle 10, \langle 0, 1 \rangle \rangle \rangle \rangle$$

38.3 Staff set

The *staff set* of a representation r in T_{Bach} , is denoted

$$\underline{\Sigma}(r) = \langle S_1(r), S_2(r), \dots, S_k(r), \dots, S_t(r) \rangle$$

It contains all and only staves $S_k(r)$ for a single score, ordered in the following manner:

1. The staves are first ordered in ascending order of location of first appearance in the score. This produces in general a partial ordering since more than one staff could appear for the first time at any given location in the score.
2. Then the staves in each subset containing staves that make their first appearance at the same location, are ordered according to their position *down* the page at their first appearance.

Figure 38.3-1 shows schematically how this system of ordering works for five staves, A-E. Staves A, B and C have earlier positions in the staff set than staves D and E because they make their first appearance earlier in the score. Staff A comes before staff B which comes before staff C because this is their order down the page at their first appearance. Note that in fact, at later positions in the score, this order of position on the page may change. The staff set for a score whose staves behave as shown in Figure 38.3-1 would therefore be $\langle A,B,C,D,E \rangle$.

38.3.1 Staff

An object is a *staff* in a representation r in T_{Bach} if and only if it is a quadruple as follows:

$$S_k(r) = \langle \underline{v}_{k,k}(r), \underline{v}_{b,k}(r), \underline{v}_{f,k}(r), \underline{v}_{o,k}(r) \rangle$$

where $\underline{v}_{k,k}(r)$ is the *key signature structure* of staff $S_k(r)$, $\underline{v}_{b,k}(r)$ is the *bar length structure* of $S_k(r)$, $\underline{v}_{f,k}(r)$ is the *clef structure* of $S_k(r)$ and $\underline{v}_{o,k}(r)$ is the *staff order structure* of $S_k(r)$.

38.3.1.4 Key signature structure

The *key signature structure* of staff $S_k(r)$ in representation r is a structure in which each node is a *key signature node*:

$$\underline{v}_{k,k}(r) = \langle n_{k,k,1}(r), n_{k,k,2}(r), \dots, n_{k,k,j}(r), \dots, n_{k,k,t}(r) \rangle$$

Each key signature node is an ordered pair as follows:

$$n_{k,k,j}(r) = \langle \underline{l}_{k,k,j}(r), \underline{\kappa}(\underline{l}_{k,k,j}(r)) \rangle$$

where the node attribute is a *key signature* which can be denoted in the following ways:

$$\underline{\kappa}_{k,j}(r) =_{\text{df}} \underline{\kappa}(n_{k,k,j}(r)) =_{\text{df}} \underline{\kappa}(\underline{l}_{k,k,j}(r))$$

The function

$$\underline{\kappa}(S_k(r), l) = \langle k_1(S_k(r), l), k_2(S_k(r), l), \dots, k_j(S_k(r), l), \dots, k_7(S_k(r), l) \rangle$$

returns the *operational key signature* for staff $S_k(r)$ at location l . $\underline{\kappa}(S_k(r), l)$ is an ordered set of seven integers. Each element in $\underline{\kappa}(S_k(r), l)$ is an integer called a *key signature displacement*. The key signature displacement $k_j(S_k(r), l)$ represents the displacement of morph $j-1$ as indicated by the key signature in operation on staff $S_k(r)$ at location l . For example, the A major key signature has three sharps—F, C and G.



Figure 38.3-1

Therefore the displacement of morphs, 5, 2 and 6 is one, and the displacement of each of the other morphs is 0. Therefore the *key signature* corresponding to A major would be represented as follows:

$$\langle 0,0,1,0,0,1,1 \rangle$$

Here are some more examples:

B minor	$\langle 0,0,1,0,0,1,0 \rangle$
G flat major	$\langle -1,-1,-1,-1,-1,0,-1 \rangle$
C minor	$\langle -1,-1,0,0,-1,0,0 \rangle$

The function $\underline{\kappa}(S_k(r), \underline{l})$ returns the operational key signature for staff $S_k(r)$ for the complete duration of segment \underline{l} . Clearly, $\underline{\kappa}(S_k(r), \underline{l})$ is defined only if

$$\underline{\kappa}(S_k(r), l_1) = \underline{\kappa}(S_k(r), l_2) \text{ for all } l_1, l_2 \in \underline{l}$$

Given a key signature node,

$$n_{k,k,j}(r) = \langle \underline{l}_{k,k,j}(r), \underline{\kappa}(\underline{l}_{k,k,j}(r)) \rangle$$

then the node attribute $\underline{\kappa}(\underline{l}_{k,k,j}(r))$ is defined to be the operational key signature for staff $S_k(r)$ for segment $\underline{l}_{k,k,j}(r)$. That is,

$$\underline{\kappa}_{k,j}(r) =_{\text{df}} \underline{\kappa}(n_{k,k,j}(r)) =_{\text{df}} \underline{\kappa}(\underline{l}_{k,k,j}(r)) =_{\text{df}} \underline{\kappa}(S_k(r), \underline{l}_{k,k,j}(r))$$

Any key signature structure,

$$\underline{v}_{k,k}(r) = \langle n_{k,k,1}(r), n_{k,k,2}(r), \dots, n_{k,k,j}(r), \dots, n_{k,k,t}(r) \rangle$$

where

$$n_{k,k,j}(r) = \langle \underline{l}_{k,k,j}(r), \underline{\kappa}(\underline{l}_{k,k,j}(r)) \rangle$$

must satisfy the following two conditions:

1. for all $\underline{l}_{k,k,j}(r)$ it must be true that $\underline{\kappa}(S_k(r), l_1) = \underline{\kappa}(S_k(r), l_2)$ for all $l_1, l_2 \in \underline{l}_{k,k,j}(r)$;
2. for all $n_{k,k,j}(r), n_{k,k,j+1}(r)$, it must be true that $\underline{\kappa}_{k,j}(r) \neq \underline{\kappa}_{k,j+1}(r)$.

For each key signature structure in a representation r there exists a unique *key signature segmentation* which can be derived directly from its corresponding key signature structure. Given the above definitions of $\underline{v}_{k,k}(r)$ and $n_{k,k,j}(r)$, then the key signature segmentation in r that corresponds to the key signature structure $\underline{v}_{k,k}(r)$ is

$$\underline{\Delta}_{k,k}(r) = \langle \underline{l}_{k,k,1}(r), \underline{l}_{k,k,2}(r), \dots, \underline{l}_{k,k,j}(r), \dots, \underline{l}_{k,k,t}(r) \rangle$$

38.3.1.5 Bar length structure

The *bar length structure* of staff $S_k(r)$ in representation r is a structure in which each node is a *bar length node*:

$$\underline{v}_{b,k}(r) = \langle n_{b,k,1}(r), n_{b,k,2}(r), \dots, n_{b,k,j}(r), \dots, n_{b,k,t}(r) \rangle$$

Each bar length node is an ordered pair as follows:

$$\mathbf{n}_{b,k,j}(r) = \langle \underline{l}_{b,k,j}(r), \lambda_b(\underline{l}_{b,k,j}(r)) \rangle$$

where the node attribute is a *bar length* which can be denoted in the following ways:

$$\lambda_{b,k,j}(r) =_{\text{df}} \lambda_b(\mathbf{n}_{b,k,j}(r)) =_{\text{df}} \lambda_b(\underline{l}_{b,k,j}(r))$$

The function $\lambda_b(S_k(r), l)$ returns a rational number greater than $\langle 0, 1 \rangle$ indicating the *operational bar length* for staff $S_k(r)$ at location l . The operational bar length for $S_k(r)$ at location l is the length of the bar that contains l on staff $S_k(r)$ expressed as a fraction of a semibreve. $\lambda_b(S_k(r), l)$ is a function of the time signature that operates in the bar that contains l on staff $S_k(r)$. The following table gives the value that is returned by $\lambda_b(S_k(r), l)$ for various time signatures:

Time signature	Operational bar length
4/4	$\langle 1, 1 \rangle$
2/4	$\langle 1, 2 \rangle$
3/4	$\langle 3, 4 \rangle$
6/4	$\langle 3, 2 \rangle$

As can be seen, if the time signature that operates in the bar that contains l on staff $S_k(r)$ is a/b then $\lambda_b(S_k(r), l)$ is simply $l \text{ df } \langle \langle a, b \rangle \rangle$.

The function $\lambda_b(S_k(r), \underline{l})$ returns the operational bar length for staff $S_k(r)$ for the complete duration of segment \underline{l} . Clearly, $\lambda_b(S_k(r), \underline{l})$ is defined only if

$$\lambda_b(S_k(r), l_1) = \lambda_b(S_k(r), l_2) \text{ for all } l_1, l_2 \in \underline{l}$$

Given a bar length node,

$$\mathbf{n}_{b,k,j}(r) = \langle \underline{l}_{b,k,j}(r), \lambda_b(\underline{l}_{b,k,j}(r)) \rangle$$

then the node attribute $\lambda_b(\underline{l}_{b,k,j}(r))$ is defined to be the operational bar length for staff $S_k(r)$ for segment $\underline{l}_{b,k,j}(r)$. That is,

$$\lambda_{b,k,j}(r) =_{\text{df}} \lambda_b(\mathbf{n}_{b,k,j}(r)) =_{\text{df}} \lambda_b(\underline{l}_{b,k,j}(r)) =_{\text{df}} \lambda_b(S_k(r), \underline{l}_{b,k,j}(r))$$

Any bar length structure,

$$\underline{\mathbf{v}}_{b,k}(r) = \langle \mathbf{n}_{b,k,1}(r), \mathbf{n}_{b,k,2}(r), \dots, \mathbf{n}_{b,k,j}(r), \dots, \mathbf{n}_{b,k,t}(r) \rangle$$

where

$$\mathbf{n}_{b,k,j}(r) = \langle \underline{l}_{b,k,j}(r), \lambda_b(\underline{l}_{b,k,j}(r)) \rangle$$

must satisfy the following two conditions:

1. for all $\underline{l}_{b,k,j}(r)$ it must be true that $\lambda_b(S_k(r), l_1) = \lambda_b(S_k(r), l_2)$ for all $l_1, l_2 \in \underline{l}_{b,k,j}(r)$;
2. for all $\mathbf{n}_{b,k,j}(r), \mathbf{n}_{b,k,j+1}(r)$, it must be true that $\lambda_{b,k,j}(r) \neq \lambda_{b,k,j+1}(r)$.

For each bar length structure in a representation r there exists a unique *bar length segmentation* which can be derived directly from its corresponding bar length structure.

Given the above definitions of $\underline{v}_{b,k}(r)$ and $n_{b,k,j}(r)$, then the bar length segmentation in r that corresponds to the bar length structure $\underline{v}_{b,k}(r)$ is

$$\underline{\Delta}_{b,k}(r) = \langle \underline{l}_{b,k,1}(r), \underline{l}_{b,k,2}(r), \dots, \underline{l}_{b,k,j}(r), \dots, \underline{l}_{b,k,t}(r) \rangle$$

38.3.1.6 Clef structure

The *clef structure* of staff $S_k(r)$ in representation r is a structure in which each node is a *clef node*:

$$\underline{v}_{f,k}(r) = \langle n_{f,k,1}(r), n_{f,k,2}(r), \dots, n_{f,k,j}(r), \dots, n_{f,k,t}(r) \rangle$$

Each clef node is an ordered pair as follows:

$$n_{f,k,j}(r) = \langle \underline{l}_{f,k,j}(r), f(\underline{l}_{f,k,j}(r)) \rangle$$

where the node attribute is a *clef* which can be denoted in the following ways:

$$f_{k,j}(r) =_{df} f(n_{f,k,j}(r)) =_{df} f(\underline{l}_{f,k,j}(r))$$

An object is a *clef* if and only if it is an ordered triple as follows:

$$f = \langle p_m(f), i(f), \lambda_f(f) \rangle$$

where the first member of the triple is a morphetic pitch called the *clef type*, the second member of the triple is a pitch interval called the *clef transposition* and the third member of the triple is an integer between 0 and 10 called the *clef height*.

The function

$$f(S_k(r), l) = \langle p_{m,f}(S_k(r), l), i_f(S_k(r), l), \lambda_f(S_k(r), l) \rangle$$

returns the clef which is the *operational clef* for staff $S_k(r)$ at location l . Each clef symbol written on a score is an embellished letter and signifies the *written* morphetic pitch of the notes on the line or space on which the clef is written. For example, a staff with a G-clef on the second line from the bottom (that is, a normal treble clef) indicates that the pitch name as written of a note on the second line up on the staff is a G and that this G is G_4 . Similarly, a normal bass clef indicates that the written pitch name of a note on the second line down on the staff is an F and that this F is, in fact, F_3 .

The clef type $p_{m,f}(S_k(r), l)$ of a clef is the morphetic pitch (as written) associated with notes on the line or space referenced by the clef. For example, the morphetic pitch of a G on the second line up on a clef with a treble G-clef is 27 so the type of this clef is 27; similarly, the type of a normal bass F-clef is 19. Tenor G-clefs are sometimes written with a small '8' attached at the bottom, or as a double G-clef. This indicates that the G on the line referenced by the clef is the first G below middle C, therefore the type of this kind of tenor clef is 20. Note that if a tenor clef is indistinguishable from a normal treble G-clef then it must be given a type of 27 even if it is clear that it is intended to indicate a tenor clef. The fact that the sounding notes are intended to be an octave lower can be represented in the clef transposition of the clef.

The clef transposition $i_f(S_k(r), l)$ of a clef is a pitch interval that indicates whether or not the sounding pitches of the notes on the staff are the same as their written pitches. For example, the treble G-clef indicates that G on the second line up on the staff is G_4 .

However, as mentioned above, the normal G-clef is sometimes used for tenor parts without any distinguishing features, in which case it is effectively a transposed clef. If p_1 is the *written* pitch of the line or space to which a clef usually refers and p_2 is the *sounding* pitch of the line or space to which a clef refers then the transposition of the clef is the pitch interval, $i(p_1, p_2)$. The transposition corresponding to the tenor G-clef would therefore be $\langle -12, -7 \rangle$. Similarly, the clef transposition of the B-flat clarinet G-clef is $\langle -2, -1 \rangle$ and the transposition of the piccolo G-clef is $\langle 12, 7 \rangle$.

The clef height $\lambda_f(S_k(r), l)$ of a clef indicates the position of the line or space to which the clef name refers on its staff. The clef height is an integer between 0 and 10 and it is determined as follows: 0, for the space below the bottom line on a normal five-line staff, 1 for the bottom line on a normal five-line staff, 2 for the space above this, three for the line above this space and so on up to 9 for the top line of the staff and 10 for the space above the top line.

For example, the normal treble clef symbol is represented by the clef, $\langle 27, \langle 0, 0 \rangle, 3 \rangle$; the normal bass clef symbol is represented as $\langle 19, \langle 0, 0 \rangle, 7 \rangle$; the normal alto C clef on the middle line of the staff is represented as $\langle 23, \langle 0, 0 \rangle, 5 \rangle$; the B-flat clarinet G-clef is represented by the clef, $\langle 27, \langle -2, -1 \rangle, 3 \rangle$; and the piccolo G-clef is represented by $\langle 27, \langle 12, 7 \rangle, 3 \rangle$.

The function $f(S_k(r), l)$ returns the operational clef for staff $S_k(r)$ for the complete duration of segment l . Clearly, $f(S_k(r), l)$ is defined only if

$$f(S_k(r), l_1) = f(S_k(r), l_2) \text{ for all } l_1, l_2 \in l$$

Given a clef node,

$$n_{f,k,j}(r) = \langle \underline{l}_{f,k,j}(r), f(\underline{l}_{f,k,j}(r)) \rangle$$

then the node attribute $f(\underline{l}_{f,k,j}(r))$ is defined to be the operational clef for staff $S_k(r)$ for segment $\underline{l}_{f,k,j}(r)$. That is,

$$f_{k,j}(r) =_{\text{df}} f(n_{f,k,j}(r)) =_{\text{df}} f(\underline{l}_{f,k,j}(r)) =_{\text{df}} f(S_k(r), \underline{l}_{f,k,j}(r))$$

Any clef structure,

$$\underline{v}_{f,k}(r) = \langle n_{f,k,1}(r), n_{f,k,2}(r), \dots, n_{f,k,j}(r), \dots, n_{f,k,t}(r) \rangle$$

where

$$n_{f,k,j}(r) = \langle \underline{l}_{f,k,j}(r), f(\underline{l}_{f,k,j}(r)) \rangle$$

must satisfy the following two conditions:

1. for all $\underline{l}_{f,k,j}(r)$ it must be true that $f(S_k(r), l_1) = f(S_k(r), l_2)$ for all $l_1, l_2 \in \underline{l}_{f,k,j}(r)$;
2. for all $n_{f,k,j}(r), n_{f,k,j+1}(r)$, it must be true that $f_{k,j}(r) \neq f_{k,j+1}(r)$.

For each clef structure in a representation r there exists a unique *clef segmentation* which can be derived directly from its corresponding clef structure. Given the above

definitions of $\underline{v}_{f,k}(r)$ and $n_{f,k,j}(r)$, then the clef segmentation in r that corresponds to the clef structure $\underline{v}_{f,k}(r)$ is

$$\underline{\Lambda}_{f,k}(r) = \langle \underline{l}_{f,k,1}(r), \underline{l}_{f,k,2}(r), \dots, \underline{l}_{f,k,j}(r), \dots, \underline{l}_{f,k,t}(r) \rangle$$

38.3.1.7 Staff order structure

The *staff order structure* of staff $S_k(r)$ in representation r is a structure in which each node is a *staff order node*:

$$\underline{v}_{o,k}(r) = \langle n_{o,k,1}(r), n_{o,k,2}(r), \dots, n_{o,k,j}(r), \dots, n_{o,k,t}(r) \rangle$$

Each staff order node is an ordered pair as follows:

$$n_{o,k,j}(r) = \langle \underline{l}_{o,k,j}(r), \lambda_o(\underline{l}_{o,k,j}(r)) \rangle$$

where the node attribute is a *staff order* which can be denoted in the following ways:

$$\lambda_{o,k,j}(r) =_{\text{df}} \lambda_o(n_{o,k,j}(r)) =_{\text{df}} \lambda_o(\underline{l}_{o,k,j}(r))$$

The node attribute of a staff order node is a non-negative integer called a *staff order*.

The functions $\lambda_o(S_k(r), l)$ and $\lambda_o(S_k(r), \underline{l})$ were defined in section 38.2.1.3 above. Given a staff order node,

$$n_{o,k,j}(r) = \langle \underline{l}_{o,k,j}(r), \lambda_o(\underline{l}_{o,k,j}(r)) \rangle$$

then the node attribute $\lambda_o(\underline{l}_{o,k,j}(r))$ is defined to be the operational staff order for staff $S_k(r)$ for segment $\underline{l}_{o,k,j}(r)$. That is,

$$\lambda_{o,k,j}(r) =_{\text{df}} \lambda_o(n_{o,k,j}(r)) =_{\text{df}} \lambda_o(\underline{l}_{o,k,j}(r)) =_{\text{df}} \lambda_o(S_k(r), \underline{l}_{o,k,j}(r))$$

Any staff order structure,

$$\underline{v}_{o,k}(r) = \langle n_{o,k,1}(r), n_{o,k,2}(r), \dots, n_{o,k,j}(r), \dots, n_{o,k,t}(r) \rangle$$

where

$$n_{o,k,j}(r) = \langle \underline{l}_{o,k,j}(r), \lambda_o(\underline{l}_{o,k,j}(r)) \rangle$$

must satisfy the following two conditions:

1. for all $\underline{l}_{o,k,j}(r)$ it must be true that $\lambda_o(S_k(r), l_1) = \lambda_o(S_k(r), l_2)$ for all $l_1, l_2 \in \underline{l}_{o,k,j}(r)$;
2. for all $n_{o,k,j}(r), n_{o,k,j+1}(r)$, it must be true that $\lambda_{o,k,j}(r) \neq \lambda_{o,k,j+1}(r)$.

For each staff order structure in a representation r there exists a unique *staff order segmentation* which can be derived directly from its corresponding staff order structure. Given the above definitions of $\underline{v}_{o,k}(r)$ and $n_{o,k,j}(r)$, then the staff order segmentation in r that corresponds to the staff order structure $\underline{v}_{o,k}(r)$ is

$$\underline{\Lambda}_{o,k}(r) = \langle \underline{l}_{o,k,1}(r), \underline{l}_{o,k,2}(r), \dots, \underline{l}_{o,k,j}(r), \dots, \underline{l}_{o,k,t}(r) \rangle$$

39 The derivation and composing algorithms of T_{Bach}

Both the composing algorithm and the derivation algorithm of T_{Bach} have been fully implemented in a working computer program called IOTA, written in Macintosh Common Lisp Version 2.0.1.⁴⁶⁴ The program was developed and tested on a Macintosh Quadra 610 computer running the System Software B1–7.1 operating system. The source code and executable file for IOTA are included on the diskettes that accompany this thesis. Also included on these diskettes are files containing the representations and derivations for 22 of the 29 chorales in the corpus, along with files containing the representations and derivations of 7 other members of the universal set of well-formed scores, randomly generated by the composing algorithm.

The implementation of the composing algorithm $\gamma(T_{\text{Bach}})$ in IOTA takes no arguments and no user input other than an optional random number seed (`*random-state*`) which determines the sequence of random numbers that the algorithm will employ to make its decisions on a given execution. The implementation of the derivation algorithm $\delta(T_{\text{Bach}})$ in IOTA takes a file containing a derivation d as input and generates as output a well-formed representation if and only if d correctly describes one possible complete execution of the composing algorithm.

The composing algorithm and the derivation algorithm are essentially equivalent. However, the composing algorithm must *search* for a well-formed representation, and finding such a representation usually involves a great deal of backtracking. On the other hand, the derivation algorithm merely has to *check* whether or not the composing algorithm would successfully generate a well-formed representation if it followed the sequence of decisions described in a given derivation. Consequently, the derivation algorithm is much simpler to describe than the composing algorithm because it does not need to employ any of the complex control structures that the composing algorithm needs in order to perform exhaustive searches.

The derivation algorithm $\delta(T_{\text{Bach}})$ takes a derivation file as input and attempts to use the data in this derivation to generate a well-formed representation. On the least detailed level of description, $\delta(T_{\text{Bach}})$ can conveniently be divided into the following steps:

1. Initialization of global variables indicating the metre structure (which is the same for each voice within any given corpus score), the phrase segmentation and the tonic and mode of the home key.
2. Derivation of a *triad structure*. A triad structure is a *structure* in the sense defined in section 37.2.4 above in which each node is a *triad node*. The node attribute of each triad node indicates the chord function (i.e. dominant, tonic, etc.) and root of the triad that is hypothesized to be operating over the duration of the triad node segment.
3. Derivation of an *inversion structure* in which each triad node in the triad structure is supplemented with a datum indicating which inversion of the triad will be employed over the duration of the triad node segment.

⁴⁶⁴ Macintosh Common Lisp Version 2.0.1, Copyright © 1988–1992 Apple Computer, Inc.

f	$t_{lit}(\langle C, \natural \rangle, f)$
i-min	$\langle \langle c, \text{natural} \rangle, \text{min} \rangle$
i-maj	$\langle \langle c, \text{natural} \rangle, \text{maj} \rangle$
ii-min	$\langle \langle d, \text{natural} \rangle, \text{min} \rangle$
ii-maj	$\langle \langle d, \text{flat} \rangle, \text{maj} \rangle$
iii-min	$\langle \langle e, \text{natural} \rangle, \text{min} \rangle$
iii-maj	$\langle \langle e, \text{flat} \rangle, \text{maj} \rangle$
iv-min	$\langle \langle f, \text{natural} \rangle, \text{min} \rangle$
iv-maj	$\langle \langle f, \text{natural} \rangle, \text{maj} \rangle$
v-min	$\langle \langle g, \text{natural} \rangle, \text{min} \rangle$
v-maj	$\langle \langle g, \text{natural} \rangle, \text{maj} \rangle$
vi-min	$\langle \langle a, \text{natural} \rangle, \text{min} \rangle$
vi-maj	$\langle \langle a, \text{flat} \rangle, \text{maj} \rangle$
vii-min	$\langle \langle b, \text{flat} \rangle, \text{min} \rangle$
vii-maj	$\langle \langle b, \text{flat} \rangle, \text{maj} \rangle$
no-triad	nil

Figure 39-1

4. Derivation of an *upper inner note structure*. In this stage of the algorithm, a simple, four-voice, homophonic piece is derived from the triad and inversion structures by distributing the notes of each triad between the four voices according to a complex set of voice-leading rules.
5. Derivation of a *representation*. The simple, homophonic ‘chorale’ generated at the end of the previous stage is ‘embellished’ using Schenker-like, recursive, ‘prolongational’ rules that insert neighbour notes, fill in thirds with ‘transition’ notes, combine repeated notes in a given voice into single notes, insert anticipations and suspensions and so on.

The first main step in the derivation algorithm is to establish values for the global variables, *mode*, *tonic genus name* and *metric length set*. The value of *mode* must be a member of the set

$$\{\text{major}, \text{minor}\}$$

If *mode* is major, then the value of *tonic genus name* must be a member of the set

$$\{\langle F, \natural \rangle, \langle G, \natural \rangle, \langle A, \natural \rangle, \langle B, \flat \rangle, \langle C, \natural \rangle, \langle D, \natural \rangle\}$$

If *mode* is minor, then it must be in the set

$$\{\langle G, \natural \rangle, \langle D, \natural \rangle, \langle B, \natural \rangle, \langle A, \natural \rangle\}$$

The value of *metric length set* must be either $\langle \rangle$, corresponding to simple duple time; or $\langle \langle 3, 0 \rangle \rangle$, corresponding to simple triple time. In any given chorale in $\underline{s}_c(T_{\text{Bach}})$, the phrase segmentations of the four voices will be equal. To improve efficiency, each representation generated by the implementation of $\gamma(T_{\text{Bach}})$ in IOTA therefore stores the phrase segmentation—which will be the same for each voice—as a single global

attribute of a representation. The phrase segmentation of a representation is established in the first main stage of the composing and derivation algorithms of T_{Bach} .

For example, for the corpus score of the chorale ‘Christus, der ist mein Leben’ (BWV 281, no.6 in Bach 1990), *metric length set* is nil, *tonic genus name* is $\langle F, \text{natural} \rangle$, *mode* is major and the phrase segmentation is

$$\langle\langle\langle 1, \langle 0, 1 \rangle \rangle, \langle 3, \langle 3, 4 \rangle \rangle\rangle, \langle\langle\langle 3, \langle 3, 4 \rangle \rangle, \langle 5, \langle 3, 4 \rangle \rangle\rangle, \langle\langle\langle 5, \langle 3, 4 \rangle \rangle, \langle 7, \langle 3, 4 \rangle \rangle\rangle, \langle\langle\langle 7, \langle 3, 4 \rangle \rangle, \langle 10, \langle 0, 1 \rangle \rangle\rangle\rangle$$

The purpose of the second main stage in $\delta(T_{\text{Bach}})$ is to derive a *triad structure*. An object is a *triad structure* if and only if it is a structure (in the sense defined in section 37.2.4 above) in which each node is a *triad node*:

$$\underline{v}^{\text{triad}} = \langle n_1^{\text{triad}}, n_2^{\text{triad}}, \dots, n_k^{\text{triad}}, \dots, n^{\text{triad}, t} \rangle$$

Each *triad node* n_k^{triad} is an ordered pair as follows:

$$n_k^{\text{triad}} = \langle \underline{l}(n_k^{\text{triad}}), \langle \underline{\tau}_{\text{fun}}(n_k^{\text{triad}}), \underline{t}_{\text{lit}}(n_k^{\text{triad}}) \rangle \rangle$$

where the function $\underline{\tau}_{\text{fun}}(n_k^{\text{triad}})$ returns the *triad function* of n_k^{triad} and the function $\underline{t}_{\text{lit}}(n_k^{\text{triad}})$ returns the *literal triad* of n_k^{triad} . The attribute of a triad node is called a *triad node attribute*.

Given a triad structure,

$$\underline{v}^{\text{triad}} = \langle n_1^{\text{triad}}, n_2^{\text{triad}}, \dots, n_k^{\text{triad}}, \dots, n^{\text{triad}, t} \rangle$$

such that

$$n_k^{\text{triad}} = \langle \underline{l}(n_k^{\text{triad}}), \langle \underline{\tau}_{\text{fun}}(n_k^{\text{triad}}), \underline{t}_{\text{lit}}(n_k^{\text{triad}}) \rangle \rangle$$

then the *triad segmentation* of $\underline{v}^{\text{triad}}$ is defined and denoted as follows:

$$\underline{\Lambda}(\underline{v}^{\text{triad}}) = \langle \underline{l}(n_1^{\text{triad}}), \underline{l}(n_2^{\text{triad}}), \dots, \underline{l}(n_k^{\text{triad}}), \dots, \underline{l}(n^{\text{triad}, t}) \rangle$$

The *triad function* $\underline{\tau}_{\text{fun}}(n_k^{\text{triad}})$ must be an ordered set of *function names* as follows:

$$\underline{\tau}_{\text{fun}}(n_k^{\text{triad}}) = \langle \underline{t}_{\text{fun}, 1}(n_k^{\text{triad}}), \underline{t}_{\text{fun}, 2}(n_k^{\text{triad}}), \dots, \underline{t}_{\text{fun}, j}(n_k^{\text{triad}}), \dots, \underline{t}_{\text{fun}, t}(n_k^{\text{triad}}) \rangle$$

where each function name $\underline{t}_{\text{fun}, j}(n_k^{\text{triad}})$ must be a member of the set $\underline{t}_{\text{fun}}^u$ which is as follows:

{i-min, i-maj, ii-min, ii-maj, iii-min, iii-maj, iv-min, iv-maj,
v-min, v-maj, vi-min, vi-maj, vii-min, vii-maj, no-triad}

Each *literal triad* $\underline{t}_{\text{lit}}(n_k^{\text{triad}})$ must be an ordered pair as follows:

$$\underline{t}_{\text{lit}}(n_k^{\text{triad}}) = \langle \underline{\text{gn}}_{\text{root}}(\underline{t}_{\text{lit}}(n_k^{\text{triad}})), \underline{\text{ty}}(\underline{t}_{\text{lit}}(n_k^{\text{triad}})) \rangle$$

where $\underline{\text{gn}}_{\text{root}}(\underline{t}_{\text{lit}}(n_k^{\text{triad}}))$ is the *root genus name* of $\underline{t}_{\text{lit}}(n_k^{\text{triad}})$ and $\underline{\text{ty}}(\underline{t}_{\text{lit}}(n_k^{\text{triad}}))$ is the *triad type* of $\underline{t}_{\text{lit}}(n_k^{\text{triad}})$ which must be a member of {maj, min}.

The function $t_{lit}(g, f)$ returns the literal triad whose function name is f in the key whose tonic genus name is g . The literal triads returned by $t_{lit}(g, f)$ for each value of $f \in t_{fun}^u$ and for $g = \langle C, \sharp \rangle$ are given in Figure 39-1. The value of $t_{lit}(g, f)$ for any tonic genus name other than C can be found by transposition of the literal triads given in Figure 39-1.

Figure 39-2 shows part of the thirds relation genus name digraph. This is the same as the thirds relation genus digraph shown in Figure 34-4 except that each genus has been replaced by its equivalent genus name. The literal triads returned by $t_{lit}(g, f)$ for $g = \langle C, \sharp \rangle$ have been indicated by placing a 'right angle' parallel to each walk whose walk set is equal to one of the triads in the right-hand column of Figure 39-1. Note that the resulting compact region of the graph falls within a 'band' encircling the cylinder in which the digraph is embedded. Note also that this band contains 16 distinct circuits of length seven and that all these circuits are members of the set $A_{thirds}^{q,\gamma}$ defined in chapter 35 above.

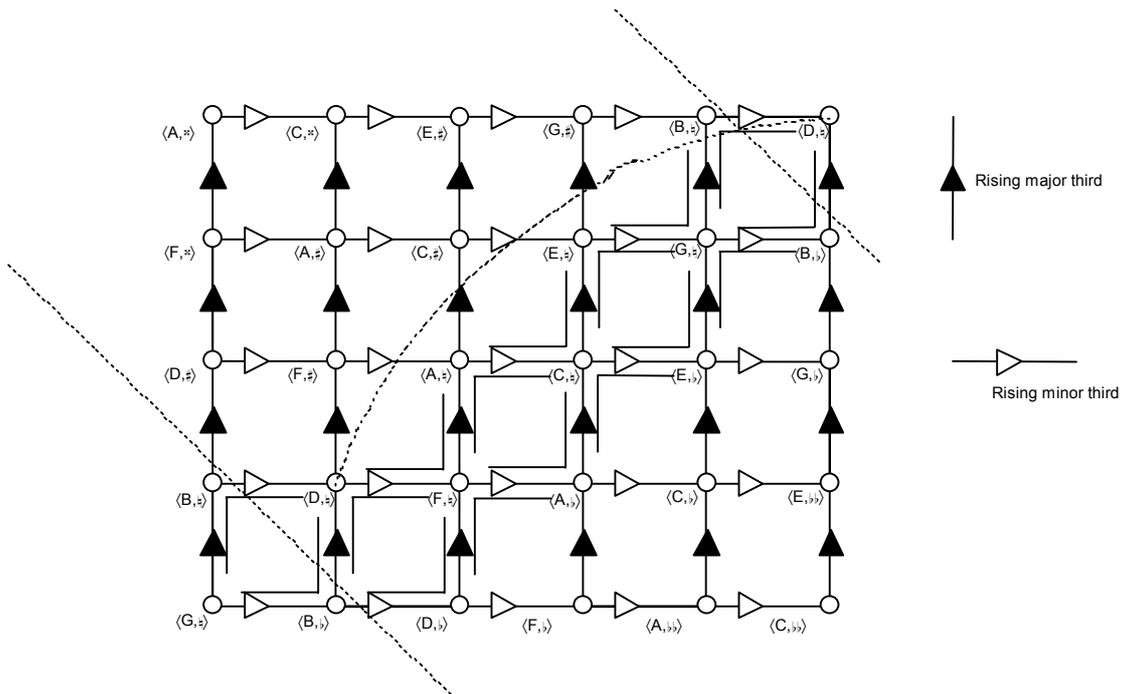
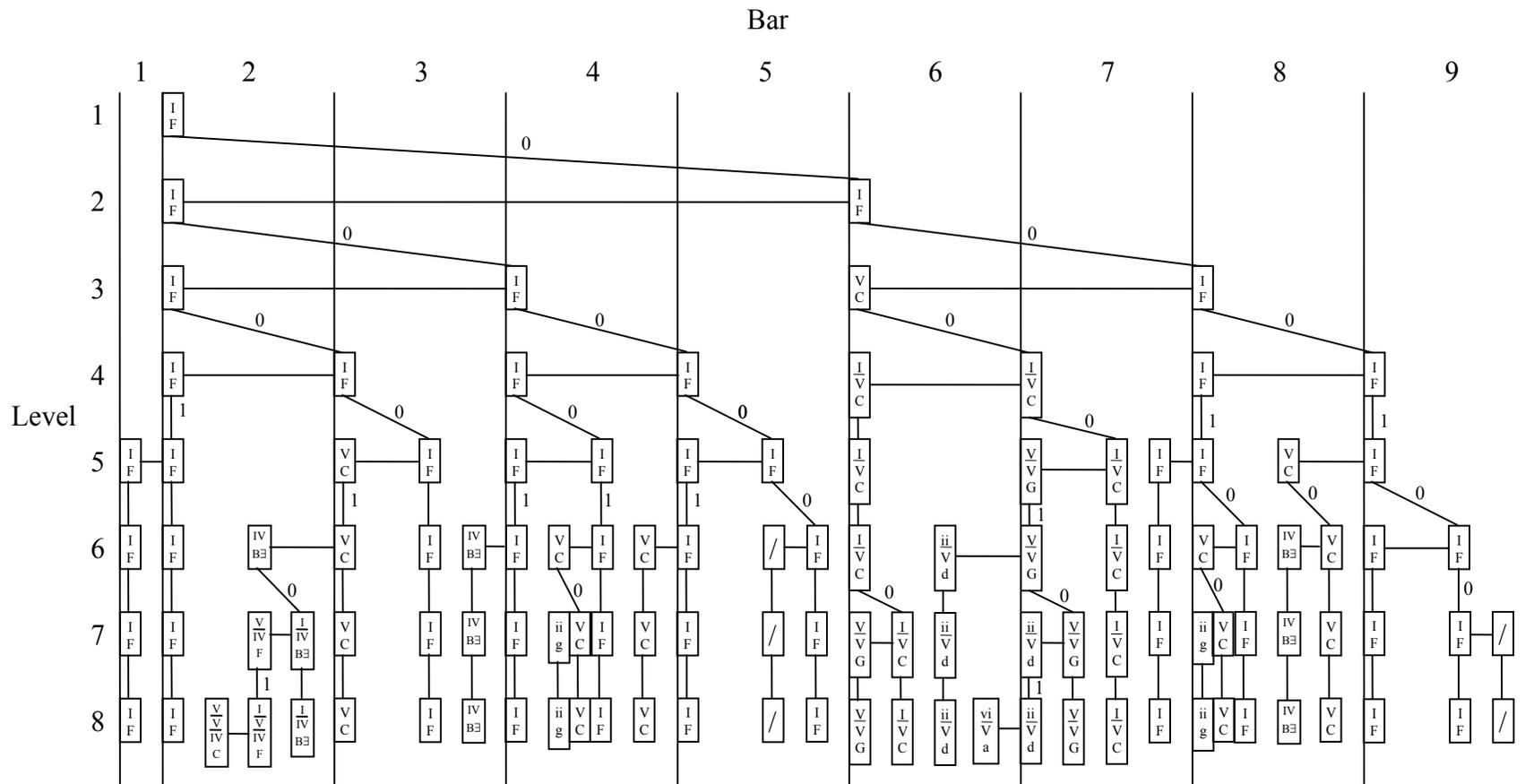


Figure 39-2

Figure 39-3



As mentioned above, the *triad function* $\underline{\tau}_{\text{fun}}(n_k^{\text{triad}})$ of a triad node must be an ordered set of *function names* as follows:

$$\underline{\tau}_{\text{fun}}(n_k^{\text{triad}}) = \langle t_{\text{fun},1}(n_k^{\text{triad}}), t_{\text{fun},2}(n_k^{\text{triad}}), \dots, t_{\text{fun},j}(n_k^{\text{triad}}), \dots, t_{\text{fun},t}(n_k^{\text{triad}}) \rangle$$

$t_{\text{lit}}(n_k^{\text{triad}})$ depends entirely upon $\underline{\tau}_{\text{fun}}(n_k^{\text{triad}})$ and can be derived directly from it.⁴⁶⁵ The literal triad of any triad function of cardinality 1 can be derived from Figure 39-1 either directly or by transposition. If a triad function has a cardinality greater than 1 then derivation of the corresponding literal triad is slightly more complicated. For example, the literal triad that corresponds to the triad function,

$$\langle \text{ii-min, v-maj} \rangle$$

is the triad that functions as the minor supertonic triad of the major dominant triad in the key whose tonic is *tonic genus name*. For example, if *tonic genus name* is $\langle C, \sharp \rangle$ then the literal triad that corresponds to the function name $\langle \text{ii-min, v-maj} \rangle$ is $\langle \langle A, \sharp \rangle, \text{min} \rangle$. In principle, triad functions may have any cardinality. For example, given a *tonic genus name* of $\langle C, \sharp \rangle$, the literal triad that corresponds to the triad function

$$\langle \text{ii-min, v-maj, vii-min, iv-maj, iii-maj} \rangle$$

is $\langle \langle E, \flat \rangle, \text{min} \rangle$.

Figure 39-3 shows one possible derivation in T_{Bach} of a triad structure from which $\gamma(T_{\text{Bach}})$ can generate the representation of the chorale ‘Christus, der ist mein Leben’ (BWV 281, no.6 in Bach 1990). This particular derivation is eight levels deep. Each level contains a number of triad nodes and each level as a whole corresponds to a triad structure. For example, the triad structure that corresponds to Level 1 in Figure 39-3 is simply

$$\langle \langle \langle \langle 1,0 \rangle, \langle 10,0 \rangle \rangle, \langle \text{i-maj} \rangle, \langle \langle F, \sharp \rangle, \text{maj} \rangle \rangle \rangle$$

and the triad structure that corresponds to Level 3 is

$$\left\langle \begin{array}{l} \langle \langle \langle 1,0 \rangle, \langle 4,0 \rangle \rangle, \langle \text{i-maj} \rangle, \langle \langle F, \sharp \rangle, \text{maj} \rangle \rangle, \\ \langle \langle \langle 4,0 \rangle, \langle 6,0 \rangle \rangle, \langle \text{i-maj} \rangle, \langle \langle F, \sharp \rangle, \text{maj} \rangle \rangle, \\ \langle \langle \langle 6,0 \rangle, \langle 8,0 \rangle \rangle, \langle \text{v-maj} \rangle, \langle \langle C, \sharp \rangle, \text{maj} \rangle \rangle, \\ \langle \langle \langle 8,0 \rangle, \langle 10,0 \rangle \rangle, \langle \text{i-maj} \rangle, \langle \langle F, \sharp \rangle, \text{maj} \rangle \rangle \end{array} \right\rangle$$

Figure 39-4 shows a listing generated by IOTA of the triad structure that corresponds to Level 8 in Figure 39-3. Note that IOTA uses a more economical notation for the structures in a representation than that defined in section 37.2.4. The terminal location of a node in a structure is equal either to the initial location of the following node or the terminal location of the universal segment. Therefore it would be inefficient to store the terminal location of every node segment explicitly in an IOTA data-

⁴⁶⁵ In IOTA, derivation of literal triads from triad functions is performed by the function `triad-function-to-triad` whose definition will be found in the file `generate-triad-structure.lisp`.

structure that represents a structure. The terminal location of the universal segment of a representation is stored in IOTA as a global attribute of the representation.

The triad structure that corresponds to Level 1 in Figure 39-3 is called the *initial triad structure* of this derivation. The initial triad structure for any derivation in T_{Bach} must equal

$$\langle\langle\langle\langle l^u, \langle\langle i\text{-maj} \rangle, \langle g_{\text{tonic}}, \text{maj} \rangle \rangle \rangle \rangle \rangle$$

where g_{tonic} is equal to the variable *tonic genus name*. In general, the triad structure that

(1	(1 3/4)	((TRIAD I-MAJ)	((F NATURAL) MAJ))
(2	(2 0)	((TRIAD I-MAJ)	((F NATURAL) MAJ))
(3	(2 1/4)	((TRIAD V-MAJ V-MAJ IV-MAJ)	((C NATURAL) MAJ))
(4	(2 1/2)	((TRIAD I-MAJ V-MAJ IV-MAJ)	((F NATURAL) MAJ))
(5	(2 3/4)	((TRIAD I-MAJ IV-MAJ)	((B FLAT) MAJ))
(6	(3 0)	((TRIAD V-MAJ)	((C NATURAL) MAJ))
(7	(3 1/2)	((TRIAD I-MAJ)	((F NATURAL) MAJ))
(8	(3 3/4)	((TRIAD IV-MAJ)	((B FLAT) MAJ))
(9	(4 0)	((TRIAD I-MAJ)	((F NATURAL) MAJ))
(10	(4 1/4)	((TRIAD II-MIN)	((G NATURAL) MIN))
(11	(4 3/8)	((TRIAD V-MAJ)	((C NATURAL) MAJ))
(12	(4 1/2)	((TRIAD I-MAJ)	((F NATURAL) MAJ))
(13	(4 3/4)	((TRIAD V-MAJ)	((C NATURAL) MAJ))
(14	(5 0)	((TRIAD I-MAJ)	((F NATURAL) MAJ))
(15	(5 1/2)	((TRIAD NO-TRIAD)	(NIL))
(16	(5 3/4)	((TRIAD I-MAJ)	((F NATURAL) MAJ))
(17	(6 0)	((TRIAD V-MAJ V-MAJ)	((G NATURAL) MAJ))
(18	(6 1/4)	((TRIAD I-MAJ V-MAJ)	((C NATURAL) MAJ))
(19	(6 1/2)	((TRIAD II-MIN V-MAJ)	((D NATURAL) MIN))
(20	(6 3/4)	((TRIAD VI-MIN V-MAJ)	((A NATURAL) MIN))
(21	(7 0)	((TRIAD II-MIN V-MAJ)	((D NATURAL) MIN))
(22	(7 1/4)	((TRIAD V-MAJ V-MAJ)	((G NATURAL) MAJ))
(23	(7 1/2)	((TRIAD I-MAJ V-MAJ)	((C NATURAL) MAJ))
(24	(7 3/4)	((TRIAD I-MAJ)	((F NATURAL) MAJ))
(25	(8 0)	((TRIAD II-MIN)	((G NATURAL) MIN))
(26	(8 1/8)	((TRIAD V-MAJ)	((C NATURAL) MAJ))
(27	(8 1/4)	((TRIAD I-MAJ)	((F NATURAL) MAJ))
(28	(8 1/2)	((TRIAD IV-MAJ)	((B FLAT) MAJ))
(29	(8 3/4)	((TRIAD V-MAJ)	((C NATURAL) MAJ))
(30	(9 0)	((TRIAD I-MAJ)	((F NATURAL) MAJ))
(31	(9 1/2)	((TRIAD I-MAJ)	((F NATURAL) MAJ))
(32	(9 3/4)	((TRIAD NO-TRIAD)	(NIL))

Figure 39-4

corresponds to Level k in Figure 39-3 is derived from the triad structure of Level $k-1$ by applying *rewrite rules* to the triad nodes in Level $k-1$. Each triad node in Level $k-1$ is either rewritten as two consecutive nodes in Level k or it is not rewritten at all. The node in Level $k-1$ is called the *parent node* of the two consecutive *daughter nodes* in Level k that result from the rewrite. If a triad node is not rewritten in a given level, then it cannot be rewritten in any later levels.

If a node n_p in Level $k-1$ is rewritten as two daughter nodes, n_{d1} and n_{d2} , in Level k such that n_{d2} follows n_{d1} , then the literal triad of n_{d2} must be the same as that of n_p .⁴⁶⁶ The triad function of n_{d1} must be a member of a set of possible values that depends upon

⁴⁶⁶ There is one exception to this—tonic triads in the home key can in certain situations be rewritten as a triad followed by a rest. See, for example, the end of Level 7 in Figure 39-3.

the triad function of n_p and the global variable, *mode*. For example, if the triad function of n_p is

$$\langle \text{ii-min, ...} \rangle$$

and the value of the global variable *mode* is *maj*, then the triad function of n_{d1} must be one of the following:

$$\begin{aligned} &\langle \text{ii-min, ...} \rangle \text{ (i.e. the same as that of } n_p) \\ &\langle \text{vi-min, ...} \rangle \\ &\langle \text{vi-maj, ii-min, ...} \rangle \\ &\langle \text{v-maj, ii-min, ...} \rangle \\ &\langle \text{iv-maj, ...} \rangle \\ &\langle \text{i-maj, ...} \rangle \end{aligned}$$

Each of the triad node rewrite rules in IOTA can be applied with either of two *rewrite parities*. The rewrite parity of any given particular application of a rewrite rule must be either 0 or 1.⁴⁶⁷

Let n_p be a parent node in Level $k-1$ which is rewritten as two consecutive daughter nodes, n_{d1} , n_{d2} in Level k such that n_{d2} follows n_{d1} . If the rewrite parity of this rewrite is 0, then:

1. the initial location of n_{d1} must be the same as the initial location of n_p ;
2. the terminal location of n_{d2} must be the same as the terminal location of n_p ; and
3. the terminal location of n_{d1} must be metrically the second strongest location in the node segment of n_p , where the metric strength of a location is determined using the metric strength function defined in section 38.2.1.2 above. (The strongest location in the node segment of n_p will be the initial location of n_p .)

If the rewrite parity of this same rewrite is 1, then:

1. the initial location of n_{d1} must be the metrically second strongest location in the node segment of the node in Level k whose terminal location is the same as the initial location of n_p ; and
2. the node segment of n_{d2} is made equal to the node segment of n_p .

In Figure 39-3, each rewritten parent node is connected by a line segment to the daughter node that inherits its literal triad. Each of these line segments is labelled with either a 0 or a 1 to indicate the rewrite parity.

Once the triad structure has been generated, the derivation algorithm then attempts to derive an *inversion structure*. The triad structure is first divided into its component phrases as indicated by the phrase segmentation. The program then works backwards through each phrase independently, attempting to assign a legal inversion to each triad node. The program also determines for each node whether or not the tenor and bass voices may cross. Let n_1 and n_2 be any pair of consecutive triad nodes within a phrase such that n_1 precedes n_2 . Assuming that n_2 has just been assigned its inversion

⁴⁶⁷ There are two exceptions to this. If a triad is rewritten as the same triad preceded by a rest then the rewrite parity must be 0. If a triad is rewritten as the same triad *followed* by a rest, then the rewrite parity is defined to be 0 but, of course, it is the *first* daughter node and not the second that must have the same literal triad as the parent node.

and the program is currently attempting to assign an inversion to n_1 , then whether or not a particular inversion k can be assigned to n_1 depends upon:

1. whether or not n_1 and n_2 are major or minor triads;
2. the interval between the roots of n_1 and n_2 ;
3. the inversion that has been assigned to n_2 ; and
4. the pitch interval that would result between the lowest notes of n_1 and n_2 if inversion k were assigned to n_1 .

Once the inversion structure has been generated, the derivation algorithm attempts to generate an *upper inner note structure*. An object is an upper inner note structure if and only if it is a structure in the sense defined in section 37.2.4 in which each node is an *upper inner note node*:

$$\underline{v}^{\text{uin}} = \langle n_1^{\text{uin}}, n_2^{\text{uin}}, \dots, n_k^{\text{uin}}, \dots, n^{\text{uin,t}} \rangle$$

Each upper inner note node is an ordered pair as follows:

$$n_k^{\text{uin}} = \langle \underline{l}(n_k^{\text{uin}}), \langle p_{\text{soprano}}(n_k^{\text{uin}}), p_{\text{alto}}(n_k^{\text{uin}}), p_{\text{tenor}}(n_k^{\text{uin}}), p_{\text{bass}}(n_k^{\text{uin}}) \rangle \rangle$$

Given a triad structure,

$$\underline{v}^{\text{triad}} = \langle n_1^{\text{triad}}, n_2^{\text{triad}}, \dots, n_k^{\text{triad}}, \dots, n^{\text{triad,t}} \rangle$$

such that

$$n_k^{\text{triad}} = \langle \underline{l}(n_k^{\text{triad}}), \langle \underline{\tau}_{\text{fun}}(n_k^{\text{triad}}), t_{\text{lit}}(n_k^{\text{triad}}) \rangle \rangle$$

then any upper inner note structure

$$\underline{v}^{\text{uin}} = \langle n_1^{\text{uin}}, n_2^{\text{uin}}, \dots, n_k^{\text{uin}}, \dots, n^{\text{uin,t}} \rangle$$

derived from $\underline{v}^{\text{triad}}$ must satisfy the following conditions:

1. $\underline{l}(n_k^{\text{uin}}) = \underline{l}(n_k^{\text{triad}})$ for all n_k^{uin} ;
2. $|\underline{v}^{\text{triad}}| = |\underline{v}^{\text{uin}}|$;
3. the pitches $p_{\text{soprano}}(n_k^{\text{uin}}), p_{\text{alto}}(n_k^{\text{uin}}), p_{\text{tenor}}(n_k^{\text{uin}}), p_{\text{bass}}(n_k^{\text{uin}})$ must all be in the major or minor triad whose root genus name and type are given in the node attribute of n_k^{triad} .

In addition, the pitches in the node attribute of an upper inner note node must satisfy a large number of voice-leading constraints. From this definition, it can be seen that an upper inner note structure is essentially a completely homophonic, four-part piece in which each chord contains only the notes of a major or minor triad. The score in Figure 39-5 represents one of the upper inner note structures from which the derivation algorithm can generate the representation of chorale no.6 in Bach 1990.

The derivation of a well-formed representation is completed by ‘embellishing’ the simple, homophonic ‘chorale’ represented by the upper inner note structure by means of ‘elaboration’-type rules that insert neighbour notes, fill in thirds with ‘transition’ notes, combine repeated notes in a given voice into single notes, insert anticipations and suspensions and so on.

Figure 39-5

The musical score for Figure 39-5 is presented in three systems, each consisting of a grand staff (treble and bass clefs) in G minor (one flat) and 6/8 time. The first system (measures 1-3) features a steady bass line and a treble line with chords and eighth-note patterns. The second system (measures 4-6) includes a triplet of eighth notes in the treble staff at the beginning of measure 4. The third system (measures 7-9) concludes the piece with a final cadence in the treble staff and a sustained bass note in the final measure.

I have used the derivation algorithm outlined above to prove that the universal output set of $\gamma(T_{\text{Bach}})$ contains the representations of 22 of the 29 chorales in the corpus. I have not yet attempted to prove that the composing algorithm is capable of generating the other 7 chorales in the corpus. Therefore it is possible at this stage to claim that $\gamma(T_{\text{Bach}})$ has not yet been shown to undergenerate.

On the other hand, $\gamma(T_{\text{Bach}})$ certainly overgenerates. None of the seven well-formed representations generated so far by $\gamma(T_{\text{Bach}})$ would be determined to be acceptable by any acceptability algorithm defined in accordance with the specification given in section 3.8 above.

Figure 39-6 shows one of the seven pieces generated so far by $\gamma(T_{\text{Bach}})$.

The algorithmic style theory associated with T_{Bach} has therefore been refuted. However, there are a number of ways in which it might be possible to modify $\gamma(T_{\text{Bach}})$ to produce a new algorithmic style theory that is not so easily refuted.

First, $\gamma(T_{\text{Bach}})$ employs a large number of independent triad structure rewrite rules of the form

$$n_p \rightarrow n_{d1} n_{d2}$$

where if n_p is a node in Level $k-1$ in the derivation of a triad structure, then n_{d1} and n_{d2} are consecutive triad nodes in Level k such that the literal triad of n_{d2} is the same as that of n_p , and the triad node attribute of n_{d1} is a member of a set that depends *only* upon the triad node attribute of n_p and the mode of the piece as a whole. The set of possible values that can be taken by the triad node attribute of the first daughter node for a given parent node attribute and a given global mode was in each case derived directly from the corpus and is consequently rather ad hoc. The voice-leading rules employed in the generation of an upper inner note structure are also very ad hoc. However, I think it might be possible to find useful generalizations from these sets of ad hoc rules by investigating the properties of these rules from the perspective of the thirds relation digraphs.

Second, the form of the triad rewrite rules employed in $\gamma(T_{\text{Bach}})$ implies that the node attribute of any given triad node is independent of the node attribute of the triad node that precedes it. In other words, if one were to isolate the implied ‘chord grammar’ in the triad rewrite rules from the metric aspects of these rules, then this chord grammar would consist entirely of rules of the form,

$$A \rightarrow BA$$

where A is a triad in Level $k-1$ and B is a triad that emerges in Level k . The form of the triad rewrite rules therefore implies that the acceptability of a chord in a tonal context is independent of the chord that precedes it and that the functional significance of a chord resides entirely in the extent to which it prepares for what remains of the phrase in which it occurs. In the words of Lerdahl and Jackendoff, this assumption reflects ‘the intuition that tonal pieces are fundamentally goal-oriented.’⁴⁶⁸ However, I think that, other things being equal, better results would probably be obtained if the node attribute of a triad node were made to depend not only on the following node attribute but also

⁴⁶⁸ Lerdahl and Jackendoff 1983, 174.

on the preceding one. This would give rise to a system of triad rewrite rules whose associated non-metric chord grammar would consist of rules of the form

$$AB \rightarrow ACB$$

However, as I discussed at some length in Chapter 16, I do not think that an ‘exclusively recursive’ grammar such as that employed in $\gamma(T_{\text{Bach}})$ could ever successfully characterize the class of chord sequences in the style of a master composer. It seems plausible that such a grammar would, at least in the early stages of a derivation, employ rewrite rules that rewrite non-terminal symbols as sequences of terminal and non-terminal symbols. I am willing to admit, however, that in the final stages of a derivation, it might be fruitful to employ recursive, ‘elaborates’-type rules to embellish the structure generated by the ‘is-a’-type rules employed in the early stages of the derivation.

A third source of inadequacy in the composing algorithm $\gamma(T_{\text{Bach}})$ is the lack of control over the way in which embellishments are applied in the final stages of the algorithm. Indeed, neighbour notes and passing notes are not even constrained to be in the same key as the surrounding notes! Again, it might well be possible to employ notions deriving from the thirds relation digraphs to formulate improved embellishment rules that are more ‘intelligently’ context-sensitive.

Finally, all the rules used in $\gamma(T_{\text{Bach}})$ are absolute rules that merely generate a class of possibilities at each decision point in the algorithm. I have made no attempt in $\gamma(T_{\text{Bach}})$ to employ notions along the lines of heuristics or preference rules to ‘lead the solution path away from a large number of unmusical patterns.’⁴⁶⁹ However, as discussed in section 7.4 above, composing algorithms that employ heuristics in conjunction with absolute rules generally perform better than those that employ absolute rules alone.

⁴⁶⁹ Ebcioğlu 1987b, 85.

Figure 39-6

Measures 1-6 of a musical score in 3/4 time, key of B-flat major. The score is written for piano with a grand staff. Measure 1 starts with a whole rest in the treble and a half note G2 in the bass. The melody in the treble consists of eighth notes: G4, A4, Bb4, A4, G4, F4, E4, D4. The bass line consists of eighth notes: G2, A2, Bb2, A2, G2, F2, E2, D2. Measure 6 ends with a fermata over the final note.

Measures 7-12 of the musical score. Measure 7 begins with a fermata over the final note of the previous system. The melody in the treble continues with eighth notes: C4, Bb3, A3, G3, F3, E3, D3. The bass line continues with eighth notes: C3, Bb2, A2, G2, F2, E2, D2. Measure 12 ends with a fermata over the final note.

Measures 13-18 of the musical score. Measure 13 begins with a fermata over the final note of the previous system. The melody in the treble continues with eighth notes: C4, Bb3, A3, G3, F3, E3, D3. The bass line continues with eighth notes: C3, Bb2, A2, G2, F2, E2, D2. Measure 18 ends with a fermata over the final note.

40 References

- Aho et al. 1983; Aho, Hopcroft and Ullman 1983:** Aho, A.V., Hopcroft, J.E. and Ullman, J.D.; 1983; *The design and analysis of computer algorithms*; Addison-Wesley, Menlo Park, Calif.
- Alphonse et al. 1988:** Alphonse, B., Pennycook, B. and Fujinaga, I.; 1988; ‘Optical music recognition: a progress report’; *Proceedings of the Conference for small computers in the arts*, pp.8–12.
- Ames 1987:** Ames, C.; 1987; ‘Automated composition in retrospect: 1956–1986’; *Leonardo*, Vol.20, No.2, pp.169–185.
- Ames 1992:** Ames, C.; 1992; ‘Quantifying Musical Merit’; *Interface*, Vol.21, pp.53–93.
- Ames and Domino 1992:** Ames, C. and Domino, M.; 1992; ‘Cybernetic Composer: An overview’; in: Balaban, M., Ebcioglu, K. and Laske, O. (eds.); *Understanding Music with AI: Perspectives on Music Cognition*; AAAI Press/MIT Press, Cambridge, Mass.; pp.186–205.
- Bach 1941:** Bach, J.S. (ed. Riemenschneider, A.); 1941; *371 Harmonized Chorales and 69 Chorale Melodies with figured bass*; Schirmer; New York.
- Bach 1990:** Bach, J.S. (ed. Schubert, K.); 1990; *371 Four-Part Chorales*; Breitkopf and Härtel, Wiesbaden.
- Balzano 1980:** Balzano, G.J.; 1980; ‘The group-theoretic description of 12-fold and microtonal pitch systems’; *Computer Music Journal*, Vol.4, No.4, pp.66–84.
- Baroni 1981:** Baroni, M.; 1981; ‘Sulla nozione della grammatica musicale’; *Rivista Italiana di Musicologia*, Vol.16, No.2, pp.240–279.
- Baroni 1983:** Baroni, M.; 1983; ‘The concept of musical grammar’; *Music Analysis*, Vol.2, No.2, pp.175–208.
- Baroni and Callegari 1984:** Baroni, M. and Callegari, L.; 1984; ‘Antiche canzoni francesi: uno studio di metrica generativa’; *Quaderni di informatica musicale*, Vol.5.
- Baroni and Jacoboni 1973:** Baroni, M. and Jacoboni, C.; 1973; ‘Analysis and generation of Bach’s Chorale melodies’; *Premier Congrès Internationale de Sémiotique Musicale*, Belgrade; pp.125–134.
- Baroni and Jacoboni 1975:** Baroni, M. and Jacoboni, C.; 1975; ‘Analysis and generation of Bach’s chorale melodies’; in: Stefani, G. (ed.); *Proceedings of the First International Congress on the Semiotics of Music*, Centro di Iniziativa Culturale, Pesaro, Italy.
- Baroni and Jacoboni 1976:** Baroni, M. and Jacoboni, C.; 1976; ‘Verso una Grammatica Della Melodia’; Università Studi de Bologna, Bologna.
- Baroni and Jacoboni 1978:** Baroni, M. and Jacoboni, C.; 1978; ‘Proposal for a grammar of melody: the Bach Chorales’; Les Presses de l’Université de Montreal, Montreal.

- Baroni and Jacoboni 1983:** Baroni, M. and Jacoboni, C.; 1983; 'Computer generation of melodies: further proposals'; *Computers and the Humanities*, Vol.17, No.1, pp.1–18.
- Baroni et al. 1984; Baroni, Brunetti, Callegari and Jacoboni 1984:** Baroni, M., Brunetti, R., Callegari, L. and Jacoboni, C.; 1984; 'A grammar for melody: relationships between melody and harmony'; in: Baroni, M. and Callegari, L. (eds.); *Musical Grammars and Computer Analysis*; Olschki, Florence; pp.201–218.
- Baroni et al. 1989; Baroni, Dalmonte and Jacoboni 1989:** Baroni, M., Dalmonte, R. and Jacoboni, C.; 1989; 'Relationships between music and poetry in the arias of Giovanni Legrenzi'; *Contemporary Music Review*, Vol.4, pp.23–44.
- Baroni et al. 1992; Baroni, Dalmonte and Jacoboni 1992:** Baroni, M., Dalmonte, R. and Jacoboni, C.; 1992; 'Theory and analysis of European melody'; in: Marsden, A. and Pople, A. (eds.); *Computer Representations and Models in Music*; Academic Press, London; pp.187–206.
- Baroni et al. 1992b:** Baroni, M., Dalmonte, R. and Jacoboni, C.; 1992; 'Analysis via synthesis: the program LEGRE'; in: Dalmonte, R. and Baroni, M. (eds.); *Proceedings of the Secondo Convegno Europeo di Analisi Musicale*; Università degli Studi di Trento, Trento; pp.599–607.
- Belfiore 1985:** Belfiore, A.; 1985; 'Algomusic: an algorithm for the genesis of musical structures'; in: Blum, T. and Strawn, J. (eds.); *Proceedings of the Venice 1982 International Computer Music Conference*; Computer Music Association, San Francisco, California.
- Bernstein 1976:** Bernstein, L.; 1976; *The Unanswered Question*; Harvard University Press, Cambridge, Mass.
- Bolognesi 1983:** Bolognesi, T.; 1983; 'Automatic composition: experiments with self-simulating music'; *Computer Music Journal*, Vol.7, No.1, pp.25–36.
- Boretz 1969:** Boretz, B.; 1969; 'Meta-variations: studies in the foundations of musical thought'; *Perspectives of New Music*, Vol.8, No.1, pp.1–74.
- Boretz 1970; Boretz 1970a:** Boretz, B.; 1970; 'Sketch of a musical system' ('Meta-variations', Part 2); *Perspectives of New Music*, Vol.8, No.2, pp.49–111.
- Boretz 1970b:** Boretz, B.; 1970; 'Meta-variations,' Part 3, Section 1; *Perspectives of New Music*, Vol.9, No.1, pp.23–42.
- Boretz 1971:** Boretz, B.; 1971; 'Meta-variations,' Part 3, Section 2; *Perspectives of New Music*, Vol.9, No.2 and Vol.10, No.1, pp.232–270.
- Boretz 1972:** Boretz, B.; 1972; 'Meta-variations,' Part 4, Section 1; *Perspectives of New Music*, Vol.11, No.1, pp.146–223.
- Boretz 1973:** Boretz, B.; 1973; 'Meta-variations,' Part 4, Section 2; *Perspectives of New Music*, Vol.11, No.2, pp.156–203.
- Borowski and Borwein 1989:** Borowski, E.J. and Borwein, J.M.; 1989; *Collins Dictionary of Mathematics*; HarperCollins, Glasgow.
- Brown and Dempster 1989:** Brown, M. and Dempster, D.J.; 1989; 'The scientific image of music theory'; *Journal of Music Theory*, Vol.33, No.1, pp.65–106.

- Busby 1818:** Busby, T.; 1818/1976; *A Grammar of Music: To which are prefixed Observations Explanatory of the Properties and Powers of Music as a Science and of the General Scope and Object of the Work*; Da Capo; New York.
- Camilleri 1992:** Camilleri, L.; 1992; 'Computational theories of music: theoretical and applicative issues'; in: Marsden, A. and Pople, A. (eds.); *Computer Representations and Models in Music*; Academic Press, London; pp.171–185.
- Carter and Bacon 1990:** Carter, N.P. and Bacon, R.A.; 1990; 'Automatic recognition of music notation'; in: *International Association for Pattern Recognition Workshop on Syntactic and Structural Pattern Recognition*; Murray Hill, New Jersey.
- Chomsky 1957:** Chomsky, N.; 1957; *Syntactic Structures*; Mouton, The Hague.
- Chomsky 1959:** Chomsky, N.; 1959; 'Review of Skinner's *Verbal Behavior*'; *Language*, Vol.35, No.1, pp.26–58.
- Chomsky 1965:** Chomsky, N.; 1965; *Aspects of the Theory of Syntax*; MIT Press, Cambridge, Mass.
- Chopin 1957:** Chopin, F. (ed. Brugnoli and Montani); *Studies for Piano*; Ricordi, Milan.
- Clarke, Brown and Thorne 1990:** Clarke, A.T., Brown, B.M. and Thorne, M.P.; 1990; 'Problems to be faced by developers of computer-based automatic music recognisers'; in: *Proceedings of the International Computer Music Conference 1990*; Glasgow.
- Coker 1964:** Coker, J.; 1964; *Improvising Jazz*; Prentice Hall, New Jersey.
- Cook 1987:** Cook, N.; 1987; 'The perception of large-scale tonal closure'; *Music Perception*, Vol.5, pp.197–205.
- Cook 1989:** *Journal of Music Theory*, Vol.33, No.1, pp.65–106.
- Cope 1987a:** Cope, D.; 1987; 'An expert system for computer-assisted composition'; *Computer Music Journal*, Vol.11, No.4, pp.30–46.
- Cope 1989a; Cope 1989:** Cope, D.; 1989; Experiments in musical intelligence (EMI): non-linear linguistic-based composition'; *Interface*, Vol.18, No.1–2, pp.117–139.
- Cope 1991:** Cope, D.; 1991; *Computers and Musical Style (Computer Music and Digital Audio Series, 6)*; Oxford University Press, Oxford.
- Cope 1993:** Cope, D.; 1993; 'A computer model of music composition'; in: Schwanauer, S. and Levitt, D. (eds.); *Machine Models of Music*; MIT Press, Cambridge, Mass.; pp.403–425.
- Cross 1995a:** e-mail communication, 7/12/95 16:24:02 +0000.
- Cross 1996a:** e-mail communication, 21/596 17:39:56 +0100.
- Cross 1996b:** e-mail communication, 20/1/96 16:39:20 +0000.
- Cross 1996c:** e-mail communication, 21/1/96 16:09:36 +0000.
- Czerny 1979:** Czerny, C. (tr. Bishop, J.); 1979; *School of Practical Composition*; Da Capo Press, New York.
- D'Indy 1912:** D'Indy, V.; 1912; *Cours de Composition Musicale*; Durand, Paris.

- Deliège 1987:** Deliège, I.; 1987; 'Grouping conditions in listening to music: an approach to Lerdahl and Jackendoff's Grouping Preference Rules'; *Music Perception*, Vol.4, No.4, pp.325–360.
- Dennett 1991:** Dennett, D.C.; 1991; *Consciousness Explained*; Penguin, London.
- Deutsch 1982:** Deutsch, D.; 1982; 'The processing of pitch combinations'; in: Deutsch, D. (ed.); *The Psychology of Music*; Academic Press, London; pp.271–317.
- Deutsch and Feroe 1981:** Deutsch, D. and Feroe, J.; 1981; 'The internal representation of pitch sequences in tonal music'; *Psychological Review*, Vol.88, No.6, pp.503–522.
- Drabkin 1984:** Drabkin, W.; 1984; 'Review of *A Generative Theory of Tonal Music*'; *The Musical Times*, Vol.125, p.273.
- Durand 1898:** Durand, E.; 1898?; *Traité de Composition Musicale*; Leduc, Paris.
- Ebcioglu 1982:** Ebcioglu, K.; 1982; 'Computer counterpoint'; in: Howe, H.S. (ed.); *Proceedings of 1980 International Computer Music Conference*; Computer Music Association, San Francisco, California; pp.534–543.
- Ebcioglu 1985:** Ebcioglu, K.; 1985; 'An expert system for Schenkerian synthesis of Chorales in the style of J.S. Bach'; in: Buxton, W. (ed.); *Proceedings of International Computer Music Conference 1984*; Computer Music Association, San Francisco, California; pp.135–142.
- Ebcioglu 1987; Ebcioglu 1987b:** Ebcioglu, K.; 1987; 'Report on the Choral Project: An expert system for Chorale harmonization', (Research report no. RC 12628); I.B.M., Thomas J. Watson Research Centre, Yorktown Heights, N.Y.
- Ebcioglu 1988:** Ebcioglu, K.; 1988; 'An expert system for harmonizing Bach Chorales'; *Computer Music Journal*, Vol.12, No.3, pp.43–51.
- Ebcioglu 1990:** Ebcioglu, K.; 1990; 'An expert system for harmonizing chorales in the style of J.S. Bach'; *Journal of Logic Programming*, Vol.8, Nos.1–2, pp.145–185.
- Ebcioglu 1992:** Ebcioglu, K.; 1992; 'An expert system for harmonizing Chorales in the style of J.S. Bach'; in: Balaban, M., Ebcioglu, K. and Laske, O. (eds.); *Understanding Music with AI: Perspectives on Music Cognition*; AAAI Press/MIT Press, Cambridge, Mass.; pp.294–333.
- Ebcioglu 1996, Ebcioglu 1996a:** e-mail communication, 2/1/96 04:48:10 EST.
- Ebcioglu 1996b:** e-mail communication, 19/2/96 13:27:11 EST.
- Feld 1974:** Feld, S.; 1974; 'Linguistic models in ethnomusicology'; *Ethnomusicology*, Vol.18, No.2, pp.197–217.
- Forte 1973:** Forte, A.; 1973; *The Structure of Atonal Music*; Yale University Press; New Haven.
- Gill 1963:** Gill, S.; 1963; 'A technique for the composition of music in a computer'; *Computer Journal*, Vol.6, No.2.
- Haflich 1977:** Haflich, S.M.; 1977; 'Harmony, Counterpoint, and Free Composition: Towards a Synthesis of the Tonal Common Practice'; Doctoral dissertation prospectus, Department of Music, Yale University.

- Harvey 1985:** Harvey, D.; 1985; 'Review of Lerdahl and Jackendoff's *A Generative Theory of Tonal Music*'; *Music Analysis*, Vol.4, No.3, pp.292–303.
- Helmholtz 1954:** Helmholtz, H. (tr. and ed. Ellis, A.J.); 1885/1954; *On the Sensations of Tone*; Dover, N.Y.
- Hofstadter 1979:** Hofstadter, D.R.; 1979; *Gödel, Escher, Bach: An Eternal Golden Braid*; Penguin.
- Hofstadter 1982:** Hofstadter, D.R.; 1982; 'Metafont, Metamathematics, and Metaphysics'; Technical Report No.136, Indiana University Computer Science Department.
- Holtzman 1980:** Holtzman, S.R.; 1980; 'A generative grammar definition language for music'; *Interface*, Vol.9, pp.1–47.
- Holtzman 1981:** Holtzman, S.R.; 1981; 'Using generative grammars for music composition'; *Computer Music Journal*, Vol.5, No.1, pp.51–64.
- Hughes 1991:** Hughes, D.W.; 1991; 'Grammars of non-western musics: a selective survey'; in: Howell, P., West, R. and Cross, I. (eds.); *Representing Musical Structure*; Academic Press; London; pp.327–362.
- Jackendoff 1977:** Jackendoff, R.; 1977; 'Review of Bernstein, *The Unanswered Question*'; *Language*, Vol.53, No.4, pp.883–894.
- Jackendoff and Lerdahl 1980:** Jackendoff, R. and Lerdahl, F.; 1980; 'Discovery procedures vs. rules of musical grammar in a generative music theory'; *Perspectives of New Music*, Vol.18, No.2, pp.503–510.
- Jackendoff and Lerdahl 1981:** Jackendoff, R. and Lerdahl, F.; 1981; 'Generative music theory and its relation to psychology'; *Journal of Music Theory*, Vol.25, No.1, pp.45–90.
- Jackendoff and Lerdahl 1982:** Jackendoff, R. and Lerdahl, F.; 1982; 'A grammatical parallel between music and language'; in: Clynes, M. (ed.); *Music, Mind and Brain: the Neuropsychology of Music*; Plenum, New York; pp.83–117.
- Johnson-Laird 1991:** Johnson-Laird, P.N.; 1991; 'Jazz improvisation: a theory at the computational level'; in: Howell, P., West, R. and Cross, I. (eds.); *Representing Musical Structure*; Academic Press, London; pp.291–325.
- Joplin 1981:** Joplin, S. (ed. Brodsky-Lawrence, V.); 1981; *Complete Piano Works*; New York Public Library.
- Kassler 1963:** Kassler, M.; 1963; 'A sketch of the use of formalized languages for the assertion of music'; *Perspectives of New Music*, Vol.1, No.2, pp.83–94.
- Kassler 1967:** Kassler, M.; 1967; 'A Trinity of Essays'; University of Princeton, Ph.D. Thesis; University Microfilms International, 68–2490.
- Kassler 1975:** Kassler, M.; 1975; 'Proving Musical Theorems I: The Middleground of Heinrich Schenker's Theory of Tonality' (Technical Report, No.103); Basser Department of Computer Science, University of Sydney.
- Kassler 1976:** Kassler, M.; 1976; 'The decidability of languages that assert music'; *Perspectives of New Music*, Vol.14, No.2, pp.249–251.

- Keiler 1978:** Keiler, A.; 1978; ‘Bernstein’s “Unanswered Question”’; *Musical Quarterly*, Vol.62, No.2, pp.195–222.
- Keiler 1981a:** Keiler, A.; 1981; ‘Music as metalanguage: Rameau’s fundamental bass’; in: Browne, R. (ed.); *Music Theory: Special Topics*; Academic Press, London.
- Keiler 1981b:** Keiler, A.; 1981; ‘Two views of musical semiotics’; in: Steiner, W. (ed.); *The Sign in Music and Literature*; University of Texas Press, Austin, Texas.
- Kendall 1981:** Kendall, G.S.; 1981; ‘Composing from a geometric model: five-leaf rose’; *Computer Music Journal*, Vol.5, No.4.
- Kippen and Bel 1992:** Kippen, J. and Bel, B.; 1992; ‘Modelling music with grammars: formal language representation in the BOL Processor’; in: Marsden, A. and Pople, A. (eds.); *Computer Representations and Models in Music*; Academic Press, London; pp.207–238;
- Koffka 1935:** Koffka, K.; 1935; *Principles of Gestalt Psychology*; Harcourt, Brace and World, N.Y.
- Köhler 1940:** Köhler, W.; 1940; *Dynamics in Psychology*; Liveright, N.Y.
- Krumhansl 1985:** Krumhansl, C.L.; 1985; ‘Perceiving tonal structure in music’; *American Scientist*, Vol.73, pp.371–8.
- Kuhn 1970:** Kuhn, T.S.; 1970; *The Structure of Scientific Revolutions* (2nd. Edition); University of Chicago Press, Chicago, IL.
- Lakatos 1970:** Lakatos, I.; 1970; ‘The structure of scientific research programmes’; in: Lakatos, I. and Musgrave, A. (eds.); *Criticism and the Growth of Knowledge*; Cambridge University Press, Cambridge.
- Laske 1973:** Laske, O.; 1973; ‘In search of a generative grammar for music’; *Perspectives of New Music*, Vol.12, No.1, pp.351–378.
- Laske 1980:** Laske, O.; 1980; ‘Toward an explicit cognitive theory of musical listening’; *Computer Music Journal*, Vol.4, No.2, pp.73–83.
- Lees 1960:** Lees, R.B.; 1960; *The Grammar of English Nominalizations*; Mouton, The Hague.
- Lerdahl and Jackendoff 1977:** Lerdahl, F. and Jackendoff, R.; 1977; ‘Toward a formal theory of tonal music’; *Journal of Music Theory*, Vol.21, No.1, pp.111–171.
- Lerdahl and Jackendoff 1983:** Lerdahl, F. and Jackendoff, R.; 1983; *A Generative Theory of Tonal Music*; MIT Press, Cambridge, MA.
- Lerdahl and Jackendoff 1985:** Lerdahl, F. and Jackendoff, R.; 1985; ‘A reply to Peel and Slawson’s review of *A Generative Theory of Tonal Music*’; *Journal of Music Theory*, Vol.29, No.1, pp.145–160.
- Levitt 1993:** Levitt, D.A.; 1993; ‘A representation for musical dialects’; in: Schwanauer, S.M. and Levitt, D.A. (eds.); *Machine Models of Music*; MIT Press, Cambridge, Mass.; pp.454–468.
- Lidov and Gabura 1973:** Lidov, D. and Gabura, J.; 1973; ‘A melody writing algorithm using a formal language model’; *Computer Studies in the Humanities and Verbal Behaviour*, Vol.4, No. 2, pp.138–148.

- Lippus 1980:** Lippus, U.; 1980; 'On the formal description of monodic melodies' (Preprint KKI-13); *Fifth International Finno-Ugric Congress*, Turku, Finland.
- Longuet-Higgins 1976:** Longuet-Higgins, H.C.; 1976; 'Perception of melodies'; *Nature*, No.263 (Oct 21, 1976), pp.646-653.
- Longuet-Higgins 1979:** Longuet-Higgins, H.C.; 1979; 'The perception of music'; *Proceedings of the Royal Society*, Vol.B205, pp.307-322.
- Longuet-Higgins and Steedman 1971:** Longuet-Higgins, H.C. and Steedman, M.; 1971; 'On interpreting Bach'; *Machine Intelligence*, Vol.6, pp.221-241.
- Lyons 1977:** Lyons, J.; 1977; *Chomsky*; Fontana, London.
- Maxwell 1988:** Maxwell, H.J.; 1988; 'An expert system for harmonic analysis of tonal music'; *Proceedings of the First Workshop on AI and Music (AAAI-88)*; AAAI, Minneapolis/St.Paul, MN.; pp.20-33.
- Mazzola et al. 1989:** Mazzola, G., Wieser, H.G., Brunner, V. and Muzzulini, D.; 1989; 'A symmetry-oriented mathematical model of classical counterpoint and related neurophysiological investigations by depth EEG'; *Computers and Mathematics with Applications*, Vol.17, Nos.4-6, pp.539-594.
- Meredith 1993:** Meredith, D.; 1993; 'A computer-aided comparison of syntax systems in three piano pieces by Debussy'; in: Cross, I. and Deliège, I. (eds.); *Contemporary Music Review*, Vol.9, Nos.2-3, pp.284-307.
- Miller 1956:** Miller, G.A.; 1956; 'The magical number seven, plus or minus two: some limits on our capacity for processing information'; *The Psychological Review*, Vol.63, pp.81-97.
- Moore 1989:** Moore, B.C.J.; 1989; *An Introduction to the Psychology of Hearing* (3rd. edition)); Academic Press. London.
- Moore and Carling 1982:** Moore, T. and Carling, C.; 1982; *Understanding Language: Towards a Post-Chomskyan Linguistics* (1st Edition); Macmillan, London.
- Moorer 1972:** Moorer, J.A.; 1972; 'Music and computer composition'; *Communications of the Association for Computing Machinery*, Vol.15, No.2, pp.104-113.
- Mouton and Pachet 1995:** Mouton, R. and Pachet, F.; 1995; 'The Symbolic vs. Numeric Controversy in Automatic Analysis of Music'; in: Widmer, G.(ed.); *Artificial Intelligence and Music (IJCAI-95 Workshop Program Working Notes)*; pp.32-40.
- Narmour 1977:** Narmour, E.; 1977; *Beyond Schenkerism: The Need for Alternatives in Music Analysis*; University of Chicago Press, Chicago, IL.
- Narmour 1990:** Narmour, E.; 1990; *The Analysis and Cognition of Basic Harmonic Structures: the Implication-realization Model*; University of Chicago Press, Chicago.
- Narmour 1992:** Narmour, E.; 1992; *The Analysis and Cognition of Melodic Complexity: the Implication-Realization Model*; University of Chicago Press, Chicago.
- Nattiez 1975:** Nattiez, J.-J.; 1975; *Fondements d'une sémiologie de la musique*; Union Générale d'Éditions, Paris.

- Neisser 1967:** Neisser, U.; 1967; *Cognitive Psychology*; Prentice-Hall, Englewood Cliffs, N.J.
- Ore 1962:** Ore, O.; 1962; *Theory of Graphs* (American Mathematical Society Colloquium Publications, Volume XXXVIII); American Mathematical Society Colloquium Publications, Providence, Rhode Island.
- Peel and Slawson 1984:** Peel, J. and Slawson S.; 1984; ‘Review of *A Generative Theory of Tonal Music*, by Fred Lerdahl and Ray Jackendoff’; *Journal of Music Theory*, Vol.28, No.2, pp.271–294.
- Peel and Slawson 1985:** Peel, J. and Slawson, W.; *Reply to a reply*; *Journal of Music Theory*, Vol.29, No.1, pp.161–167.
- Pelinski 1984:** Pelinski, R.; 1984; ‘A generative grammar of personal Eskimo songs’; in: Baroni, M. and Callegari, L. (eds.); *Musical Grammars and Computer Analysis*; Olschki, Florence; pp.273–286.
- Penrose 1994:** Penrose, R.; 1994; *Shadows of the Mind*; Oxford University Press, Oxford.
- Popper 1983:** Popper, K.R.; 1983; *Realism and the Aim of Science* (Postscript to *The Logic of Scientific Discovery*, Volume 1); Hutchinson, London.
- Powers 1980:** Powers, H.; 1980; ‘Language models and musical analysis’; *Ethnomusicology*; Vol.24, No.1, pp.1–60.
- Rader 1974:** Rader, G.M.; 1974; ‘A method for composing simple traditional music by computer’; *Communications of the Association for Computing Machinery*, Vol.17, No.11, pp.631–38.
- Roads 1979:** Roads, C.; 1979; ‘Grammars as representations for music’; *Computer Music Journal*; Vol.3, No.1, pp.48–55.
- Roads 1985:** Roads, C.; 1985; ‘Grammars as representations for music’; in: Roads, C. and Strawn, J. (eds.); *Foundations of Computer Music*; MIT Press, Cambridge, Mass.; pp.403–442.
- Rothgeb 1966:** Rothgeb, J.E.; 1966; ‘Some uses of mathematical concepts in theories of music’; *Journal of Music Theory*, Vol.10, pp.200–215.
- Rothgeb 1968:** Rothgeb, J.E.; 1968; ‘Harmonizing the Unfigured Bass: A Computational Study’; Unpublished Ph.D. Dissertation; Yale University, New Haven, Connecticut.
- Rothgeb 1980:** Rothgeb, J.E.; 1980; ‘Simulating musical skills by digital computer’; *Computer Music Journal*, Vol.4, No.2, pp.30–40.
- Schenker 1979:** Schenker, H. (tr. Oster, E.); *Free Composition*; Longman, New York.
- Schoenberg 1969:** Schoenberg, A.; 1969; *Structural Functions of Harmony*; Faber and Faber, London.
- Schottstaedt 1984:** Schottstaedt, W.; 1984; ‘Automatic Species Counterpoint’ (CCRMA Report no. STAN–M–19); Stanford University Center for Computational Research in Music and Acoustics, California.

- Schottstaedt 1989:** Schottstaedt, W.; 1989; 'Automatic counterpoint'; in: Mathews, M.V. and Pierce, J.R. (eds.); *Current Directions in Computer Music Research*; MIT Press, Cambridge, MA.; pp.199–214.
- Shepard 1982:** Shepard, R.N.; 1982; 'Structural representations of musical pitch'; in: Deutsch, D.; *The Psychology of Music*; Academic Press, London; pp.344–390.
- Sinclair et al. 1991:** Sinclair, J.M. et al. (eds.); 1991; *Collins English Dictionary* (3rd Edition); HarperCollins, Glasgow.
- Smoliar 1974:** Smoliar, S.W.; 1974; 'Process structuring and music theory'; *Journal of Music Theory*, Vol.18. pp.308–336.
- Smoliar 1976:** Smoliar, S.W.; 1976; 'An approach to music theory through computational linguistics'; *Journal of Music Theory*, Vol.20, No.1, pp.105–131.
- Smoliar 1980:** Smoliar, S.W.; 1980; 'A computer aid for Schenkerian analysis'; *Computer Music Journal*, Vol.4, No.2, pp.41–48.
- Snell 1979:** Snell, J.L.; 1979; 'Design for a formal system for deriving tonal music' (Master's theses (State University of New York at Binghamton) no. 668); M.A. Dissertation, State University of New York at Binghamton, Dept. of Music.
- Steedman 1984:** Steedman, M.J.; 1984; 'A generative grammar for jazz chord sequences'; *Music Perception*, Vol.2, No.1, pp.52–77.
- Sundberg and Lindblom 1976:** Sundberg, J. and Lindblom, B.; 1976; 'Generative theories in language and music descriptions'; *Cognition*, Vol.4, pp.99–122.
- Sundberg and Lindblom 1993:** Sundberg, J. and Lindblom, B.; 1993; 'Generative theories in language and music descriptions'; in: Schwanauer, S.M. and Levitt, D.A. (eds.); *Machine Models of Music*; MIT Press, Cambridge, Mass; pp.262–286.
- Turing 1950:** Turing, A. M.; 1950; 'Computing machinery and intelligence'; *Mind*, Vol.59, pp.433–460.
- Twaddell 1935:** Twaddell, W.F.; 1935; 'On Defining the Phoneme' (Language Monograph No.16); reprinted in part in: Joos, M. (ed.); 1957; *Reading in Linguistics*; Washington.
- Wenck 1988:** Wenck, A.; 1988; 'Parsing Debussy'; in: Boroda, M.G. (ed.); *Musikometrika, I. Quantitative Linguistics*, vol.37; Studienverlag Dr. N. Brockmeyer, Bochum; pp.237–256.
- Widmer 1995:** Widmer, G.; 1995; 'Modeling the rational basis of musical expression'; *Computer Music Journal*, Vol.19, No.2, pp.76–96.
- Widmer 1996:** Widmer, G.; 1996; 'Learning expressive performance: the structure-level approach'; unpublished research paper, Dept. of Medical Cybernetics and Artificial Intelligence, University of Vienna, and Austrian Research Institute for Artificial Intelligence, Schottengasse 3, A–1010, Vienna, Austria.
- Wilson 1979:** Wilson, R.J.; 1979; *Introduction to Graph Theory* (2nd Edition); Longman; London.
- Winograd 1968:** Winograd, T.; 1968; 'Linguistics and the computer analysis of tonal harmony'; *Journal of Music Theory*, Vol.12, No.2, pp.2–49.

Xenakis 1971: Xenakis, I.; 1971; *Formalized Music*; Indiana University Press, Bloomington.