

**Automatic Harmonic Analysis of
Jazz Chord Progressions Using a
Musical Categorical Grammar**

Mark Wilding



Master of Science
School of Informatics
University of Edinburgh
2008

Abstract

This study concerns the use of formal grammars commonly applied to language to model the process of harmonic analysis and the human understanding of the language of jazz harmony. It builds on the Combinatory Categorical Grammar (CCG) of Steedman (1996) for jazz chord sequences. The coverage of the grammar is extended and its semantic productions based on Longuet-Higgins' tonal space theory are developed according to literature on functional harmonic analysis and examination of bodies of jazz chord sequences.

A language of underspecified harmonic semantic expressions is developed that can be used to express generalizations over movements in Longuet-Higgins' tonal space. The language is applied successfully to the problem of recognizing chord sequences that are variations on a general harmonic form; in particular, it is used to recognize examples of the 12-bar blues.

A parser for the harmonic grammar has been implemented and applied to jazz chord sequences. The grammar is evaluated with respect to its applicability to jazz standards outside the domain of the blues. Shortcomings of the grammar as a model of musical analysis are discussed and suggestions are made for future development. Some further examples are considered from outside the domain of jazz standards. The high lexical ambiguity of the grammar calls for statistic approaches similar to those used for natural language parsing. These are discussed but not implemented, due to the current lack of a suitable annotated corpus. The grammatical approach to harmonic analysis appears to provide a good means for modelling human perception of jazz chord sequences, which promises to generalize well to interpretation of harmony in a wider spectrum of Western tonal music.

Acknowledgements

I would like to thank Mark Steedman for all his guidance and succinct, indispensable feedback throughout the project. I would also like to thank John Elliott for his collaboration at various stages.

I must also express my gratitude to Hanna for lending a scientist's and musician's ear to my ceaseless prattling, reading and commenting on the finished work and much support besides throughout the project. As always, thanks to Leonie for super-accurate, fast, free, professional proofreading.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Mark Wilding)

Table of Contents

1	Introduction	1
1.1	Music as a Language	1
1.2	Harmony	3
1.3	Jazz	4
1.4	Combinatory Categorical Grammar	5
1.5	The Present Work	6
2	Grammar Development	9
2.1	Introduction	9
2.2	The Original Grammar	9
2.2.1	Chords and Chord Classes	10
2.2.2	The Original Categories Explained	12
2.2.3	New Optional Minor Notation	14
2.2.4	The Original Grammar Rewritten	15
2.3	Minor Resolution in Authentic Cadences	15
2.4	Tritone Substitutions	16
2.4.1	Major Tritone Substitutions	16
2.4.2	New Tritone Substitution Categories	16
2.5	Similar-Chord Sequences	19
2.5.1	New 0-Categories	19
2.5.2	Major-Minor Transitions	20
2.6	Cadential Slash Modes	20
2.7	Cadence-Raising Rules	23
2.7.1	New Semantics	23
2.7.2	Minor Cadences	24
2.8	Modulation	24
2.8.1	Introducing Modulations	24

2.8.2	Recognizing Modulations with the Grammar	25
2.8.3	The Blues as a Modulating Form	28
2.8.4	The Extreme Ambiguity of Modulations	28
2.9	Diminished Sevenths	30
2.9.1	Theoretical Background to the Chord's Ambiguity	30
2.9.2	The True Diminished Seventh	31
2.9.3	The Inverted Minor Ninth Chord	32
2.9.4	A Chord of Simultaneous Leading Notes	33
2.10	Temporal Information	34
2.10.1	Adding Temporal Information to the Signs	34
2.10.2	Propagation of Temporal Information	35
2.11	A New CCG Chord Grammar	37
3	Semantic Interpretation	41
3.1	Introduction	41
3.2	Underspecified Semantic Expressions	42
3.2.1	Language Definition	43
3.2.2	Temporal Constraints	46
3.3	The Twelve-Bar Blues	47
3.3.1	Characteristics of the Form	47
3.3.2	An Underspecified Semantic Expression	48
3.4	I Got Rhythm	48
3.5	Interpretation in the Longuet-Higgins Space	49
4	Jazz Parser Implementation	55
4.1	Introduction	55
4.2	Parsing Algorithm	55
4.3	Parser Tool	58
4.3.1	Basic Parser Input	58
4.3.2	Interactive Mode	59
4.3.3	File Input	61
4.3.4	Output Format	62
4.3.5	Underspecified Semantic Expression Matching	63
4.4	Chord Sequence Input	64
4.4.1	Textual Chord Input	64
4.4.2	Other Forms of Input	67

4.5	Grammar Specification	67
4.5.1	morph.xml	68
4.5.2	lexicon.xml	69
4.5.3	rules.xml	70
4.6	Statistical Parsing	70
4.6.1	Requirement of Statistical Methods	70
4.6.2	Extensibility of the Parser	71
4.7	Speeding Up Parsing	72
5	Results and Discussion	75
5.1	Introduction	75
5.1.1	Quantitative Evaluation	75
5.1.2	Isolating Chord Sequences Musically	77
5.2	Coverage of Blues Changes	79
5.3	Coverage of Rhythm Changes	79
5.4	Coverage of Jazz Standards	80
5.5	Coverage of Other Music	83
5.5.1	Beethoven, Op. 2, No. 1	83
5.5.2	Bach, BWV 553	85
5.6	Circularity	86
5.7	Analysis of Omissions	87
5.7.1	Blues Changes D and Rhythm Changes C Opening	88
5.7.2	Convergent Cadences Problem	89
5.7.3	Movements to Relative Majors and Minors	90
5.7.4	Movements Down a Major Third	92
5.7.5	Pennies Ending Problem	93
5.7.6	Abrupt Modulations	94
5.8	Is Solar a Blues?	95
6	Future Work and Conclusion	97
6.1	Future Work	97
6.2	Conclusion	98
A	Chord Sequences for Jazz Standards	101
A.1	Ain't Misbehavin'	101
A.2	All of Me	102

A.3	Autumn Leaves	102
A.4	Blue and Sentimental	103
A.5	Embraceable You	103
A.6	The Girl from Ipanema	104
A.7	Here's That Rainy Day	104
A.8	The Joint is Jumpin'	105
A.9	Pennies from Heaven	105
A.10	Solar	105
A.11	Hey Joe	106
A.12	Coker, App D., 10	106
B	Score for Bach Example	107
	Bibliography	109

Chapter 1

Introduction

1.1 Music as a Language

It is an old and popular idea that music may be treated as a language: an abstract and ill-defined, but nevertheless powerfully expressive language. Lindblom & Sundberg (1969) take this idea a step further by suggesting that music might be described by the same formal mechanisms that have been applied to so-called “natural languages” since the early work of Chomsky (Chomsky (1957)). Despite the various added complexities of describing music over other natural languages – the variety and subtlety of notation and the greater impact of precise issues of timing, for instance – formal grammars may be applied to the language of music in much the same way as they are to other languages. They provide us with a powerful tool to describe not only the processes that allow a composer to select sequences of notes and chords that sound coherent and conventional to any listener familiar with Western music, but also those that allow a listener, who may have no formal training in the process of composition, to declare a piece of music coherent or not. Longuet-Higgins & Lisle (1989) draw a more specific correlation between poetry and music.

Lindblom & Sundberg (1969) introduce the idea that music, a structured and to a large extent rule-driven form of expression, cannot be adequately described by Markovian models, but that instead formal grammars provide tools fit for the task. Further, they refer to models using these tools as potentially providing insight into the psychological processes behind music. Given the strong correspondence between music and other natural languages, it is not an unreasonable hypothesis that the psychological processes associated with them have much in

$\sharp V^-$	$\sharp II^-$	$\sharp VI^-$	$\sharp III$	$\sharp VII$	$\sharp\sharp IV$	$\sharp\sharp I$	$\sharp\sharp V^+$	$\sharp\sharp III^+$
III^-	VII^-	$\sharp IV^-$	$\sharp I$	$\sharp V$	$\sharp II$	$\sharp VI$	$\sharp III^+$	$\sharp VII^+$
I^-	V^-	II^-	VI	III	VII	$\sharp IV$	$\sharp I^+$	$\sharp V^+$
$\flat VI^-$	$\flat III^-$	$\flat VII^-$	IV	I	V	II	VI^+	III^+
$\flat IV^-$	$\flat I^-$	$\flat V^-$	$\flat II$	$\flat VI$	$\flat III$	$\flat VII$	IV^+	I^+
$\flat\flat II^-$	$\flat\flat VI^-$	$\flat\flat III^-$	$\flat\flat VII$	$\flat IV$	$\flat I$	$\flat V$	$\flat\flat III^+$	$\flat\flat VI^+$

FIGURE 1.1: Longuet-Higgins’ two-dimensional tonal space, with points denoted by roman numeral intervals relative to a key. The horizontal represents movements in perfect fifths, the vertical movements in major thirds. The naming system repeats itself every four horizontal steps. $+$ s denote tones slightly higher than their synonyms in the immediate key of I , $-$ s those slightly lower.

common, a hypothesis supported by neurological studies (e.g. Patel (1998), Patel (2003) and Koelsch et al. (2005)).

Longuet-Higgins & Lisle (1989) apply an approach to modelling the perception of music based on Chomskian grammars, in which a language corresponds to a musical idiom, an utterance to a composition and meaning to an affective interpretation of the music. Steedman (1984) focuses on harmony, abstracting away from the more intricate details of the realization of the harmony by dealing only with notated chord sequences. He builds a grammar for a specific idiom – the particular style of jazz music known as *the blues*. The grammar formalism used is that of context-free grammars (CFGs).

Longuet-Higgins (1979) presents the basis for a model of human perception of music. He describes a three-dimensional *tonal space*, based on the close relationship between tones separated by intervals that correspond to the lowest pitches in the harmonic series. The bases of the space represent the intervals of an octave, a perfect fifth and a major third (usually projected onto two dimensions, treating tones separated by an octave as equivalent, see figure 1.1). Tones close in this space are closely harmonically and perceptually related. Steedman (1996)

presents a new grammar, again to describe the blues, but using the formalism of Combinatory Categorical Grammar (CCG). This grammar now builds a semantic representation of the chord sequence in terms of movements of the chord roots in Longuet-Higgins' tonal space. It therefore serves as a mechanism to produce harmonic analyses of chord sequences.

1.2 Harmony

Study of the theory of musical harmony has developed over several centuries. A major step towards the modern approach to harmonic analysis was made by the work of Hugo Riemann around the end of the 19th century, whose most important contribution was to introduce the idea of the *tonal function* of chords. This is the idea that all chords (that is, the implicit or explicit harmony notes underlying a momentary fragment of music) have one of three functions in driving the harmony of the music. A *tonic* chord remains settled in the music's current, localized tonality and exerts no harmonic drive. A *dominant* chord is rooted a perfect fifth above the tonic and has the effect of driving towards the tonic. A *subdominant* is rooted a perfect fifth below the tonic and also drives towards the tonic. These functions can be expressed in terms of movements in Longuet-Higgins' tonal space. The tonic brings about no movement, the dominant a movement one step to the left, and the subdominant a step to the right.

Much music theory literature is devoted to describing the functions that chords may have, often dependent on their harmonic context, and chords that may behave as substitutes with equivalent function for others. In cases where this function is particularly strong, it may seem to raise an expectation of the harmonic root of the following chord, though this expectation is never unambiguous. A chord which fulfills some expectation raised by its preceding chord by completing its harmonic movement is referred to as its *resolution*.

A *cadence* is a device used as a conclusion at the end of a musical phrase. A variety of different sorts of cadence are given their own names. The most common cadence, due to its strong finality, is the *authentic* cadence, which involves the resolution of a dominant to the tonic. Less common is the *plagal* cadence, which resolves to the tonic from a subdominant function chord. Not all tonic-resolved dominants constitute cadences; in this work, however, as in Steedman (1996), a practical distinction is not made between the dominant resolution that has the

finality of a cadence, due to its time of occurrence and its harmonic context, and the more localized resolution with a weaker effect than a cadence.

A dominant function chord may resolve to another dominant function chord, which must then itself resolve. This first is referred to as a *second-level* dominant. Further levels of dominant are permissible and are very common in jazz harmony. A cadence may be brought about by unboundedly extended sequences of dominant chords. In the tonal space analysis, such a cadence will correspond to a chain of left movements. The same may be done with subdominant, plagal cadences, though this is less common.

The harmonic analysis performed by the grammar of Steedman (1996) is based entirely on cadences, interpreting indefinitely long sequences of left or right movements in the tonal space.

1.3 Jazz

The present study, like Steedman (1984) and Steedman (1996), focuses mainly on grammatical descriptions of chord sequences of jazz standards, as used by jazz musicians as the basis for improvisation.

One reason for this is that jazz is a particularly apt source of examples. There are several reasons why this is so: there is a plentiful supply of transcribed chord sequences to describe the harmony of its songs; it is common in jazz to make extensive use of extended cadences, a particular focus of the development of Steedman's grammar; and there are well-established conventions for producing variations on a given chord sequence without disturbing its functional form, exemplified in transcribed examples.

Jazz also provides us with an important tool in the form of the *12-bar blues*. This general musical form is the basis for many chord sequences, often with very little in common, but which are all recognized as being closely related by human listeners. If, as is hoped, a musical grammar can serve as a good model of human perception of harmony, it should be possible to identify the features that perceptually link all of these chord sequences in the features of the semantic representations it produces.

The necessity for a musical grammar to be a description of only a specific idiom is very much reduced by the use of an abstract chord notation as the material for analysis and by the generality of the theory of harmonic function once style-

specific issues of realization have been removed. However, jazz does make use of certain harmonic devices, such as long extended cadences, more commonly than others and makes less use of some devices common in other styles. It is therefore helpful to take the particular idiom as the basis of the study and consider later how well the grammar may generalize to others.

Pachet (2000) presents a system of computational analysis of jazz chord sequences and discusses the particular question of whether Miles Davis' standard *Solar* can be interpreted by the system (or any other) as a blues. He discusses the limitations of three preceding models of analysis, including Steedman (1984). He suggests three features of the analysis of Steedman (1984) which present major drawbacks. Two of these – the problem of implementing the model and the usefulness of its analysis, consisting of a yes/no answer and a derivation tree – were overcome by the CCG approach proposed by Steedman (1996). The remaining criticism is that such a form of grammar can only describe chord sequences deriving from some basic form like the blues and cannot provide a general harmonic analysis of chord sequences. The grammar of Steedman (1996) went some way towards overcoming this. In this study I consider the generality of the model presented, not only beyond the 12-bar blues form, but beyond the domain of jazz harmony.

1.4 Combinatory Categorical Grammar

Steedman (1996) replaced the CFG of Steedman (1984) with a new grammar using the formalism of Combinatory Categorical Grammar (CCG, Steedman (2000)). This formalism has various attractive properties, identified with regard to written language, which extend fairly directly to its use for harmonic analysis. Most notable is its ability to produce left-branching analyses of what are typically thought of as right-branching syntactic constructions. For harmonic analysis, this is most important to the process of constructing interpretations of cadences, which appear to be unboundedly extendable backwards from their final resolution. CCG allows each chord to be described in terms of its role in driving towards that resolution, a representation of the cadential structure being built up incrementally as each chord is encountered. This property in particular is important in presenting the musical grammar as a model of human interpretation of harmony.

A closely related property is the grammar's direct expression of a chord's

harmonic function. In CCG, A/B denotes a category that may combine with a category B to its right to produce a category of type A . In the grammar of Steedman (1996), categories of this form express the function of a chord in raising an expectation of their resolution to a particular following harmonic root. For example, the category I_X^7/IV_X^7 , assigned to a chord $X(7)$, expresses an interpretation of this minor seventh chord in which it expects a resolution to a chord rooted a perfect fifth below it. Furthermore, the semantics associated with this category – $\lambda x.leftonto(x)$ – specifies the movement in the tonal space that accompanies this interpretation – namely a single step to the left – independently of its harmonic context.

One of the key insights of Steedman (2000) was to treat syntax as a device to map surface forms into semantic structures. In the case of harmonic grammars using CCG, unlike the earlier CFG grammar, the syntactic categories do not in themselves constitute a harmonic analysis, but serve to produce a predicate-argument structure semantics describing a harmonic analysis in terms of the Longuet-Higgins tonal space. Most modern Western tonal music uses the tuning system of *equal temperament*, in which twelve tones are spaced at equal pitch intervals between octaves. The theories of tonality of Longuet-Higgins (1979) are based on the mapping between a natural system of tuning by *just intonation*, with intervals based on the low-frequency harmonics, and equal temperament. By reference to these theories, the harmonic analysis of the grammar can explain some elements of the psychological and emotional impact that a harmonic progression has. Some examples are given by Steedman (2004).

CCG grammars can be parsed using various chart-parsing algorithms. Steedman (2000) describes a modification of the CKY incremental chart-parsing algorithm suitable for parsing using a CCG grammar. It is this algorithm that is used here as the basis for the process of parsing musical grammars.

1.5 The Present Work

This work embarks upon a major redevelopment of the grammatical theory of harmony presented by Steedman (1996), with the aim of producing a partial model of human perception of tonal harmony, focussing particularly on the domain of jazz chord sequences. The steps in this process fall into four categories.

In chapter 2 I describe many individual modifications made to the original

CCG chord grammar. The original categories are revised and some new categories are added. Each change is motivated by reference to works on harmonic theory from Riemann (1895) onwards and is supported by examples of chord sequences from jazz standards taken from Elliott (to appear 2008) and Coker (1964). Changes are also made to the rules that may be used by the grammar and at the end a complete new grammar is presented.

In chapter 3 I propose a method for generalized interpretation of musical semantics that allows common musical abstractions to be expressed as constraints on the predicate semantics produced by the grammar. This model, expressed in *underspecified semantic expressions*, is applied to the 12-bar blues, using it to capture the most important perceptual attributes that define the form.

In chapter 4 I describe the implementation of a parser to apply the grammar and the process of matching underspecified semantic expressions to actual chord sequences. I outline various tools provided by the parser to aid the testing of the grammar as a model for musical interpretation.

In chapter 5 I apply the parser to example chords sequences. I demonstrate the use of the underspecified semantic expressions for recognizing examples of the 12-bar blues and another similar form (Gershwin's *I Got Rhythm*). I then demonstrate the application of the parser to a broader range of jazz standards. I also discuss the generality of the model that the grammar provides, with some specific examples from outside the jazz idiom. The direct correspondence between the grammar and the theory of harmonic function and the use of tonal space movements as a model for analysis allow the chord grammar to be viewed not just as a model of blues chord sequences, but a model for harmonic perception and analysis universal within Western tonal music. By reference to examples of chord sequences of jazz standards, I highlight the current limitations of the grammar and suggest ideas for future development that could make it more broadly applicable.

Chapter 2

Grammar Development

2.1 Introduction

Steedman (1996) devised a CCG grammar for jazz chord sequences which set out to produce semantic interpretations of 12-bar blues chord sequences in terms of movements in the Longuet-Higgins tonal space. I have developed a grammar to provide interpretations of jazz chord sequences taking Steedman's grammar as a starting point. Each step in developing the new grammar is supported by examples from chord sequences found in Coker (1964) and Elliott (to appear 2008), applying musical intuition and making reference to texts on musical theory and analysis as appropriate. The aim of this was to build a grammar which can produce musically meaningful analyses of chord sequences in terms of tonal space movements which correspond to analyses that would be given by performers or musically aware listeners. Indeed, a good analysis should be motivated by the intuitions of any listener familiar with Western tonal music regarding the expectations, resolutions and continuity of chords in a chord sequence.

Below I set out Steedman's most recently published grammar and explain the purpose of some of its components. Then, in the subsequent sections, I describe each of my modifications to the grammar and their justification. Finally, I summarize the whole resulting grammar.

2.2 The Original Grammar

Steedman's CCG chord grammar contained the following lexical categories.

- 0a. $X(m) := I_X(m) \setminus I_X(m) : \lambda x.x$
 0b. $X(m)^7 := I_X(m)^7 \setminus I_X(m)^{(7)} : \lambda x.x$
 1a. $X := I_X : X$
 1b. $Xm := I_X m : X$
 2a. $X := V_X \setminus V_X : \lambda x.x$
 2b. $Xm := V_X m \setminus V_X m : \lambda x.x$
 3a. $Xm^7 := I_X m^7 / IV_X^7 : \lambda x.leftonto(x)$
 3b. $X^7 := I_X^7 / IV_X(m)^7 : \lambda x.leftonto(x)$
 4. $Xm^7 := \sharp IV_X(m)^7 / VII_X(m)^7 : \lambda x.leftonto(x)$
 5a. $X := I_X / V_X : \lambda x.rightonto(x)$
 5b. $Xm := I_X m / V_X m : \lambda x.rightonto(x)$
 6. $Xm := \flat VII_X m \setminus \flat VII_X m : \lambda x.x$
 7. $X\circ 7 := \flat V_X / \flat V_X : \lambda x.x \mid \flat II_X / \flat II_X : \lambda x.x \mid \flat VII_X m^7 / \flat VII_X m^7 : \lambda x.x$

The standard CCG rules of *function application* and *composition*, both crossing and harmonic, may be used in derivations, with the addition of a rule akin to the type-raising rule of natural language grammars which serves to provide a full interpretation of authentic cadences:

$$X : origin \Rightarrow (I_X \setminus I_X) \setminus (Y^7 / I_X^7) : \lambda cadence.\lambda origin.origin + cadence(origin)$$

A similar rule must also be used to construct plagal cadences:

$$X : origin \Rightarrow (I_X \setminus I_X) \setminus (Y / I_X) : \lambda cadence.\lambda origin.origin + cadence(origin)$$

This rule and its derivative (see section 2.7) are henceforth referred to as *cadence-raising*, in reference to their similarity to type-raising.

2.2.1 Chords and Chord Classes

On the left-hand side of the categories above are basic chord types to which the categories may be assigned. Only in the case of category 7, which interprets diminished seventh chords, does this refer specifically to a single chord type. In all other cases, the symbol represents a class of chord types. X and Xm denote chords in the classes of major and minor tonic chords respectively; X^7 and Xm^7 the classes of major and minor dominant chords. Section 4.4.1 gives a full explanation of the notational conventions used here for chords. Throughout this paper, actual chords expressed using these conventions are written in the

same form as parser input – e.g. $\flat III m(7)$ – whilst grammatical categories are written $\flat III m^7$ and names of points in the tonal space are similarly formatted (e.g. $\flat III$).

The class of major tonic chords includes the simple major chord, but also major chords with additions for colouration. The class must include the chord $X(7)$, since the chord of the added minor seventh may be a a tonic chord with a minor seventh colouration or a dominant seventh chord, with the seventh signaling the chord's dominant function. These two additions have distinct pitches in just intonation and correspond to distinct points in Longuet-Higgins' tonal space, but are mapped to the same pitch by equal temperament. The dominant seventh is the more common use of the addition, but the grammar must permit this ambiguity. The minor seventh with a tonic function is particularly common in the blues. The class of minor tonic chords is similar to the major tonic class.

The classes of major and minor dominant chords also include the minor seventh chords, as well those with other additions or modifications that may add further strength to the dominant function of the chord, such as $X(7+5)$. For a fuller introduction to the basic chord classes see Coker (1964).

The classes used here are based on those given in Steedman (2004), with the addition of several chord types, including simple triads without any additions, which were not previously explicitly included, and several of the more common additions. A further addition is called for by an examination of notated chord sequences, such as those in Elliott (to appear 2008). There are many cases where a chord with a dominant function clear from its context – in the middle of a cadence, for example – does not have this function signaled by a seventh or any other addition. The dominant drive of a major chord is made especially powerful by the addition of the minor seventh, but any major chord may have a dominant function (Pratt (1984)). A minor chord will not have the function of a first-level dominant, since even in minor keys the major triad is typically used for the dominant chord, but as further levels of dominants are added a minor chord with dominant function is equally likely. In order to give correct interpretations to these chords the grammar must incorporate still further ambiguity by including simple X and Xm chords in the major and minor dominant classes.

The resulting classes are as follows:

$\{X, X(M7), X(7), X(9), X(13), X(6)\}$	$:= X$
$\{X_m, X_m(7), X_m(6)\}$	$:= X_m$
$\{X(7), X(b9), X(b10), X(7+5), X_{aug},$ $X(7+11), X(7+9), X_{aug}(7), X, X(M7)\}$	$:= X^7$
$\{X_m(7), X_m(9), X\%7, X_m\}$	$:= X_m^7$

2.2.2 The Original Categories Explained

The 0 categories are intended to allow sequential similar chords to combine into a single category, giving a single interpretation to the whole sequence. A sequence of chords X should be able to receive any category that could be given to the single chord X and a sequence $X(7)$ $X(7)$ or X $X(7)$ should be able to receive the same category as $X(7)$. Note that any sequence in which at least the final chord is an added seventh chord can be interpreted as if a single added seventh chord – that is, can receive a dominant seventh interpretation as well as a tonic interpretation. This is because the function of a dominant seventh chord is to raise the expectation of a subsequent resolution (a $V - I$ resolution), so it is the final chord that must perform this function. Henceforth, any sequence of chords which can be combined to perform a single function in these ways is referred to as a sequence of similar chords.

The 1 categories are the basic interpretations of a chord as the atomic category associated with that chord. The semantics is just a single point in the tonal space. In practice, this may function as an initial tonic, like the first I in $I - II(7) - V(7) - I$, or as the target (resolution) of a cadence, like the second.

The 2 categories are motivated by the movement to the IV chord in the 12-bar blues and allow these sequences to be accepted by the grammar. It is also common to see $I - IV - I$ sequences, which are merely a colouration of what is really just a stationary I section. Coker (1964) calls this *tonic relief*, describing it initially as a short-term modulation to IV , but conceding that it is so temporary that the term “modulation” is inaccurate. Accordingly, the category has an empty semantics – there is no movement in the harmonic space associated with such a fragment. As is elaborated in section 2.8.3, the movement to the IV in the second section of a 12-bar blues is not semantically empty. I later distinguish this case from the tonic relief case and show that it should not be interpreted using this category, but as a genuine modulation.

The 3 categories are used to interpret steps in an extended authentic cadence,

made by dominant seventh chords. As Steedman (1996) describes, such a cadence may be arbitrarily long and is represented in the semantics by *leftonto*(x) predicates, denoting left steps in the tonal space. A cadence of arbitrary length can be interpreted by composition of lexical 3 categories to produce a category X^7/Y^7 . This can then be used by the subsequent cadence-raised tonic, which serves as the target of the cadence.

Category 4 handles the tritone substitution – the substitution for a dominant seventh chord of the dominant seventh rooted on its tritone (its augmented fourth, three tones above it). This phenomenon is often thought of as specific to jazz, though it occurs frequently in much earlier music¹ (see section 5.5.2, for example). This substituted chord behaves in the same way as would its tritone. It is therefore treated as if it were this dominant seventh chord, both in its category and semantics.

The 5 categories serve the same purpose as the 3 categories, only this time for plagal cadences. Each step in a plagal cadence behaves as a subdominant to the chord that follows it. As Steedman (1996) notes, there is no obvious signal of this chord function, such as the seven in the case of dominant seventh chords. Any major or minor chord may be given this interpretation – the correct cases may be recognized only by their resolution. Note that this plagal resolution is carried out either by composition, in the case of extended plagal cadences, or by backward application of a plagal cadence-raised category. It is incorrect to resolve such a subdominant chord by direct application of its sign to that of the following chord. This is discussed further in section 2.6.

Category 6 is motivated by a progression found in certain examples of blues sequences in which a *IIm* leads to a *IIIIm*. It handles few cases other than this example and it is not clear what musical justification could support it.

Category 7 handles diminished seventh chords. These have a very ambiguous resolution and the categories given here encapsulate a few of the possibilities. All three interpretations of the chord consider its colouration and result in no movement in the tonal space.

¹The chord is traditionally analyzed as an inverted augmented sixth chord with added flattened ninth tone – the so-called *German sixth* (see Pratt (1984)). This is enharmonically equivalent to a dominant seventh chord with the root a tritone higher. This is an alternative analysis of precisely the same phenomenon as the tritone substitution.

2.2.3 New Optional Minor Notation

The original grammar uses a notation for optional minors in which a chord category may be written $X(m)$ to indicate that it may refer to a major or a minor chord. If an optional minor occurs on the left-hand side of a category's definition, the category may be used for a major or minor chord (or chord class). If one occurs on the right-hand side, the category may be equated to a major or a minor category during rule application. If optional minors occur on both sides, the "minorness" of the chord on the left and the category on the right are bound together – the category is minor if and only if the chord on the left is minor.

It should be noted that a category using the optional minor notation may always be expressed as multiple categories without using the optional minors. For example,

$$3b. \quad X^7 := I_X^7/IV_X(m)^7 : \lambda x.leftonto(x)$$

may be less concisely written

$$3b-m. \quad X^7 := I_X^7/IV_X m^7 : \lambda x.leftonto(x)$$

$$3b-M. \quad X^7 := I_X^7/IV_X^7 : \lambda x.leftonto(x)$$

Later modifications of the grammar use a new notation which extends this notation to allow still more concise expression of related categories. Each optional minor has a subscript index associated with it; its minorness is bound explicitly to any other optional minors within the complex category or on the left-hand side that share the index. It is therefore possible, for example, to have an optional minor category within the right-hand side whose minorness is independent of an optional minor on the left-hand side, or multiple independent classes of optional minors within a category.

By convention, optional minors may be written without a subscript index, in which case all unindexed optional minors are implicitly bound. To simplify implementation, it is assumed that any unindexed optional minor is in the optional minor class 0. All categories written using the old notation will therefore be interpreted identically if the new notation is assumed, since all optional minors will be assumed to be in class 0 and will be bound to each other.

There are no examples in the initial grammar of circumstances in which this notation is useful, but examples will arise as a result of later grammar modifications.

2.2.4 The Original Grammar Rewritten

The original grammar can be slightly more concisely expressed using (unindexed) optional minors.

- 0a. $X(m) := I_X(m) \setminus I_X(m) : \lambda x.x$
- 0b. $X(m)^7 := I_X(m)^7 \setminus I_X(m)^{(7)} : \lambda x.x$
- 1. $X(m) := I_X(m) : X$
- 2. $X(m) := V_X(m) \setminus V_X(m) : \lambda x.x$
- 3a. $Xm^7 := I_X m^7 / IV_X^7 : \lambda x.leftonto(x)$
- 3b. $X^7 := I_X^7 / IV_X(m)^7 : \lambda x.leftonto(x)$
- 4. $Xm^7 := \sharp IV_X(m)^7 / VII_X(m)^7 : \lambda x.leftonto(x)$
- 5. $X(m) := I_X(m) / V_X(m) : \lambda x.rightonto(x)$
- 6. $Xm := \flat VII_X m \setminus \flat VII_X m : \lambda x.x$
- 7. $X\circ 7 := \flat V_X / \flat V_X : \lambda x.x \mid \flat II_X / \flat II_X : \lambda x.x \mid \flat VII_X m^7 / \flat VII_X m^7 : \lambda x.x$

2.3 Minor Resolution in Authentic Cadences

A simple addition that I made to the 3 categories – those used for building authentic cadences – was to allow a minor dominant seventh chord to resolve by a left step in the tonal space to another minor chord. In category 3a of the original grammar a minor dominant seventh chord is interpreted as expecting a resolution to the major chord a fifth below it:

- 3a. $Xm^7 := I_X m^7 / IV_X^7 : \lambda x.leftonto(x)$

There is no reason why such a chord should not equally well resolve to a minor chord, in the same way that 3b allows a major dominant seventh to do. There are examples of such a resolution in standards to support this modification, such as Coker (1964), app. D, 11, which includes the following cadence at the end:

$$\flat VII\text{Im}(7), \flat III\text{Im}(7) \quad V\text{Im}(7), II(7) \quad \flat VI\text{Im}(7), \flat II(7) \quad I$$

This extended cadence moves largely in left steps in the tonal space. The first left step – $\flat VII\text{Im}(7) \flat III\text{Im}(7)$ – is made by two minor chords, providing an example of the requirement of a minor resolution for the minor dominant seventh interpretation, category 3a.

The new rule 3a therefore incorporates an optional minor resolution:

- 3a'. $Xm^7 := I_X m^7 / IV_X(m)^7 : \lambda x.leftonto(x)$

The new pair of dominant seventh categories, 3a' and 3b, can be compressed into a single category using the new notation for multiple bound optional minor

classes. Thus,

$$3a'. \quad X m^7 := I_X m^7 / IV_X(m)^7 : \lambda x. leftonto(x)$$

$$3b. \quad X^7 := I_X^7 / IV_X(m)^7 : \lambda x. leftonto(x)$$

becomes

$$3. \quad X(m)_0^7 := I_X(m)_0^7 / IV_X(m)_1^7 : \lambda x. leftonto(x)$$

or just

$$3. \quad X(m)^7 := I_X(m)^7 / IV_X(m)_1^7 : \lambda x. leftonto(x)$$

2.4 Tritone Substitutions

2.4.1 Major Tritone Substitutions

Tritone substitutions were previously interpreted using category 4:

$$4. \quad X m^7 := \sharp IV_X(m)^7 / VII_X(m)^7 : \lambda x. leftonto(x)$$

This allows a semitone step down between diminished seventh chords to be interpreted as a left movement in the tonal space, substituting the initial chord, which would be a fifth above the second chord, with its tritone, which is a semitone above the second chord.

There is no clear reason why this should be allowed to be done only using minor chords and there is plenty of evidence to support the use of major chords in the same way. There is also no reason why the $\sharp IV_X$ should be optionally minor, since it is always a substitute for a minor chord and the result of the category should signify this. We therefore allow major chords to be used in the same way and bind the minorness of the $\sharp IV_X$ category to that of the chord. This gives the following replacement for category 4:

$$4'. \quad X(m)^7 := \sharp IV_X(m)^7 / VII_X(m)_1^7 : \lambda x. leftonto(x)$$

2.4.2 New Tritone Substitution Categories

The original categories correctly accepted tritone substitutions such as those found in the jazz standard *Autumn Leaves*, but left no indication in the semantics produced that a tritone leap had been made in the tonal space. This category seems to describe well the traditional analysis of these chords as inverted augmented sixth chords, rather than the jazz approach, in which the chords are treated as dominant sevenths substituted for their tritones. It also meant that in a cadence with a tritone leap in the middle (or one implied by a downward

semitone step), all chords preceding the tritone had to be interpreted as tritone substitutions.

Autumn Leaves provides a good example of this very common application of the tritone substitution.

$$\begin{array}{c} \text{IVm}(7) \text{ bVII}(7) \text{ bIII} \text{ bVI} \text{ II}\%7 \text{ V}(7) \\ \text{Im}(7), \text{VIIo}7 \text{ bVIIIm}(7) \text{ bIII}(7) \text{ bVI}(7+9) \text{ V}(7+9) \text{ Im} \end{array}$$

This whole sequence can be interpreted as a single, long authentic cadence. Largely it moves in left steps – down in fifths. Between the **bVI** and the **II%7** it makes a leap of a tritone and there are three examples of a downward semitone step, interpreted as a tritone substitution for a left step. The original grammar does not cover the explicit tritone jump between dominant sevenths, but it can be simply extended to do so. Using the original grammar with such an extension, we compose simple dominant seventh categories (3) for the first cadential passage up to the tritone leap and compose this with tritone-substituted dominant seventh categories (4) for the chords following simple left-step sequence. At the **VIIo7** we switch back to a 3 category, then switch straight back again to 4 categories at the **bVIIIm(7)**, until the **bVI(7+9)**, and finish using a 3 category. All of these categories can be composed and the result used as a cadence for the cadence-raised final tonic.

However, this description of the cadence is unsatisfactory. It should not be necessary to describe cadential steps on either side of a tritone jump using different categories, even though this corresponds to the traditional analysis of the substitution. A cadential sequence may include a tritone jump without disturbing the sequence of left steps and it should be possible to use essentially the same category for each ordinary left step and to apply a special interpretation only at the points where a tritone or semitone jump is made. Within a sequence of left steps by dominant sevenths, the chords are perceived as dominant sevenths regardless of whether they are in fact dominant sevenths or inverted augmented sixths. It is therefore preferable if the same category can be used with such a sequence.

Furthermore, the current categories leave no indication in the produced semantics that a tritone substitution was made. Again this is not a problem if the analysis treats tritone-substituted segments of the cadence as inverted augmented sixths, but if each segment is to be treated as a partial cadence by dominant sevenths, some representation of the transition between the segments needs to be

encoded in the semantics.

A better approach is to interpret the tritone leap at the point at which it occurs, allowing a cadence before it and after it to be interpreted simply as cadential dominant sevenths using category 3. A transition between a tritone-substituted chord and a dominant seventh a semitone below is now seen as a left step combined with a tritone leap. An explicit tritone leap between equivalent dominant sevenths, as seen in *Autumn Leaves* above, must also be allowed. We now introduce a new unary semantic predicate *tritone*, denoting a tritone movement in the tonal space. As is discussed in more detail in section 3.5, unlike the other predicates, this is an ambiguous movement, since there are two points in the space relative to any starting point which both represent a tritone jump and are both equidistant from the starting point. This ambiguity is left for resolution at a later stage.

The following new categories interpret tritone leaps and signify them in the semantics with a *tritone* predicate. 4a allows an explicit tritone jump to appear in a cadence. 4b interprets a semitone step down between dominant sevenths as a *leftonto* step composed with a tritone jump.

$$4a. \quad X(m)^7 := I_X(m)^7 / \flat V_X(m)_1^7 : \lambda x. \text{tritone}(x)$$

$$4b. \quad X(m)^7 := I_X(m)^7 / VII_X(m)_1^7 : \lambda x. \text{tritone}(\text{leftonto}(x))$$

Note that a single tritone substitution for a dominant seventh, as frequently occurs at the end of an authentic cadence, will now be viewed semantically as a tritone jump plus a left step onto the chord, followed by a tritone jump (back again) plus another left step onto the following chord. So, for example, the cadence I II(7) \flat II(7) I receives the following categories:

$$I := I : I$$

$$II(7) := II^7 / \flat II^7 : \lambda x. \text{tritone}(\text{leftonto}(x))$$

$$\flat II(7) := \flat II^7 / I^7 : \lambda x. \text{tritone}(\text{leftonto}(x))$$

$$I := I : I$$

This interpretation gives the cadence the final sign:

$$I : I + \text{tritone}(\text{leftonto}(\text{tritone}(\text{leftonto}(I))))$$

2.5 Similar-Chord Sequences

2.5.1 New 0-Categories

The handling of sequences of similar chords using the 0 categories of the original grammar was flawed. The original categories $0a$ and $0b$ are assigned to non-initial chords in a similar sequence and combine backwards with their preceding chord. This means that it is always the first chord in the sequence that must be assigned the category that will be used for the whole sequence.

For example, in the sequence $I_0 I_1 I_2$, chords 1 and 2 are given 0 categories and chord 0 is given the category corresponding to the interpretation of the whole sequence, say category 1.

As explained in section 2.2.2, it is the last chord in the sequence that must have an added minor seventh if the sequence is to be interpreted as a dominant seventh overall. Category $0b$ allows dominant seventh chords to combine backwards with preceding similar chords, which may or may not be sevenths themselves. However, in the case where they are not (say, $I I I(7)$), a dominant seventh interpretation will never be possible, since the overall category must come from the first chord, which is not a seventh chord².

Another undesirable result is that derivations such as that in 1 can lead to nonsensical atomic X^7 categories. X^7 s should only ever be part of a complex category representing a cadence, since, in our broad definition of a cadence, this is the only purpose of the dominant seventh interpretation.

$$\begin{array}{c}
 (1) \quad \frac{\frac{\frac{I \quad V(7)}{I : I \quad V^7/I^7 : \lambda x. \text{leftonto}(x)} \quad \frac{I}{I : I}}{\frac{(I \setminus I) \setminus (Y^7/I^7) : \lambda C. \lambda P. P + C(I)}{I \setminus I : \lambda x. x + \text{leftonto}(I)}}{\frac{I : I + \text{leftonto}(I)}{I^7 : I + \text{leftonto}(I)}}
 \end{array}$$

The solution to this is to rewrite the 0 categories so that they combine *forwards* instead of backwards. The actual category for the sequence will come from the final chord, all others receiving a 0 category. The sequence $I I I(7)$ may now

²Allowing major chords to behave as dominants in fact allows the dominant seventh category to be assigned to this sequence. However, the principal that it is the final chord of the sequence that must determine its function should still motivate these categories.

receive a dominant seventh category, since the final chord has the added seventh and the other chords are effectively ignored.

The new categories are:

$$0a. \quad X(m) := I_X(m)/I_X(m) : \lambda x.x$$

$$0b. \quad X(m)^{(7)} := I_X(m)^7/I_X(m)^7 : \lambda x.x$$

The sequence $I \ I \ I(7)$ can now receive a dominant seventh category, as shown in 2.

$$(2) \quad \begin{array}{ccc} I & I & I(7) \\ \hline I^7/I^7 : \lambda x.x & I^7/I^7 : \lambda x.x & I^7/IV^7 : \lambda x.leftonto(x) \\ \hline & I^7/I^7 : \lambda x.x & \xrightarrow{\mathbf{B}} \\ \hline & I^7/IV^7 : \lambda x.leftonto(x) & \xrightarrow{\mathbf{B}} \end{array}$$

2.5.2 Major-Minor Transitions

It is common to see cases in which a major chord follows a minor chord on the same root or vice versa. This seems to be applied freely and has no effect on the semantics produced by the grammar. The jazz standard *Pennies from Heaven*, for example, contains some examples of this. This example leads up to the end of the first half:

$$\begin{array}{c} bIII\flat_7 \ II\flat(m) \ V(7) \ | \ V\flat(m) \ I(7) \ IV \ | \\ V\flat(m) \ II(7) \ II\flat(m) \ V(7) \ | \ I \end{array}$$

It is easy to allow such transitions as these in the grammar simply by modifying the 0 categories: we detach the minorness of the category on the right of the slash from that of the category on the left and the chord itself. We can do this neatly using multiple indexed optional minor classes:

$$0a. \quad X(m) := I_X(m)/I_X(m)_1 : \lambda x.x$$

$$0b. \quad X(m)^{(7)} := I_X(m)^7/I_X(m)_1^7 : \lambda x.x$$

Chords $X \ X\flat$ and $X\flat \ X$ can now be combined and treated as if a single chord.

2.6 Cadential Slash Modes

As previously discussed, authentic cadences are produced by a dominant seventh interpretation of a minor seventh chord and a cadence-raised target chord.

Extended cadences can be interpreted by composing the dominant seventh categories³:

$$(3) \quad \frac{\frac{\frac{\frac{\mathbf{I}}{\mathbf{I} : \mathbf{I}} \quad \frac{\mathbf{II}(7)}{\mathbb{II}^7/\mathbb{V}^7 : \lambda x.L(x)} \quad \frac{\mathbf{V}(7)}{\mathbb{V}^7/\mathbb{I}^7 : \lambda x.L(x)}}{\mathbb{II}^7/\mathbb{I}^7 : \lambda x.L(L(x))} \xrightarrow{\mathbf{B}} \frac{\frac{\mathbf{I}}{\mathbf{I} : \mathbf{I}}}{(\mathbf{I} \setminus \mathbf{I}) \setminus (\mathbf{Y}^7/\mathbb{I}^7) : \lambda C.\lambda P.P + C(\mathbf{I})} \xrightarrow{\mathbf{T}}}{\mathbf{I} \setminus \mathbf{I} : \lambda x.x + L(L(\mathbf{I}))} \xrightarrow{\mathbf{I}}}{\mathbf{I} : \mathbf{I} + L(L(\mathbf{I}))} \xrightarrow{\mathbf{I}}$$

The dominant seventh category I_X^7/IV_X^7 correctly indicates that a chord X^7 raises the expectation of a chord IV_X to follow it. This combines nicely with the cadence-raising rule to produce interpretations of cadences because symbols X^7 only enter the derivation through dominant seventh categories like this (including also the tritone substitution categories). In particular, there is no category $X^7 := I_X^7$ that would give rise to something like the 4, meaning that the X^7 symbols only combine with other categories by composition, or with the cadence endings by application.

$$(4) \quad \frac{\frac{\mathbf{II}(7)}{\mathbb{II}^7/\mathbb{V}^7} \quad \frac{\mathbf{V}(7)}{\mathbb{V}^7}}{\mathbb{II}^7} \xrightarrow{\mathbf{I}}$$

However, with right steps – plagal cadences – there is no distinct category symbol, like the X^7 symbols, for its chord categories. As a result, as well as good interpretations like 5, we get nonsensical ones like 6.

$$(5) \quad \frac{\frac{\frac{\frac{\mathbf{I}}{\mathbf{I} : \mathbf{I}} \quad \frac{\mathbf{IV}}{\mathbb{IV}/\mathbf{I}} \quad \frac{\mathbf{I}}{\mathbf{I}}}{(\mathbf{I} \setminus \mathbf{I}) \setminus (\mathbf{Y}/\mathbf{I})} \xrightarrow{\mathbf{T}}}{\mathbf{I} \setminus \mathbf{I}} \xrightarrow{\mathbf{I}}}{\mathbf{I} : \mathbf{I} + \text{rightonto}(\mathbf{I})} \xrightarrow{\mathbf{I}}$$

$$(6) \quad \frac{\frac{\frac{\frac{\mathbf{I}}{\mathbf{I} : \mathbf{I}} \quad \frac{\mathbf{IV}}{\mathbb{IV}/\mathbf{I}} \quad \frac{\mathbf{I}}{\mathbf{I} \setminus \mathbf{I}}}{\mathbf{IV} \setminus \mathbf{I}} \xrightarrow{\mathbf{B}_\times}}{\mathbf{I} : \text{rightonto}(\mathbf{I})} \xrightarrow{\mathbf{I}}$$

The key problem is that category 5, which gives us the plagal steps, is not intended to be combined by function application with its resolution, but is only meant to be used to build a plagal sequence, which should be eventually combined with its cadence-raised target. There is no valid interpretation that involves

³Where necessary to save space, the *leftonto*(x) predicates are abbreviated to $L(x)$ and *rightonto*(x) to $R(x)$.

applying the plagal category to its successor. However, for every valid interpretation using plagal cadence-raising there is also a nonsensical one, resulting in a considerable number of systematically bad interpretations.

What is needed is a way to constrain the ways in which these categories may combine with others. These categories may be involved in composition with other cadential categories, but may not act as functors in function application or compose with non-cadential categories. This is almost the opposite of the * modality used in natural language CCG grammars to constrain the way complex may combine (Baldrige & Kruijff (2003)). The existing slash modes for natural language grammars cannot be applied directly, since we require a different set of limitations on rule applicability here.

I therefore now introduce a single modality symbol “c”, akin to the slash modes used in natural language grammars, denoting that a complex category is cadential and so may combine with other categories only by composition or as the argument in function application. Other categories have a non-cadential modality. I will mark this mode where necessary as “ ϕ ”, but this is the implicit mode where none is given.

This gives us the following new category 5. Similar changes are made to other categories that introduce cadential complex categories.

$$5. \quad X(m) := I_X(m)/_cV_X(m)$$

The combinatory rules must be modified to be conditional on the modality of their inputs⁴.

$$(>) \quad X/_\phi Y \quad Y \Rightarrow X$$

$$(<) \quad X \quad Y \backslash_\phi X \Rightarrow Y$$

$$(>\mathbf{B}) \quad X/_m Y \quad Y/_m Z \Rightarrow X/_m Z \quad \text{where} \quad \begin{cases} m_3 = c & \text{if } m_1 = c \text{ or } m_2 = c, \\ m_3 = \phi & \text{otherwise} \end{cases}$$

$$(<\mathbf{B}) \quad X \backslash_{m_1} Y \quad Z \backslash_{m_2} X \Rightarrow Z \backslash_{m_3} Y \quad \text{where} \quad \begin{cases} m_3 = c & \text{if } m_1 = c \text{ or } m_2 = c, \\ m_3 = \phi & \text{otherwise} \end{cases}$$

$$(>\mathbf{B}_\times) \quad X/_m Y \quad Y \backslash_\phi Z \Rightarrow X \backslash_m Z$$

$$(<\mathbf{B}_\times) \quad X/_m Y \quad Z \backslash_\phi X \Rightarrow Z/_m Y$$

$$(\mathbf{T}_a) \quad X \Rightarrow (I_X \backslash I_X) \backslash (Y^7/_c I^7)$$

$$(\mathbf{T}_p) \quad X \Rightarrow (I_X \backslash I_X) \backslash (Y/_c I)$$

⁴It is not really necessary to use the cadential mode for authentic cadences, since they are already distinguished by the $\bar{\tau}$ s, but I am treating them in this way to be consistent with plagal cadences.

The result is to cut down the number of results produced by parses of cadences quite dramatically, since all of the poor uses of the plagal cadence categories are eliminated.

2.7 Cadence-Raising Rules

2.7.1 New Semantics

Steedman's type-raising-style rule to allow a chord to behave as the target of an authentic cadence, referred to here as *cadence-raising*, was as follows:

$$X : orgn \Rightarrow (I_X \setminus I_X) \setminus (Y^7 / I_X^7) : \lambda cad. \lambda orgn. orgn + cad(orgn)$$

This includes an implicit condition on the second combination by function application of the resulting category with another: namely, that when the $(I_X \setminus I_X)$ category is the functor the argument category's semantics must be *orgn*. That is to say that the semantics of the origin of the cadence must be equivalent to that of the target.

This condition is overly restrictive and also involves an undesirable mixing of the processes of combining categories and combining their semantics. Whilst simple derivations are possible using this rule and the implicit semantic condition can be built into the implementation, no sequence can ever be recognized as containing consecutive cadences. The origin of the second cadence from the end would be the category representing all the previous cadences and would have the semantics built from these cadences. However, it would be required to have the semantics of the target of the final cadence (usually just *I*). The example Steedman (2004) gives is just such a sequence of cadences and the semantic condition is in fact not met in the derivation.

The condition on the cadence origin is fundamentally flawed. There is no reason why the semantics of the origin and target should be the same, only that their categories should be. We can replace the cadence-raising rule with the following, which still requires that the origin has the same category as the target, but places no equality condition on their semantics.

$$X : trgt \Rightarrow (I_X \setminus I_X) \setminus (Y^7 / {}_c I_X^7) : \lambda cad. \lambda orgn. orgn + cad(trgt)$$

By using this, the semantics of the target is not lost during cadence-raising, but instead is included in the semantics of the cadence-raised category.

The same must be done to the plagal cadence-raising rule:

$$X : \text{trgt} \Rightarrow (I_X \setminus I_X) \setminus (Y /_c I_X) : \lambda \text{cad} . \lambda \text{orgn} . \text{orgn} + \text{cad}(\text{trgt})$$

2.7.2 Minor Cadences

The form of the cadence-raising rule originally proposed does not allow for minor cadences. In such a cadence, the mid-cadential category itself is built in the same way, since this is able to mix major and minor chords freely already, but the final target of the cadence is minor. The argument taken for the origin of the cadence is also minor and so, of course, is the final category produced. A cadence-raising rule that builds major and minor cadences may be expressed using the same optional minor binding conventions as have been applied in lexical categories:

$$X(m) : \text{trgt} \Rightarrow (I_X(m) \setminus I_X(m)) \setminus (Y^7 /_c I_X(m)^7) : \lambda \text{cad} . \lambda \text{orgn} . \text{orgn} + \text{cad}(\text{trgt})$$

$$X(m) : \text{trgt} \Rightarrow (I_X(m) \setminus I_X(m)) \setminus (Y /_c I_X(m)) : \lambda \text{cad} . \lambda \text{orgn} . \text{orgn} + \text{cad}(\text{trgt})$$

2.8 Modulation

2.8.1 Introducing Modulations

It is common in many musical styles, including the forms of jazz which this study primarily concerns, for pieces to *modulate* temporarily from the key in which they begin to another key. Usually the piece will have modulated back to its original key by the end, though not always.

The jazz chord grammar currently is mainly concerned with cadences. It builds semantic structures that represent cadential movements; these structures are related by the concatenation, or +, operator. Where there are several consecutive cadences, they will each be represented by a predicate structure and linked by +s. The original grammar presents no means by which a chord sequence can be recognized as modulating to a new key. A cadence may start at any point in the tonal space, but must end at the same point, or one that is enharmonically equivalent in equal temperament.

The interpretation of modulations is highly ambiguous. In many circumstances, one analysis may consider a harmonic progression to have arrived in a new tonality (or *key*) at points when an alternative reading suggests that it is

still in the original tonality. However, the decision is an important one and lies at the heart of the attempt to produce a harmonic analysis of a chord sequence. As Pachet (2000) describes, harmonic structures may often be equally validly analyzed at various levels of granularity. A cadence, for example, may be considered in a single tonality, or may be broken down to a more fine-grained level in which it progresses through other tonalities. Indeed, Pratt (1984) mentions that some theorists would consider every resolved dominant-function chord in an extended cadence to be a momentary modulation. The importance of this issue for jazz is that the analysis that is used affects the way a performer improvises, since it is the analysis of the tonality of a particular sequence that determines the set of notes (scales) that are suitable to construct the new improvised melody.

The sort of extended cadences that we have considered in terms of the grammar are a source of great ambiguity in this respect. A short and very simple example, $I(M7) II(7) V(7) V(7) I(M7)$, may be considered a single cadence, starting at I in the tonal space and returning to the same point. Under this interpretation, it remains throughout in the tonality of I . However, an alternative view is that, beginning at I , the sequence modulates via a single-step cadence to V , treating the first $V(7)$ as a tonic chord, and then back to I at the end⁵. Under this reading, the sequence is recognized as beginning in the key of I , having a section (dominant plus resolution) in the key of V and having a final section (again, dominant plus resolution) in the key of I .

2.8.2 Recognizing Modulations with the Grammar

The grammar can be simply extended to produce interpretations of modulations of this kind. The cadence-raising rule is modified to remove the condition that the origin of a cadence must be categorically equal to the target. Instead of the previous

$$X(m) : trgt \Rightarrow (I_X(m) \setminus I_X(m)) \setminus (Y^7 / {}_c I_X(m)^7) : \lambda cad. \lambda orgn. orgn + cad(trgt)$$

⁵The repetition of the $V(7)$ makes this reading clearer, though often such a repeated chord would not be notated. This means that, even with the handling of modulations presented here, some of the possible modulation analyses will be missed, since a single chord cannot be interpreted both as the target of a cadence and the beginning of another. This can theoretically be solved by inserting a duplicate of every chord, though in practice this will have too much of an impact on parse time to be practicable.

we use

$$X(m) : \text{trgt} \Rightarrow (I_X(m) \setminus Z(m)_1) \setminus (Y^7 / {}_c I_X(m)^7) : \lambda \text{cad.} \lambda \text{orgn.} \text{orgn} + \text{cad}(\text{trgt})$$

in which Z is a new variable that may unify with any atomic category. This allows a cadence to function as a means of modulation from the starting tonality Z to the ending tonality I_X . The overall category assigned to the section (that covered by the preceding category and this cadence) is that corresponding to the new tonality that the cadence arrives at. Thus, a passage that begins in the key of I and modulates to IV will receive the category IV , with semantics $I + \dots(IV)$ – that is, I concatenated with some cadential recursion onto IV .

As mentioned above, it is common (though not required) for a modulating sequence to return to the initial key by the end of the sequence. Sequences interpreted in such a way will receive the category I overall (assuming the sequence began in the key of I) and will give in their semantics a full description of the sequence of tonalities as cadences onto a point in the tonal space corresponding to each tonality.

As an example of this, consider the A section of *Autumn Leaves* again, this time with an initial key tonic prepended and only as far as the first return to I :

$$\text{Im} \mid \text{IVm}(7) \text{ bVII}(7) \text{ bIII} \text{ bVI} \text{ II}\%7 \text{ V}(7) \text{ Im}(7)$$

The most obvious interpretation of this using the grammar is as a single long cadence: the sequence begins on I and cadences back to I by the end in left steps, including one tritone jump (from bVI to $\text{II}\%7$). However, an alternative and valid interpretation is that it begins on I , modulates via an authentic cadence to the key of bVI , then finally modulates back to I via another authentic cadence. This interpretation may now be produced by the grammar as in the derivation shown in 7 (logical forms are omitted for the sake of space).

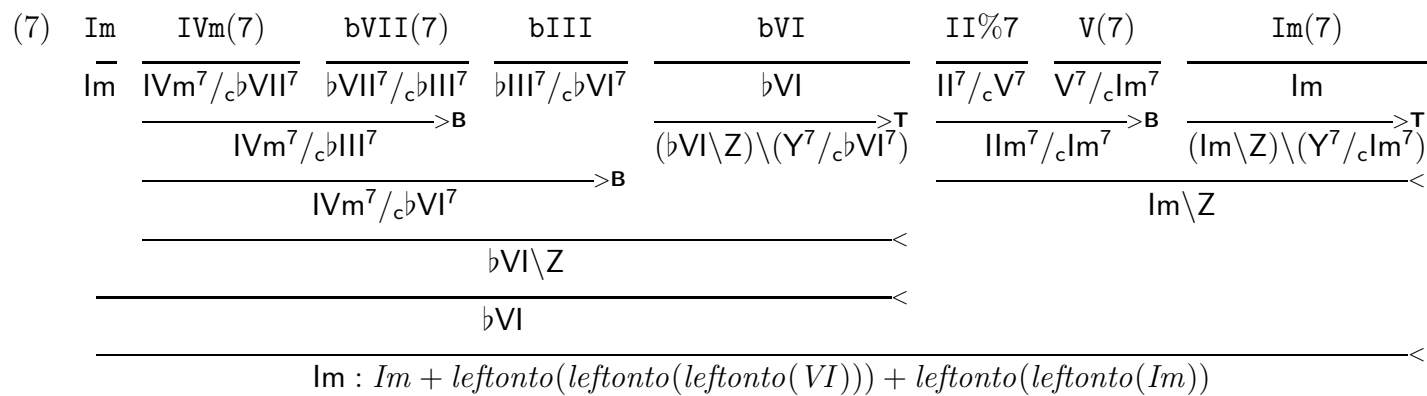


FIGURE 2.1: A CCG derivation for the A section of *Autumn Leaves*

2.8.3 The Blues as a Modulating Form

One of the characteristics of the 12-bar blues can be identified purely from the semantics produced by the grammar. The main structure of a 12-bar blues involves beginning in the tonality of *I*, moving to a section based on a *IV* chord and then finishing with an authentic cadence in the tonality of *I*. The three sections are characterized by their tonality: the first in *I*, the second in *IV* and the third back in *I*. The movement between these tonalities is typically made using authentic cadences, of the sort recognized by the grammar, since these are the most commonly used device for establishing a new tonality. The grammar is now able to interpret these movements using the modulating cadence-raising rule and produce a result that is recognizable as a blues (in this characteristic at least) from only its semantics. We need only look for a *I* at the beginning, some form of authentic predicate structure ending up on *IV* following this, and another authentic predicate structure leading to a *I* at the end (it may return to the *I* before this final cadence, but does not in all cases).

There are further requirements that must be satisfied before a sequence would be recognized by a listener as a 12-bar blues; these and the precise nature of the modulations introduced above are explained in section 3.3. The important point for now is that the grammar can now assign a musically meaningful interpretation to a 12-bar blues that encodes in the semantics this particular characteristic of the form.

2.8.4 The Extreme Ambiguity of Modulations

As I have argued above, the ambiguity introduced by this new freedom allowed for cadences is an ambiguity that exists in human interpretation and is therefore not only justified in the grammar's interpretations but required of them. However, as a result of this ambiguity, the process of parsing chord sequences involving long sequences of dominant seventh chords becomes debilitatingly slow.

As with other components of the grammar, this is not a sign of any fundamental problem with the feature. As I argue more fully in section 4.6, the correct way to overcome problems of parsing a highly ambiguous grammar is to apply statistical parsing techniques. By analogy to natural language grammars, this is likely to be a reasonable simulation of the sort of probabilistic effects that allow humans to parse languages modelled by highly ambiguous grammars. Such a

model is without the scope of this study. For now let us simply note that, with a suitable statistical model, this ambiguity need not be a serious problem for the parser.

However, as with other features that must rely on a statistical model I must impose artificial constraints on the grammar in order that it may be practically usable by the parser. The ambiguity of modulations may be reduced whilst still maintaining a large part of the benefit of fully licensed modulation by restricting the possible modulations to just a small subset of relative movements between keys.

The 12-bar blues is discussed above in the context of modulation. Since recognition of the 12-bar blues is a major focus of this study, it is important that the grammar should not be limited such that it cannot cover the modulations required for this form. The only modulations used in the 12-bar blues form are a modulation to the subdominant key ($I - IV$) and a modulation back to the tonic key ($IV - I$). This means that a cadence-raising rule that restricts modulations to three target keys relative to the starting key – I (for a simple non-modulating cadence) and IV and V (for the $I - IV$ and the returning $IV - I$ modulations) – can produce the correct interpretations of modulations in 12-bar blues examples. This constraint is imposed by adding to the categorial variable notation a restriction of the variable to certain values⁶. A variable may be followed by a choice of the form $\langle A|B|... \rangle$, denoting the allowed roots, relative to the root of the cadence-raised category, with which it may be unified.

Our new, artificially restricted version of the cadence-raising rule for authentic cadences is then:

$$X(m) : trgt \Rightarrow (I_X(m) \setminus Z^{(I_X(m)|IV_X(m)_1|V_X(m)_1)}) \setminus (Y^7 / {}_c I_X(m)^7) \\ : \lambda cad. \lambda orgn. orgn + cad(trgt)$$

⁶Another way to impose the same constraint would be to allow the cadence-raising rule to output several signs for addition to the chart, each allowing one of the origin-target pairs. This would remove the necessity of extending the notation, but would be detrimental to the implementation. A cadence-raised sign thus restricted is only instantiated once and the restriction on the variable is checked when the variable is unified during rule application.

2.9 Diminished Sevenths

The original grammar provided several categories that could be assigned to diminished seventh chords:

$$7. \quad X\circ 7 := \flat V_X / \flat V_X : \lambda x.x \mid \flat II_X / \flat II_X : \lambda x.x \mid \flat VII_X m^7 / \flat VII_X m^7 : \lambda x.x$$

The diminished seventh chord, unlike the dominant seventh, has a highly ambiguous resolution. It may in fact be followed by a chord rooted on any chromatic root. Much has been written in music theory literature about the interpretation of the diminished seventh chord and it is these interpretations that motivate the following reformulation of the 7 categories. The categories in the original grammar, as their semantic representations suggest, interpret all diminished sevenths as colouration, causing no movement in the tonal space. In the new lexicon, the semantics of the category will depend on the interpretation to which it corresponds. In the following sections I motivate a complete reformulation of the categories for this chord type.

2.9.1 Theoretical Background to the Chord's Ambiguity

There are two reasons for the extent of the ambiguity of this chord. The first is that the chord is made up of four notes separated by minor thirds. So, for example, an $A\circ 7$ chord contains A , C , $\flat E$ and $\flat G$. This means that a diminished seventh chord rooted on, say, A may be voiced in precisely the same way as the first inversion⁷ of a the diminished seventh rooted a minor third below it ($\sharp F$), the second inversion of that a diminished fifth below it ($\sharp D$) and the third inversion of that a diminished seventh below it ($\flat G$). It is only by the harmonic context that one may make a judgement as to the likelihood that any one of these notes is the true root of the chord. Therefore, if the chord is expected for one of the reasons below to resolve to a particular chord, without accounting for the harmonic context, it may equally be expected to resolve to the chords rooted a minor third, diminished fifth and diminished seventh above. For this reason, diminished seventh chords rooted on these four tones may be considered in some sense equivalent, since they share interpretations. There are therefore only three distinct diminished seventh chords, each occupying four of the twelve possible

⁷An *inversion* of a chord is a voicing of the chord which does not have the chord root as the lowest note. The *first inversion* is the chord which has the second chord note from the root as its bass note. Similarly, the *second inversion* has the third note up as its bass.

roots.

The second reason for ambiguity is that the diminished seventh chord may appear to perform different harmonic functions. Cork (1996) mentions the three acceptable resolutions, but does not explain why the chord may function in these ways. In actual fact, a true diminished seventh chord may perform only one function. However, there are two other types of chord that are voiced in exactly the same way as a diminished seventh chord and which perform different functions. However, these chords will almost invariably, and indeed very reasonably, be notated in jazz chord sequences as diminished seventh chords. The background to each type of chord is given in the following sections and its effect on the grammar is explained.

2.9.2 The True Diminished Seventh

A true diminished seventh chord behaves in the same way as a dominant seventh chord. The simple diminished triad⁸ (rarely used on its own in jazz), consists of the root (prime) note, its minor third and its diminished fifth. This is introduced by Clarke (188?) as an alternative, with equivalent resolution, to the dominant seventh chord. A chord that is extensively used in jazz is the *half-diminished* chord: a diminished chord with an added minor seventh. This chord always has a dominant function. The true diminished seventh chord is very similar and has identical function to both the diminished triad and the half-diminished chord.

This interpretation of a diminished seventh is supported by ample evidence from standards. The second line of *A Ghost of a Chance*, for example, contains this authentic cadence onto *I*:

$$\text{IIIIm}(7) \text{ bIII}\flat 7 \text{ IIIm}(7) \text{ V}(7) \text{ I}$$

Here the $\text{bIII}\flat 7$ behaves in the same way a $\text{VI}(7)$ would: as a left step in the sequence of left steps eventually leading to *I*. Note that $\text{bIII}\flat 7$ could equivalently be notated as $\text{VI}\flat 7$.

Since the true diminished seventh is equivalent in its function to a dominant seventh, it is clear that the function it performs in the tonal space is a left step. A category that interprets the chord thus must therefore have the left-step

⁸It is conventional to refer in jazz to the diminished seventh chord as the “diminished” chord. However, I will only use the term “diminished chord” to refer to a classical diminished triad (without a diminished seventh tone), since the diminished seventh has its own name and “minor, flat five” is needlessly cumbersome.

semantics of the dominant seventh categories. The categories also bear the “c” cadential modality on the slash, since they will function as steps in a cadence. The grammar must therefore include a category such as the following. (The 1 index is required since index 0, or no index, is bound by convention to an optional minor on the left-hand side, even though there is none here.)

$$7. \text{X}\circ 7 := (\text{h}) I_X^7 / \text{c}IV_X(m)_1^7 : \lambda x. \text{leftonto}(x)$$

Since the diminished seventh chord may be notated in any of its inversions, the grammar must also allow for the possibility that the chord written as $\text{I}\circ 7$ is in fact a $\text{bIII}\circ 7$, $\text{bV}\circ 7$ or $\text{bVII}\circ 7$. This requires the inclusion of the following additional categories:

$$\begin{aligned} 7. \text{X}\circ 7 := & (\text{b}) \text{b}V_X^7 / \text{c}VII_X(m)_1^7 : \lambda x. \text{leftonto}(x) \mid \\ & (\text{e}) \text{bIII}_X^7 / \text{c}\text{b}VI_X(m)_1^7 : \lambda x. \text{leftonto}(x) \mid \\ & (\text{k}) VI_X^7 / \text{c}II_X(m)_1^7 : \lambda x. \text{leftonto}(x) \end{aligned}$$

2.9.3 The Inverted Minor Ninth Chord

The most prevalent theoretical explanation of the diminished seventh chord is as a minor ninth chord – the major chord with added minor seventh and flattened ninth tones – with its prime omitted (Clarke (188?), Schenker (1906) and Pratt (1984)). The minor ninth chord functions in the same way as a dominant seventh – it is in fact a dominant seventh with an added flattened ninth tone. This function is not affected by the omission of the prime, since the notes that give the chord its strong expectation of the dominant resolution – the III and bVII (Cork (1996)) – are still present. The interpretation, then, is of the chord as resolving onto the perfect fifth below its true root (which is a major third below the apparent root) by a left step in the tonal space.

This interpretation is also supported by numerous examples from jazz standards, such as the cadence that begins the A section of *The Joint is Jumpin'*:

$$\text{I } \text{bII}\circ 7 \text{ II}\text{m}(7) \text{ V}(7) \text{ I}$$

The minor ninth interpretation of the $\text{bII}\circ 7$ allows us to treat this chord as the first step of the cadence.

Clarke (188?) tells us that the chord may appear in all of its inversions. Therefore, as before, we must take account of the inverted chords this apparent diminished seventh could be, as well as the simple resolution of a $\text{I}\circ 7$ chord onto

$\flat\text{II}$ (the missing root of the minor ninth chord being $\flat\text{VI}$, left-stepping onto $\flat\text{II}$).

This gives us the following categories to add to the grammar:

7. $\text{X}\circ 7 :=$ (c) $\text{IV}_X^7 / \flat\text{VII}_X(m)_1^7 : \lambda x.\text{leftonto}(x) \mid$
 (f) $\text{II}_X^7 / \flat\text{V}_X(m)_1^7 : \lambda x.\text{leftonto}(x) \mid$
 (i) $\text{VII}_X^7 / \flat\text{III}_X(m)_1^7 : \lambda x.\text{leftonto}(x) \mid$
 (l) $\flat\text{VI}_X^7 / \flat\text{II}_X(m)_1^7 : \lambda x.\text{leftonto}(x)$

Of particular note is that these categories have a set of resolutions distinct from those of the previous section, which distinguishes them from the previous type.

2.9.4 A Chord of Simultaneous Leading Notes

Clarke (188?) describes what he calls *transient chords*, which is much the same as what Schenker (1906) refers to as *passing harmony*. The term *passing note*, or *transient note*, describes a note in the voicing that is not a part of the harmonic progression of the music, but which forms a join between notes that are. A transient chord is the simultaneous use of transient notes in all lines of the harmony. The result is what appears to be a diminished seventh chord, but is in fact not part of the harmonic progression and should be ignored during harmonic analysis. Such a chord is no more than colouration of the harmony and it serves no function in terms of tonal space movements. The chord is nevertheless frequently notated in chord sequences. We can tell which diminished seventh chords are of this sort since the resolutions of this type are again distinct from either of the two discussed above.

Clarke (188?) summarizes this in his question and answer:

Q. Are the inversions of [transient] chords employed?

A. They are: and assume the appearance of inversions of the minor ninth, from which, however, they are distinguished by their resolution.

Although Clarke (188?) does not handle true diminished seventh chords (he considers only diminished chords, with the same function), so does not mention them here, the resolution of transient chords is also distinct from that of true diminished sevenths. There are plenty of examples to be found in Elliott (to appear 2008) of transient chords notated as diminished sevenths, so the grammar must include categories for this type as well.

Since transient chords have no effect on movements in the tonal space, the semantics produced by this interpretation is the identity function. In this case, we effectively ignore the chords, as the original grammar did for all diminished sevenths. As before, we handle all inversions identically. The following categories, then, are added:

7. $X\circ 7 :=$ (a) $I_X(m)_1/I_X(m)_1 : \lambda x.x \mid$
 (d) $VI_X(m)_1/VI_X(m)_1 : \lambda x.x \mid$
 (g) $\flat V_X(m)_1/\flat V_X(m)_1 : \lambda x.x \mid$
 (j) $\flat III_X(m)_1/\flat III_X(m)_1 : \lambda x.x$

2.10 Temporal Information

2.10.1 Adding Temporal Information to the Signs

In section 3.1, the importance of temporal information in the perception of music is explained. For now, note that the time at which an event (a movement in the tonal space) occurs, in terms of some musical abstraction of time, such as beats or bars, must be associated with the event itself in semantic representations produced by the grammar. For most purposes considered in this study, temporal information at the granularity of bars will be sufficient. Where necessary, fractional bars may be used.

At the point of the assignment of lexical categories to chords, the time of the chord's onset can be computed simply by summing up the amount of time that has passed. Chords may be sustained for different lengths of time and this may be notated in various ways. For simplicity, and because it is sufficient for current applications, it will be assumed that, unless otherwise specified, a chord in a sequence lasts for a whole bar. A series of chords in the input which is separated by commas is assumed to fill a bar and, in a further simplification, all the chords in this series are assumed to have equal duration. This means that the duration of the bar is split evenly between its chords.

An onset time is associated with each of the lexical signs. The times are expressed in terms of bars elapsed (starting from 0), with fractional values used where smaller units are required. These time values are associated with what will be referred to as *time-bearing semantic objects* in the logical form of the chord's lexical sign. Time-bearing objects are predicates, such as *leftonto* and *tritone*, and tonal denotations, such as *I*, as well as variables, such as *x*. Predicates and

tonal denotations will be required to bear time values in the semantics of a final result. The time values on variables will be used during derivation and may end up on a predicate or tonal denotation. When a time value is associated with a lexical sign, it is given to each time-bearing object in the sign's semantics.

A new notation is introduced to represent the time values associated with time-bearing semantic objects. An object x that occurs at time 3 is written $x^{\textcircled{3}}$. Thus, a cadence beginning at time 0, left-stepping at 3 and 4, and arriving back at I at 6 would be written $I^{\textcircled{0}} + \textit{leftonto}^{\textcircled{3}}(\textit{leftonto}^{\textcircled{4}}(I^{\textcircled{6}}))$.

It is also necessary for signs to have a duration value, denoting the duration of the contiguous chords it covers. For the most part, this will only be used to test the length of final parse results, so it could equally be calculated and stored at the same time as the assignment of time values to lexical signs. However, assigning durations to all signs has the same effect and allows for the possibility of using the durations for other purposes in the future.

Where it is useful to notate sign durations, a sign $X : x$ with duration n is written $(X : x)^{\sim n}$. Equivalently, the whole semantics of the sign may be written $x^{\sim n}$. This, of course, only makes sense where x is the top-level semantic representation of a sign and not where it is a subexpression in another semantic representation.

The following example shows the time values in bars assigned to semantics and duration values assigned to signs for chords in a short sequence:

	I(M7),	IV	I	IV(7),	bVII(7),	bIII _m (7),	II(7)	V(7),	I
Time	0	$\frac{1}{2}$	1	3	$3\frac{1}{4}$	$3\frac{1}{2}$	$3\frac{3}{4}$	4	$4\frac{1}{2}$
Duration	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$

2.10.2 Propagation of Temporal Information

Some additional processing must be done when producing new signs by application of grammatical rules to ensure that the temporal information associated at the lexical level is correctly propagated to the final signs. Time values must be passed around as logical forms are combined during parsing. Note, for example, that in a sequence I V V7 I I V7 I, with semantics $I + \textit{leftonto}(I) + \textit{leftonto}(I)$, the correct time values are $I^{\textcircled{0}} + \textit{leftonto}^{\textcircled{1}}(I^{\textcircled{3}}) + \textit{leftonto}^{\textcircled{5}}(I^{\textcircled{6}})$. That is, the time value of the first *leftonto* comes from its beginning V even though its semantics came from the V7. Similarly, the time value of the middle I comes from the first I of the pair, even though its lexical logical form was $\lambda x.x$.

2.10.2.1 Sign Duration Propagation

Durations are assigned to whole signs at the lexical level. When any two signs are combined by a binary rule, the duration given to the resulting sign is the sum of the durations on its inputs. Since signs derive from contiguous, non-overlapping sequences of chords, this will give the correct durations to the resulting sign. During unary rule application, the duration is simply copied from the input sign to the result.

2.10.2.2 Semantic Time Value Propagation

It is not necessary to have multiple signs on the same arc of the chart which are identical apart from the time values on the semantic objects. The time values do not affect the applicability of rules and only affect the time values on the results of rule applications, not their predicate structure. A semantic object is therefore given a set of time values, denoting all the times at which this event can occur.

Each time-bearing object has a *time assignment mapping*. This is a 1-1 mapping from time assignment indices to time values that may be assigned to this object. When lexical entries are inserted into the chart, there is only one possible assignment of times to the semantic objects. Time-bearing objects within the same sign's semantics all receive a mapping from the same assignment index to the same time value (the time value of the chord in the input). Every lexical logical form uses a distinct time assignment index.

For example, a sign I may have a time value associated with it as $I^{\{0 \mapsto 3\}}$, indicating that, under time assignment 0, it occurs at time 3. $leftonto^{\{0 \mapsto 2, 1 \mapsto 4\}}(I^{\{0 \mapsto 3, 1 \mapsto 5\}})$ contains two possible time assignments: 0, in which the *leftonto* occurs at time 2 and the I at 3; and 1, in which the *leftonto* is at time 4 and the I at 5.

During parsing, a sign B may be added to the chart which is identical apart from its objects' time assignments to a sign A already added to the same arc. Any time assignment mappings associated with the objects in B are added to the corresponding objects in A. This operation is simple, since the time assignment indices are already distinct.

When any rule is applied to a sign, the semantics of the result is always given a distinct set of assignment indices from the input. This ensures that, whenever two signs are combined by a binary rule, they will always have distinct sets of

indices. Before the rule is applied, we produce a 1-to-n mapping from the indices used in the input semantics to a new set of indices (potentially increasing the number of assignments by splitting a single index into multiple indices). We then continue with the rule application and finally apply the mapping to all results.

When a unary rule is applied to a sign, we simply map each assignment index to a single fresh index. When a binary rule is applied, the result must contain time assignments for every possible combination of time assignments in the two inputs. A mapping is produced that maps each input assignment index to a new set of indices, each representing a different combination of that time assignment with a time assignment in the other sign. Essentially, this is a map from the cross product of the two input sets of indices to a set of fresh indices.

It is not necessary to alter the time values themselves in any way when rules are applied, but the following definition is used to define the resulting time assignment set when β -reducing a function application. It ensures that the times that appear in the resulting semantics adhere to the observations made above about the starting times of predicates and tonal denotations.

The set of time assignments on the result of β -reduction of the application of a λ -abstracted functor $\lambda x.A^{\textcircled{T}_0}$ to an argument B is T_R , where T_1 is the set of time assignments that would otherwise be associated with the result of β -reduction (i.e. $C^{\textcircled{T}_1}$, where $C = A[B/x]$ and is in β -normal form).

$$T_R = \{(i \mapsto \min(\tau_0, \tau_1)) \mid (i \mapsto \tau_0) \in T_0, (i \mapsto \tau_1) \in T_1\} \quad (2.1)$$

2.11 A New CCG Chord Grammar

The sections above have described various developments made to the original grammar. These result in a CCG grammar with the following lexical categories:

- 0a. $X(m) := I_X(m)/I_X(m)_1 : \lambda x.x$
- 0b. $X(m)^{(7)} := I_X(m)^7/I_X(m)_1^7 : \lambda x.x$
1. $X(m) := I_X(m) : X$
2. $X(m) := V_X(m)\backslash V_X : \lambda x.x$
3. $X(m)^7 := I_X(m)^7/cIV_X(m)_1^7 : \lambda x.leftonto(x)$
- 4a. $X(m)^7 := I_X(m)^7/c\backslash V_X(m)_1^7 : \lambda x.tritone(x)$
- 4b. $X(m)^7 := I_X(m)^7/cVII_X(m)_1^7 : \lambda x.tritone(leftonto(x))$
5. $X(m) := I_X(m)/cV_X(m)_1 : \lambda x.rightonto(x)$
6. $Xm := \backslash VII_X m \backslash \backslash VII_X m : \lambda x.x$
7. $X\circ 7 :=$ (a) $I_X(m)_1/I_X(m)_1 : \lambda x.x$ |
 (b) $\backslash V_X^7/cVII_X(m)_1^7 : \lambda x.leftonto(x)$ |
 (c) $IV_X^7/c\backslash VII_X(m)_1^7 : \lambda x.leftonto(x)$ |
 (d) $VI_X(m)_1/VI_X(m)_1 : \lambda x.x$ |
 (e) $\backslash III_X^7/c\backslash VI_X(m)_1^7 : \lambda x.leftonto(x)$ |
 (f) $II_X^7/cV_X(m)_1^7 : \lambda x.leftonto(x)$ |
 (g) $\backslash V_X(m)_1/\backslash V_X(m)_1 : \lambda x.x$ |
 (h) $I_X^7/cIV_X(m)_1^7 : \lambda x.leftonto(x)$ |
 (i) $VII_X^7/cIII_X(m)_1^7 : \lambda x.leftonto(x)$ |
 (j) $\backslash III_X(m)_1/\backslash III_X(m)_1 : \lambda x.x$ |
 (k) $VI_X^7/cII_X(m)_1^7 : \lambda x.leftonto(x)$ |
 (l) $\backslash VI_X^7/c\backslash II_X(m)_1^7 : \lambda x.leftonto(x)$

The rules permitted for use with the grammar are function application, composition with conditions to support cadential slash modes, and the new forms of the cadence-raising rules for authentic and plagal cadences:

- (\gt) $X/\phi Y : f \quad Y : y \Rightarrow X : f y$
- (\lt) $X : x \quad Y \setminus_{\phi} X : f \Rightarrow Y : f x$
- ($\gt\mathbf{B}$) $X/m_1 Y : f \quad Y/m_2 Z : g \Rightarrow X/m_3 Z : \lambda x.f (g x)$
 where $\begin{cases} m_3 = c & \text{if } m_1 = c \text{ or } m_2 = c, \\ m_3 = \phi & \text{otherwise} \end{cases}$
- ($\lt\mathbf{B}$) $X \setminus_{m_1} Y : f \quad Z \setminus_{m_2} X : g \Rightarrow Z \setminus_{m_3} Y : \lambda x.f (g x)$
 where $\begin{cases} m_3 = c & \text{if } m_1 = c \text{ or } m_2 = c, \\ m_3 = \phi & \text{otherwise} \end{cases}$
- ($\gt\mathbf{B}_{\times}$) $X/m Y : f \quad Y \setminus_{\phi} Z : g \Rightarrow X \setminus_m Z : \lambda x.f (g x)$
- ($\lt\mathbf{B}_{\times}$) $X/m Y : f \quad Z \setminus_{\phi} X : g \Rightarrow Z/m Y : \lambda x.g (f x)$
- (\mathbf{T}_a) $X(m) : target \Rightarrow (I(m)_X \setminus Z^{(I(m)_X | IV(m)_{1X} | V(m)_{1X})}) \setminus (Y^7 /_c I(m)_X^7)$
 $: \lambda cadence.\lambda origin.origin + cadence(target)$
- (\mathbf{T}_p) $X(m) : target \Rightarrow (I(m)_X \setminus I(m)_X) \setminus (Y /_c I(m)_X)$
 $: \lambda cadence.\lambda origin.origin + cadence(target)$

Chapter 3

Semantic Interpretation

3.1 Introduction

In other natural languages, as in music in this study, the purpose of a grammar is to produce possible semantic interpretations of any string that is in the language. The grammar described in the previous section produces semantic representations of chord sequences which amount to a harmonic analysis of the sequence in terms of movements in the Longuet-Higgins tonal space. It is a premise of this study that the interpretations produced by the grammar, which look for analyses by small movements in the space, correspond closely to the conscious or unconscious analyses made by composers, listeners and improvisers. These analyses manifest themselves in the form of expectations of chord resolutions and feelings such as coherence and movements towards and arrival at destinations. The premise broadly follows as a consequence of the perceptual significance of small movements in the tonal space due to the strong harmonic relationship between points that lie close to one another.

This study therefore works on the assumption that the Longuet-Higgins semantics produced by the grammar is a good model, for some purposes at least, of human perception of harmony. Particularly notable are arrivals at points in the tonal space (tonal denotations) and generalizations over the movements made to reach these points. For example, a chord sequence with semantics $I + \textit{leftonto}(\textit{leftonto}(I))$ may be described in words as beginning at I , followed by an authentic cadence onto I . Another sequence with semantics $I + \textit{leftonto}(\textit{tritone}(\textit{leftonto}(\textit{leftonto}(I))))$ may also be described so. Such abstractions are useful and are used during harmonic analysis, particularly of jazz,

in which such extended cadences are common. Coker (1964) quotes Richmond Browne on this subject:

The only thing left of a tune after it has really been worked over in modern jazz is the general *functional* structure – in very broad terms, too. The exact root progression is usually gone; the melody is gone, but the number of bars in the tune remains the same, and you generally reach the tonic, dominant, or subdominant in the same measures as they occur in the original version of the tune.

The semantic structures that the grammar produces lend themselves to such generalizations and they are the sort of generalizations we would wish our musical interpretations to be able to make. Since the grammar focuses primarily on cadences, the semantics may be easily generalized to an analysis of a piece in terms of its cadences. As is discussed in section 3.3, the general form of the 12-bar blues, like other seemingly loosely constrained forms, may be described in part in terms of cadences onto tonal denotations.

However, such a description is not sufficient to encapsulate all the characteristics that humans use to recognize musical forms like the 12-bar blues. One respect in which a perceptual model of notated music must differ from a model of written language is that it must take account of temporal characteristics encoded in the notation. The 12-bar blues form provides a good example of where the timing of tonal space movements makes an important difference to how the music is perceived.

In section 3.2 a language of underspecified semantic expressions is described, which may be used to specify a generalized semantic form to which specific semantic instances may be compared. In section 3.3 I relate this to the general form of the 12-bar blues and derive an underspecified semantic expression that encapsulates the main perceptual characteristics of the form. Then in section 3.4 I do the same for another, similarly widely used form – *I Got Rhythm*. In section 3.5 I describe a procedure that may be used to interpret a semantic representation in terms of a concrete path through the Longuet-Higgins tonal space.

3.2 Underspecified Semantic Expressions

In this section I describe an expression language akin to regular expressions which may be used to make explicit generalizations over the movements in the tonal space which are represented in the grammar's semantic representations. The

language provides a means to express constraints on the movements themselves, points in the tonal space, time values associated with time-bearing objects and the duration of a whole semantics.

The next section demonstrates how this language may be used to encapsulate all of the perceptual characteristics that seem to define the 12-bar blues. The language could be similarly used to describe constraints on any similar musical form whose instances are generated by the grammar. I first define a basic expression language to match the simple predicate semantic representations. I then add to this constraints on time values and duration.

3.2.1 Language Definition

The following recursive rules define recursively a *matches* relation between underspecified semantic expressions and fully-specified semantic instances, and, implicitly, the form an underspecified semantic expression may take. If an underspecified semantic expression α does not match a semantics A by any application of the rules set out below, α does not match A .

In the underspecified expressions, square brackets are used in the places where rounded brackets are used in the semantic representations: this allows rounded brackets to be used for grouping as in regular expressions. Here concatenations are written using the predicate *cat*. The parser represents them as multi-argument predicates in this way, though when writing them it is normal to use the more readable infix operator $+$.

1. α matches A if $\alpha = A$.
2. $\beta[\alpha]$ matches $B(A)$ if $\beta = B$ and α matches A .
3. $(\beta\gamma)[\alpha]$ matches A if $\beta[\gamma[\alpha]]$ matches A .
4. $(\beta)?[\alpha]$ matches A if α matches A
or $\beta[\alpha]$ matches A
5. $(\beta|\gamma)[\alpha]$ matches A if $\beta[\alpha]$ matches A
or $\gamma[\alpha]$ matches A .
6. $(\beta)*[\alpha]$ matches A if α matches A
or $((\beta)(\beta)*)[\alpha]$ matches A .

7. $(\beta) + [\alpha]$ matches A if $((\beta)(\beta)^*)[\alpha]$ matches A .
8. $(\alpha|\beta)$ matches C if α matches C
or β matches C .
9. (a) $cat[\{\gamma, min, max\}]$ matches A if $min < 2$
and $max > 0$
and γ matches A .
(b) $cat[\{\gamma, min, max\}, \Gamma]$ matches A if $min = 0$
and $cat[\Gamma]$ matches A .
(c) $cat[\{\gamma, min, max\}, \Gamma]$ matches $cat(A, B)$ if $max > 0$
and γ matches A
and $cat[\{\gamma, pred(min), pred(max)\}, \Gamma]$ matches B .
 $min, max \in \mathbb{N}_0 \cup \{\infty\}$,
 $min \leq max$,
 $pred(x) = \begin{cases} 0 & \text{if } x = 0 \\ \infty & \text{if } x = \infty \\ x - 1 & \text{otherwise} \end{cases}$

It is important to normalize semantic representations so that equivalent forms can always be recognized using the same expressions. There are two reasons why semantic representations which are not identical, but which are equivalent in terms of tonal space movements can be produced by the grammar. Both are related to concatenations and the expression language and expression writers may assume that semantic representations have been normalized in the following ways prior to matching against expressions.

- Rules 9 assume that all concatenations of more than two expressions are written as multiple two-argument concatenations, bound to the right. That is, a concatenation $(A + B + C)$ must be represented as $(A + (B + C))$, and not $(A + B + C)$ or $((A + B) + C)$. This entails no loss of generality and can be enforced through a simple pre-processing of the semantic inputs.
- A semantics that contains a concatenation nested inside a predicate's argument is equivalent to one with the predicate applied only to the first argument of the concatenation, with the other arguments concatenated at the top level in the semantics. That is, $A + p_0(\dots(p_n(B + C))) \equiv$

$A + p_0(\dots(p_n(B))) + C$. When writing expressions it is reasonable to assume that all nested concatenations have been converted to the latter form, leaving only top-level concatenations. Once again, this may be achieved by a simple pre-processing step.

The following further notes apply to the definition of *matches*.

- By convention, an underspecified concatenation may be written $cat[\alpha, etc.]$ as a shorthand for $cat[(\alpha, 1, 1), etc.]$; that is, the expression must match exactly one argument.
- The choice $(\alpha|\beta)$ extends in the obvious way to $(\alpha|\beta|\gamma)$, etc.
- Also by convention, brackets may be omitted around the predicate heads in rules 3, 4, 6 and 7. The ambiguity introduced by this omission is resolved by stipulating that the $?$, $*$ and $+$ operators bind to precisely one semantic object preceding them, where a semantic object is a head choice (rule 5), a predicate head name, any bracketed expression or the result of another such operator¹. In particular, where the operators appear at the end of a sequence as in rule 3, they bind to only the immediately preceding object, not the whole sequence.
- The necessity of rule 3 may not be immediately obvious. It allows operators to be applied to a sequence of head expressions systematically, as in $((tritone)(leftonto)) * [I]$ for example.

In addition to these expressions, the following shorthand notations are useful. They are written simply as named expressions that may be replaced prior to processing of an expression. They allow the common generalizations of unboundedly extended cadences, both plagal and authentic, to be expressed concisely. The plagal cadence is simply a sequence of *rightonto* movements. The authentic cadence is a sequence of at least one *leftonto* movement, but may also include any number of tritone steps.

1. $cadence \equiv (plagal|authentic)$

¹The definition of the result of $?$, $+$ and $*$ operators as semantic objects is included for completeness, but there is no recursive combination of more than one of these operators that is not equivalent to a single operator.

2. *plagal* \equiv *rightonto*+
3. *authentic* \equiv *tritone* * *leftonto*(*tritone*|*leftonto*)*

3.2.2 Temporal Constraints

Every time-bearing object in the semantics of a result has a set of indexed time values associated with it, since all time-bearing objects receive these values when read from the lexicon (see section 2.10). The expression language must allow specification of restrictions on the timing of these elements. For now we are only interested in restricting the times of actual tonal denotations. In effect, this allows us to constrain the timing of the resolution of cadences (potentially modulations). The notation below could be simply extended to express restrictions on any other part of the semantics.

A tonal denotation X may have a time range attached to it: $X^{\textcircled{x-y}}$. x denotes the earliest time at which it may occur and y the latest. A time restriction written $X^{\textcircled{x}}$ is equivalent to $X^{\textcircled{(x-x)}}$ and requires the event to occur at exactly time x .

To match logical forms in such a way that incorporates the checking of these time restrictions, the *matches* relation must be supplemented. The *permits* function is a function from an underspecified semantic expression, a semantics and a set of time assignment indices to a set of time assignment indices.

First let us consider a logical form containing elements bearing time assignment index to time value mappings (that is, time assignments) to be a function from a time assignment index to a fully time-specified semantic expression, in which the indexed time assignment has been selected for all time-bearing objects. Now let us define $T_{\text{permitted}} = \text{permits}(\alpha, A, T_{\text{available}})$ as the subset of the time assignment indices $T_{\text{available}}$ that is mapped by A to semantic representations matched by α . This matching condition is decided using the definition of the *matches* relation given above, supplemented by an additional condition to perform the crucial check of time values on tonal denotations. The relation *matches'* is defined in the same way as *matches*, except that the following condition is added when the expression has a time range:

$$\alpha^{\textcircled{x-y}} \text{ matches}' A^{\textcircled{z}} \text{ if } \alpha \text{ matches}' A \text{ and } x \leq z \leq y$$

And *permits* is defined as:

$$\text{permits}(\alpha, A, T_{\text{available}}) = \{\tau \in T_{\text{available}} \mid \alpha \text{ matches } A(\tau)\}$$

An underspecified semantic expression may also have a constraint at its top level on a whole semantic representation's duration. An expression α which may only match semantic representations of minimum length x and maximum length y is written $\alpha^{\sim(x,y)}$. As with time constraints, duration constraints may be abbreviated as $\alpha^{\sim x} \equiv \alpha^{\sim(x,x)}$. Where no duration constraint is given at all, $\alpha^{\sim(0,\infty)}$ is assumed.

The *conforms* function may now be used in the same way that the *matches* relation was, in order to decide whether a semantic representation A conforms to an underspecified semantic expression α .

$$\text{conforms}(\alpha^{\sim(x,y)}, A^{\sim n}) \text{ iff } \text{permits}(\alpha, A, \text{dom}(A)) \neq \emptyset \text{ and } x \leq n \leq y$$

3.3 The Twelve-Bar Blues

3.3.1 Characteristics of the Form

The 12-bar blues, also known as *Blues Changes*, is the most common form of chord progression used for jazz improvisation and underlies a staggeringly large proportion of pieces in jazz and many of its derivative genres (such as rock and roll). Instances of the general form are recognized by jazz musicians with no difficulty, but constructing a concrete definition is less easy. Here I consider the characteristics of the form that seem to carry most weight in identifying examples as deriving from it.

The most obvious characteristic to begin with is the length of the sequence. A 12-bar blues is in general 12 bars in length. These 12 bars may be repeated ad infinitum, but we assume that we are dealing with only a single instance of the chord sequence. Some examples may in fact be 6 bars long, with the chord changes at double speed, or even 24 bars long. We will assume that any example chord sequence has been transcribed in such a way that it is 12 bars long. Generalizing to 6-bar and 24-bar sequences would be trivial.

Coker (1964) breaks the 12-bar blues form down into harmonic sections as follows: the progression begins in the tonic key (*I*), then moves to a section in the subdominant key (*IV*) beginning at the fifth bar, then returns to the tonic

key for the final six bars. In terms of the restrictions this places on the semantic representations, this means that it must begin with a I , feature an authentic cadence reaching IV at bar 5 and then return to I , re-establishing this key with an authentic cadence.

3.3.2 An Underspecified Semantic Expression

These constraints can all be expressed using the language of underspecified semantic expressions as follows:

$$\text{cat}\{\{I^{\textcircled{0}}\}, \{\text{authentic}[I], 0, \infty\}, \{\text{authentic}[IV^{\textcircled{4}}]\}, \\ \{\text{plagal}[I], 0, 1\}, \{\text{authentic}[I]\}^{\sim 12}$$

The important harmonic movements contained in the expression are the initial I , the authentic cadence to IV , arriving at the start of bar 5 (time 4) and the final authentic cadence onto I . In addition to this skeletal structure, some other movements are permitted. The first section in I is allowed to contain an unlimited number of authentic cadences (in practice it is unlikely that there is more than one). At most one plagal cadence returning from the short IV section to I is allowed; this is to accommodate progressions such as the first, and almost canonical, example in Coker (1964), which moves directly back to I from IV without a cadence:

$$\begin{array}{cccc} I(M7) & IV(7) & I(M7) & I(7) \\ IV(7) & IV(7) & I(M7) & I(M7) \\ V(7) & V(7) & I(M7) & I(M7) \end{array}$$

The final I may be reached at any time, though most commonly it is at bar 11. The overall length is constrained to be 12 bars.

3.4 I Got Rhythm

Another chord sequence used very frequently as the basis for improvisation and composition in jazz, in the same way that the 12-bar blues is, is that of George Gershwin's *I Got Rhythm*, known as *Rhythm Changes*. Like the 12-bar blues, this chord sequence has many variations, preserving only the general functional and temporal structure of the original, but these variations are all recognizable when played as deriving from the same chord sequence. The sequence has an AABA form and I focus here on its A section.

Without the discussion devoted to the characteristics of the blues, I postulate a semantic expression that captures the main characteristics of the A section of *Rhythm Changes*. The sequence begins on the tonic then, perhaps after a short cadence returning to the tonic, modulates to the subdominant via an authentic cadence, arriving there at the start of bar 6. It then returns to the tonic, either by an authentic cadence or directly, by a plagal cadence.

$$\text{cat}[\{I^{\textcircled{0}}\}, \{\text{authentic}[I], 0, \infty\}, \{\text{authentic}[IV^{\textcircled{5}}]\}, \{\text{cadence}[I]\}]^{\sim 8}$$

3.5 Interpretation in the Longuet-Higgins Space

The semantic representations that the grammar produces for signs with atomic categories are made up of directions for movements in the Longuet-Higgins tonal space and points in the space. Although every sequence of movements eventually arrives at a tonal denotation, the tonal denotations themselves do not unambiguously specify a point in the space. A tonal denotation X may refer to the point labeled X , X^+ , X^{++} , etc., or X^- , X^{--} , etc. If the $+$ s and $-$ s are ignored, the space wraps every four horizontal steps left-to-right, shifted down one step every time. The tonal denotations are ambiguous in this way because under the assumption of equal temperament, these pitches are identical.

Furthermore, the parser does not distinguish enharmonic equivalents. Once again this is because, assuming equal temperament, tones that are enharmonically equivalent have identical pitch. Enharmonically equivalent tones are found every four vertical steps in the space. A tonal denotation may therefore denote a number of points in the harmonic space infinite in two dimensions. These equivalences are shown graphically in the tonal space in figure 3.1.

This ambiguity is resolved by the assumption, fundamental to the Longuet-Higgins system, that the sequence of roots proceeds by the smallest possible steps in the tonal space. That is, if a sequence begins on I at (0,0) and steps to III , it will be the III at (0,1) to which it moves, not that at (4,0) or at (-4,2), nor its enharmonic equivalents at (0,4) and (0,-2).

The *leftonto* and *rightonto* predicates define single left and right steps in the space. The beginning of a cadence can be found relative to its end by reversing its steps. The notational ambiguity of the target (final resolution) of the cadence is then resolved by choosing the position for the beginning of the cadence that takes

$\#V^-$	$\#II^-$	$\#VI^-$	$\#III$	$\#VII$	$\#\#IV$	$\#\#I$	$\#\#V^+$	$\#\#II^+$
III^-	VII^-	$\#IV^-$	$\#I$	$\#V$	$\#II$	$\#VI$	$\#III^+$	$\#VII^+$
I^-	V^-	II^-	VI	III	VII	$\#IV$	$\#I^+$	$\#V^+$
bVI^-	$bIII^-$	$bVII^-$	IV	I	V	II	VI^+	III^+
bIV^-	bI^-	bV^-	bII	bVI	$bIII$	$bVII$	IV^+	I^+
$bbII^-$	$bbVI^-$	$bbIII^-$	$bbVII$	bIV	bI	bV	bII^+	bVI^+

FIGURE 3.1: Tonal space diagram. Marked tones are all equal in pitch to I in equal temperament. Circled tones are mapped to I by the detuning of fifths when wrapping the space. Squared tones are enharmonically equivalent to I and its wrapped tones.

the smallest step from the cadence's origin. For example, consider the cadence

$$I^0 + \text{leftonto}^1(\text{leftonto}(\text{leftonto}(\text{leftonto}(I^2))))$$

We assume its origin (0) is at the point (0,0). The target (2) may be at any point labeled by I or its enharmonic equivalents. We know by backstepping that the beginning of the cadence (1) must be at a point (4,0) relative to the target – that is, at any point labeled by III or its enharmonic equivalents. Of all these possible points, (0,1) is closest to the cadence's origin at (0,0). By choosing this point, the ambiguity of the cadence target is resolved to (-4,1).

The *tritone* predicate, however, does not determine an unambiguous movement. As a consequence of assuming equal temperament, a tritone movement to a point (x,y) may come from the point $(x+4, y+1)$ or $(x-4, y-1)$, since they are equidistant from the destination. In a case where two tritone movements follow in close succession, it is most likely that one should be realized as a movement in one direction – say, (4, 1) – and the other in the opposite direction – (-4, -1) – since this results in the shortest overall distance travelled in the space and therefore the smallest discrepancy between the pitch in equal temperament and in just intonation. When building the path, we must allow for both possibilities. At a later stage, we may eliminate any paths that are longer than the shortest

$\#V^-$	$\#II^-$	$\#VI^-$	$\#III$	$\#VII$	$\#\#IV$	$\#\#I$	$\#\#V^+$	$\#\#II^+$
III^-	VII^-	$\boxed{\#IV^-}$	$\#I$	$\#V$	$\#II$	$\#VI$	$\#III^+$	$\#VII^+$
I^-	V^-	II^-	VI	III	VII	$\boxed{\#IV}$	$\#I^+$	$\#V^+$
bVI^-	$bIII^-$	$bVII^-$	IV	$\bigcirc I$	V	II	VI^+	III^+
bIV^-	bI^-	$\boxed{bV^-}$	bII	bVI	$bIII$	$bVII$	IV^+	I^+
$bbII^-$	$bbVI^-$	$bbIII^-$	$bbVII$	bIV	bI	\boxed{bV}	bII^+	bVI^+

FIGURE 3.2: Tritones of I in the tonal space. Solid squares mark the two closest (and equally close) tritones. Dashed squares mark the two other tritones within this region.

possible overall path and assume that the path moves the shortest possible overall distance in the space. The ambiguity of tritones in the tonal space is shown in figure 3.2.

Figure 3.3 expresses in pseudocode the full routine for extracting a path through the tonal space with minimal ambiguity. The path is built up as a linked list. Two structures are needed to represent the path. A node may be a `PathPoint`, which denotes a point in the space and links to the next point, or a `PathChoice`, which represents a branch in the possible paths. A `PathChoice` splits the path into multiple possibilities, storing links to the head of each possible segment of path that may follow, and also has a link to the node that will follow after the branches converge.

The pseudocode uses certain functions to manipulate the path. `prepend_relative_point` adds a point to the start of a path, or to each of the possible first nodes if there is a choice; the new point is given as an offset from the previous first node(s). `remove_duplicate_branches` checks for any branches that result in the same path and eliminates all but one of them. `first_on_path` and `last_on_path` return the set of the possible first or last nodes for all possible branches of a path.

A concatenation `cat(X,Y)` must recursively build paths for its two arguments X and Y and concatenate them, joining each of the possible first nodes in ar-

```

def get_path_from_semantics(semantics):
    if semantics is a tonal denotation:
        point = arbitrary choice of a point denoted by semantics
        return PathPoint(point)
    elif semantics is a predicate:
        case (name of predicate semantics):
            “leftonto”:
                next = get_path_from_semantics(semantics.argument)
                return prepend_relative_point(next, 1, 0)
            “rightonto”:
                next = get_path_from_semantics(semantics.argument)
                return prepend_relative_point(next, -1, 0)
            “tritone”:
                next = get_path_from_semantics(semantics.argument)
                this = prepend_relative_choice(next, [(2,1),(-2,-1)])
                remove_duplicate_paths(this)
                return this
            “cat”:
                pathx = get_path_from_semantics(semantics.argumentX)
                pathy = get_path_from_semantics(semantics.argumentY)
                pathx_ends = last_on_path(pathx)
                pathy_starts = first_on_path(pathy)
                for xend in pathx_ends:
                    possible_ys =  $\emptyset$ 
                    for ystart in pathy_starts:
                        x1 = (ystart.x - xend.x + 1) % 4
                        y1 = (ystart.y - xend.y + 1 + (ystart.x - xend.x + 1) / 4) % 3
                        if (x1,y1) == (3,2):
                            (tritone_x,tritone_y) = (-1,0)
                        elif (x1,y1) == (3,0):
                            (x1,y1) = (-1,1)
                        new_y = transpose_path(ystart,x1-1,y1-1)
                        possible_ys = possible_ys  $\cup$  new_y
                        if tritone values set:
                            tritone_path = transpose_path(ystart,tritone_x-1,tritone_y-1)
                            possible_ys = possible_ys  $\cup$  tritone_path
                        minimal_paths =  $\operatorname{argmin}_{\text{path} \in \text{possible\_ys}}(\text{path\_length}(\text{path}))$ 
                        if |minimal_paths| == 1:
                            xend.next = y_path  $\in$  minimal_paths
                        else:
                            xend.next = PathChoice(minimal_paths)

```

FIGURE 3.3: Pseudocode routine to extract a minimally branching path through the tonal space from a semantic representation

$\sharp V^-$	$\sharp II^-$	$\sharp VI^-$	$\sharp III$	$\sharp VII$	$\sharp\sharp IV$	$\sharp\sharp I$	$\sharp\sharp V^+$	$\sharp\sharp III^+$
III^-	VII^-	$\sharp IV^-$	$\sharp I$	$\sharp V$	$\sharp II$	$\sharp VI$	$\sharp III^+$	$\sharp VII^+$
I^-	V^-	II^-	VI	III	VII	$\sharp IV$	$\sharp I^+$	$\sharp V^+$
bVI^-	$bIII^-$	$bVII^-$	IV	I	V	II	VI^+	III^+
bIV^-	bI^-	bV^-	bII	bVI	$bIII$	$bVII$	IV^+	I^+
$bbII^-$	$bbVI^-$	$bbIII^-$	$bbVII$	bIV	bI	bV	bII^+	bVI^+

FIGURE 3.4: The tonal space showing the region containing the closest instance of each equal temperament tone to the central I . The tritones at the top right and bottom left of the region are equidistant from I . The dashed line marks the rectangular region that is used for the initial transposition, with two adjustments to transform it into the correct region.

gument Y to each of the possible last nodes in argument X , branching the path at the beginning of X as necessary. The path for Y must be transposed in the space to a position equivalent in equal temperament to its initial position, which was arbitrarily chosen when the tonal denotation was evaluated. The start of Y 's path must be placed as close as possible to the end of X 's path. Any branches at the join must be fully expanded and the branches made at the beginning of the whole path, since the different branches may result in different transpositions of Y 's path. If one of the intervals at the join is a tritone, two possible positions for Y 's path must be allowed in the resulting path.

The transposition of the path is done by moving the start point of Y 's path, y_{start} , into the almost rectangular space around the end of X 's path, x_{end} , containing the closest instances of each tone (see figure 3.4). y_{start} is transposed into a rectangular space around x_{end} (figure 3.4, dashed line) and then two adjustments are made: if the point is now at the tritone at the top right corner, the possibility of the bottom left tritone is added; if the point is now at the bottom right of the rectangle ($bVII$ in the figure) it is moved to the point two steps to the left of x_{end} ($bVII^-$ in the figure).

This routine produces an analysis of the underlying harmonic progression of any particular result from the parser, and so together with the parser may produce all possible harmonic analyses of a chord sequence. In conjunction with the preceding sections in this chapter, it is now possible to produce a list of all harmonic analyses of a sequence that correspond to interpretation of it as an instance of a form such as the 12-bar blues.

Chapter 4

Jazz Parser Implementation

4.1 Introduction

In order to test the grammar developed in the previous section, to observe its performance for the purpose of motivating development and to put it to practical use, I wrote a parser, the *Jazz Parser*. This chapter introduces the parser tool, its interface and some of the issues faced during implementation. The parser uses a standard parsing algorithm often used for natural language CCG grammars, with certain modifications to make it appropriate for use with musical grammars. It is written in Python (van Rossum (n.d.)) and uses no libraries not contained in the standard Python 2.3 distribution. It has a command-line interface, whose usage is described in section 4.3.

The parsing algorithm used performs a complete parse, without using statistical techniques. It has been implemented in such a way as to allow easy extension to parsing using a statistical model; this feature is discussed in section 4.6. Since a statistical model is not used, in order to make the parser usable certain restrictions need to be placed on the grammar. These are discussed in section 4.7.

4.2 Parsing Algorithm

The parsing algorithm used to produce derivations with the chord grammar was the CKY algorithm, according to the description of it with respect to CCG grammars in Steedman (2000). The basic top-level parser loop is described in figure 4.1.

```

chart.initialise_lexical_categories()
for end = 1 to num_tokens:
    chart.apply_unary_rules(end-1, end)
    for start = end-2 downto 0:
        for middle = start+1 to end-1:
            chart.apply_binary_rules(start, middle, end)
            chart.apply_unary_rules(start, end)

```

FIGURE 4.1: Pseudocode for the basic top-level parser loop

The spurious ambiguity problem (Komagata (2004)) is overcome by checking each sign that is added to the chart for categorial and semantic equivalence to existing entries on the arc. A further step must be added when handling the musical signs, which contain temporal information as a part of their semantics. Two logical forms identical in their predicate structure, but which differ in their timings are still considered equivalent: the timings will not make any difference to future rule application, so it would be redundant to include both signs. The timing information of the new sign which will not be added to the chart must supplement the timing information on the existing sign. The procedure for doing this is described in section 2.10.2.2.

The parser optionally stores a trace of all the possible derivations of each result during parsing. This may be examined as a derivation tree afterwards. If this is enabled, a further step must be added at the point at which a duplicate sign would be added to the chart. Each sign has associated with it an object that stores the full trace of signs and rule applications that resulted in its derivation. When two identical signs (excluding temporal information) A and B are found for the same arc, only A is added, but the derivation trace for the B must supplement A's derivation trace, so that all possible derivations are represented in the trace. The feature is disabled by default, to avoid the overhead it adds to parsing.

Both of these operations are performed in the chart's functions `apply_unary_rules` and `apply_binary_rules`, which try applying all the grammar's unary and binary rules respectively to the signs on the specified arcs (*start-end*, in the unary case; *start-middle* and *middle-end*, in the binary case). They then add the resulting signs to the arc (*start-end*), performing the equivalence check and supplementing time assignments and derivation traces as necessary.

The chart is stored internally as a two-dimensional array *table* such that the arc (x, y) is stored in the cell $table[x][y - x - 1]$. An arc is implemented as a hash

```

def apply_unary_rules(start, end):
  for sign in table[start][end-start-1]:
    if not sign.unary_rules_applied:
      for rule in unary_rules:
        results = rule.apply_rule(sign)
        if results !=  $\emptyset$ :
          for result in results:
            if store_derivations:
              result.derivation_trace = updated trace from sign.derivation_trace
            normalise_concatenations(result.semantics)
            table[start][end-start-1].extend(results)
          sign.unary_rules_applied = True

```

FIGURE 4.2: Pseudocode for the chart function that applies unary rules to an arc

```

def apply_binary_rules(start, middle, end):
  input_pairs = table[start][middle-start-1]  $\times$  table[middle][end-middle-1]
  for sign_pair in input_pairs:
    for rule in binary_rules:
      results = rule.apply_rule(sign_pair)
      if results !=  $\emptyset$ :
        for result in results:
          if store_derivations:
            result.derivation_trace = updated trace from sign.derivation_trace
          normalise_concatenations(result.semantics)
          remove_unsequential_time_assignments(result.semantics)
          table[start][end-start-1].extend(results)

```

FIGURE 4.3: Pseudocode for the chart function that applies binary rules to an arc

table to speed up the membership checks that must be performed before each sign is added to the chart.

The routines to apply all unary and binary rules to an arc or pair of arcs in the chart are described in pseudocode in figures 4.2 and 4.3. The implementation of the hash table takes care of the correct addition of new signs to an arc, including the operations described above that are required when an equivalent sign already exists. All of this is contained in the `extend` function used in the pseudocode.

The semantics of a sign is transformed into a normal form in which all concatenations are made at the top level and never nested inside predicates. We assume that this has been done when it comes to underspecified semantic matching (see section 3.2.1). It is beneficial to do it before adding signs to the chart, since

otherwise equivalent signs with unnormalized concatenations in their semantics may not be recognized as equivalent.

The application of binary rules involves an additional step after the actual rule application. `remove_unsequential_time_assignments` removes from the semantics any time assignments that violate the monotonicity of time. These are obviously nonsensical and the speed of parsing is improved by removing these meaningless signs. The necessity of this step and a potential better approach are discussed in section 4.7.

4.3 Parser Tool

The Jazz Parser is used by a command-line interface, accepting input from the command line or a file and outputting results to the standard output. In this section, I describe the basic use of the parser through this interface, the form of the output and the interface to some of the more advanced features. The python module `jazzparser.py` provides the main parser interface.

4.3.1 Basic Parser Input

The simplest way to use the parser is to specify an input chord sequence as an argument on the command line. This will use the default grammar (“jazz0.5” – that described in chapter 2) to perform a complete parse of the input chord sequence and will output a list of all the results. Each result is displayed as a CCG sign of the form `Category:Semantics`. Both the category and the semantics are output in a plain text adaptation of the notations used in chapter 2.

```
$ python jazzparser.py "I IIm(7) V(7+5) I(M7)"
```

```
Parsing: I IIm(7) V(7+5) I(M7)
```

```
Results:
```

```
0> (I/Im):\x1.(x1 + leftonto(leftonto(I@{3})))
1> (I\Im):\x2.(x2 + leftonto(I@{3,3}))
2> (I\Zx<I|IV|V>):\x1.(x1 + leftonto(leftonto(I@{3})))
3> (I/I):\x1.(x1 + leftonto(leftonto(I@{3})))
4> I:(I@{0} + leftonto(leftonto(I@{3})))
5> (I\V):\x1.(x1 + leftonto(leftonto(I@{3})))
6> (IIm7\cV7):\x2.leftonto(x2)
7> (I/cVm):\x1.(rightonto(x1) + leftonto(leftonto(I@{3})))
```

```

8> (I/cV):\x1.(rightonto(x1) + leftonto(leftonto(I@{3})))
9> ((I\Zx<I|IV|V>)\(Yx7/cI7)):\cadence0,prior0.(prior0 + (cadence0 (I@{0}
    + leftonto(leftonto(I@{3}))))))
10> ((I\Zx<I|IV|V>)\(Yx/cI)):\cadence0,prior0.(prior0 + (cadence0 (I@{0}
    + leftonto(leftonto(I@{3}))))))

```

Generally, the only interesting results will be those with atomic categories. The `-a` switch causes only these to be output. The `-t` switch includes the parse time in the output.

```

$ python jazzparser.py -a -t "I IIm(7) V(7+5) I(M7)"
Parsing: I IIm(7) V(7+5) I(M7)

```

Only displaying atomic results

Results:

```

0> I:(I@{0} + leftonto(leftonto(I@{3})))
Parse took 0.623096 seconds

```

The `-p` switch displays a summary progress report on the state of the chart after each input chord has been processed.

4.3.2 Interactive Mode

Once the parse is complete, further operations may be carried out on the results using the interactive mode, enabled by the `-i` switch. This will present a prompt, allowing commands to be input. Some of the commands are explained in this section.

The command `chart` displays all the signs in the chart, row by row, as it stood when the parse was completed. For example:

```

$ python jazzparser.py -ai "I IIm(7) V(7+5) I(M7)"
Parsing: I IIm(7) V(7+5) I(M7)

```

Only displaying atomic results

Results:

```

0> I:(I@{0} + leftonto(leftonto(I@{3})))

```

```
>> chart
```

Edges starting at 0

```

(0,1): (I/Im):\x0.x0, (I/I):\x0.x0, (I7/Im7):\x0.x0, (I7/I7):\x0.x0,
I:I@{0}, (V\V):\x0.x0, (V7\V7):\x0.x0, (I/cVm):\x0.rightonto(x0), (I/cV):

```

```

\x0.rightonto(x0), ((I\Zx<I|IV|V>)\(Yx7/cI7)):\cadence0,prior0.(prior0 +
(cadence0 I@{0})), ((I\Zx<I|IV|V>)\(Yx/cI)):\cadence0,prior0.(prior0 +
(cadence0 I@{0}))
(0,2): (I\Im):\x2.x2
(0,3): ...

```

The command `lh` takes a single argument of a result number and displays the path through the Longuet-Higgins space built from the directions in the semantics of that result. The procedure for building the path is outlined in section 3.5. The output for the above example is as follows:

```

>> lh 0
Longuet-Higgins path for result 0 ((I@{0} + leftonto(leftonto(I@{3}))))
I@0, II@1, V@2, I@3

```

The interactive mode can be used to examine derivation trees for individual results. The command `d` displays the derivation tree for the result given as an argument. Derivation trees are only available if the parser was run with the `-d` switch, since they must be built during parsing and are disabled by default for efficiency. The derivation tree shows the rule applications and signs of every possible derivation of the result. The derivation tree output for the single atomic result produced from the example input is:

```

>> d 0
Derivation trace for result 0: I:(I@{0} + leftonto(leftonto(I@{3})))

I:(I@{0} + leftonto(leftonto(I@{3})))
| from Y X\Y => X (<) applied to
| I:I@{0}
| | "I"
| (I\Zx<I|IV|V>):\prior0.(prior0 + leftonto(leftonto(I@{3})))
| | from Y X\Y => X (<) applied to
| | (IIm7/cI7):\x2.leftonto(leftonto(x2))
| | | from X/Y Y/Z =>B X/Z (>B) applied to
| | | (IIm7/cV7):\x0.leftonto(x0)
| | | | "IIm(7)"
| | | (V7/cI7):\x0.leftonto(x0)
| | | | "V(7+5)"
| | ((I\Zx<I|IV|V>)\(Yx7/cI7)):\cadence0,prior0.(prior0 + (cadence0 I@{3}))
| | | from X(m) => (I[x](m)\Z[x](m))\ (Y7/I[x]7) (>T(s)) applied to
| | | I:I@{3}
| | | | "I(M7)"

```

4.3.3 File Input

As well as taking single chord sequences as command-line arguments, the parser can read sequences in from a text file. The command-line option `-f` specifies an input file. Each line is processed as a separate chord sequence. There are several special kinds of lines that will not be treated as chord sequences. Blank lines are ignored. A line beginning “//” is a comment and is also ignored. A line beginning “>>” is also a comment, but the remainder of the line is printed directly out to the standard output. This allows notes on the chord sequences to be included in the output, making it more readable. If either kind of comment begins with the character “=” (i.e. “//=” or “>>=”), as well as being treated as usual the remainder of the line is used as the name of the next chord sequence. This is useful for identifying chord sequences in a results summary (see below).

A range of lines may be selected from a file for input to the parser using the argument syntax `-f <filename>:<start>-<end>`, where `start` and `end` are the start and end line numbers and may be omitted to default to the first and last line respectively.

When multiple chord sequences are being processed at once it is useful to be able to see a summary of the results at the end, rather than having to trawl through lengthy output. The `-r` switch causes a results table to be printed after all sequences have been parsed, including the names of the sequences (if available) and the number of results, atomic results and *I*-category results.

The following example input file demonstrates the form of the input.

```
// Example file input

>>=A short chord sequence
I(M7) IV(7) I(M7) I V(7) I(M7)
>>This is a similar sequence, but demonstrates a tritone substitution
>>=A second short chord sequence
I(M7) IV(7) I(M7) II(7) bII(7) I(M7)
```

The use of this file as input is demonstrated below. The `-a` switch is used to limit the size of the output and the `-r` requests the final results table.

```
$ python jazzparser.py -arf ../input/short_inputs
A short chord sequence

Processing input string: I(M7) IV(7) I(M7) I V(7) I(M7)
```

Parsing: I(M7) IV(7) I(M7) I V(7) I(M7)

Only displaying atomic results

Results:

```
0> I:(IV@{1} + leftonto(I@{5}))
1> I:(I@{0,3,2,3,2} + leftonto(I@{5,5,5,5,5}))
2> V:(I@{0} + rightonto(V@{4}))
3> I:(I@{0,0} + rightonto(I@{3,2}) + leftonto(I@{5,5}))
4> V:(I@{0} + rightonto(I@{2}) + rightonto(V@{4}))
5> V:(I@{0} + rightonto(rightonto(V@{4})))
```

This is a similar sequence, but demonstrates a tritone substitution

A second short chord sequence

Processing input string: I(M7) IV(7) I(M7) II(7) bII(7) I(M7)

Parsing: I(M7) IV(7) I(M7) II(7) bII(7) I(M7)

Only displaying atomic results

Results:

```
0> I:(IV@{1} + tritone(leftonto(tritone(leftonto(I@{5}))))))
1> I:(I@{2,0} + tritone(leftonto(tritone(leftonto(I@{5,5}))))))
2> I:(I@{0} + rightonto(I@{2}) + tritone(leftonto(tritone(leftonto(I@{5}))))))
```

Summary of results

=====

Name	Results	Atoms	Is	Sequence
A short chord sequence	82	6	3	I(M7) IV(7) ...
A second short chord sequence	30	3	3	I(M7) IV(7) ...

4.3.4 Output Format

The plain text adaptation of the notation of signs is for the most part self-explanatory. In time assignments, the assignment indices are omitted. Instead a consistent ordering of assignment values is used throughout a sign, so that the n^{th} time value attached to each tonal denotation refers to the n^{th} time assignment to the whole semantics. By default, for readability only the time values on tonal denotations are displayed in the output. However, predicates and variables also have associated time values, which is why it often appears that there are duplicate time assignments. These assignments in fact differ in the times assigned to these other semantic objects, which are not displayed.

By using the `-l` switch, the results may be output as Latex source. The

whole output constitutes a full Latex document and may be compiled directly. The following is an extract from the output of the above parses with the Latex output option enabled:

```
Parsed string: I(M7) IV(7) I(M7) II(7) bII(7) I(M7)
```

Only displaying atomic results

Results:

1. $I : (IV^{\textcircled{1}} + \text{tritone}(\text{leftonto}(\text{tritone}(\text{leftonto}(I^{\textcircled{5}}))))))$
2. $I : (I^{\textcircled{2,0}} + \text{tritone}(\text{leftonto}(\text{tritone}(\text{leftonto}(I^{\textcircled{5,5}}))))))$
3. $I : (I^{\textcircled{0}} + \text{rightonto}(I^{\textcircled{2}}) + \text{tritone}(\text{leftonto}(\text{tritone}(\text{leftonto}(I^{\textcircled{5}}))))))$

4.3.5 Underspecified Semantic Expression Matching

An underspecified semantic expression may be specified on the command line using the `-m` option. The argument to the option is a textual representation of an expression of the form described in section 3.2. Any superscript notations are simply written as sequential text and ∞ is written “`inf`”. For each chord sequence parsed, every result is tested against this expression and a list of the results that match the expression is output after the main results list.

The expressions may use any of the notation described in section 3.2, including the shorthands `cadence`, `authentic` and `plagal`, and additionally the shorthand `blues`, which expands to the expression for the 12-bar blues given in section 3.3.2, and `rhythmA`, which expands to the *Rhythm Changes A* section expression of section 3.4. When an expression is given, if a results table is displayed it includes an additional column to show the number of results that were found to match the expression.

The following example (output abbreviated) parses the same chord sequences as in the previous sections and matches them against the expression `cat[{I}, {authentic[I], 0, inf}]`.

```
$ python jazzparser.py -arf ../input/short_inputs -m \
    "cat[{I}, {authentic[I],0,inf}]"
A short chord sequence

Processing input string: I(M7) IV(7) I(M7) I V(7) I(M7)
...
```

```
5> V:(I@{0} + rightonto(rightonto(V@{4})))
```

```
Results that match semantic expression "cat[{{I}, {authentic[I],0,inf}}]":
```

```
0 (1)> I:(I@{3,2,0,3,2} + leftonto(I@{5,5,5,5,5}))
```

```
...
```

```
Processing input string: I(M7) IV(7) I(M7) II(7) bII(7) I(M7)
```

```
...
```

```
Results that match semantic expression "cat[{{I}, {authentic[I],0,inf}}]":
```

```
0 (1)> I:(I@{2,0} + tritone(leftonto(tritone(leftonto(I@{5,5}))))))
```

```
Summary of results
```

```
=====
```

Name	Results	Matches	Atoms	Is	...
A short chord sequence	82	1	6	3	...
A second short chord sequence	30	1	3	3	...

4.4 Chord Sequence Input

4.4.1 Textual Chord Input

The input sequences are written in plain text and are whitespace-separated chords specified in roman numeral notation. The roman numerals represent scale degrees in the piece's initial key. Modulations are recognized by the grammar in the form of cadences arriving at scale degrees different from their origin, but the roman numerals are always notated relative to a globally constant key in the input and in the semantic and categorical output.

The choice of key to write the roman numeral chords in does not affect the form of the results produced. If a sequence is transposed by an interval x , the only difference it makes to the results is that the categories and tonal denotations are also transposed by x . Nevertheless, in general it is best to write input sequences relative to the key they begin in, since the most meaningful results can then be expected to be those with the category I . What is more, for the purposes of recognizing a 12-bar blues sequence, the underspecified semantic expression used will only match sequences which do begin in the key I , since the expression language does not currently extend to relative chord roots.

The chord notation used to write jazz chords varies vastly between transcribers. The chord type notation for a minor seventh chord, for example, may be $I_{\text{min}}(7)$, I^- , $I-7$, $I\text{m}^7$, or numerous other variants. In some cases, there are even

symbols whose meanings differ between systems. *Idim*, for instance, can mean a diminished seventh chord or simply a diminished triad. To avoid any confusion or ambiguity, a system is set out here which is used to specify chords in the parser's input and only the notations given here are permitted by the parser.

The input chords are initially checked against the following regular expression.

```
^(b|#)?
(I{1,3}|I?V|VI{0,2})
(m|aug|o7|%7|sus4)?
(\(?6|7|M7|9|b9|b10|13|7\+5|7\+9|7\+11\)\)??$
```

The expression permits many chord strings that are not valid chord types (“*Io7(7)*”, for example), but these will result in an error when the chord is looked up in the lexicon. The purpose of the regular expression is not to validate the chord name, but to split it up into four parts: an optional accidental; the roman numeral chord root; a basic triad type (omitted for major chords); and any of a selection of common additions.

The accidental before a chord name may be *b*, for *b* denoting a flattened note, *#*, for *♯* denoting a sharpened note, or nothing. The roman numeral itself may be a roman numeral between *I* and *VII*. Since equal temperament is assumed, all combinations of accidentals and roman numerals are mapped to a chromatic scale degree in the range 0-11. The notation used for chord types largely follows the conventions used by Steedman (1996).

Table 4.1 describes the names and notation used in the implementation of the parser and throughout this paper. This does not cover all possible chord types: not all combinations of triad and additions are explicitly named here, only those that have non-obvious names or potentially confusing definitions.

Some seventh chords with further additions are permitted as input to the parser. These do not constitute further chord types and, in general, applying the approach of Cork (1996), should be notated simply as dominant seventh chords. Further additions such as 11 and 13 will usually be added only because they appear in a melody line and may be included or not at the discretion of an improviser. However, some of these additions occur fairly commonly in chord sequences, so they are permitted for convenience and are included in the dominant chord class.

Chord type	Notation	Description
Major triad	X	In a simple major triad, no chord type is written.
Minor triad	Xm	A simple minor triad.
Augmented triad	Xaug	The triad formed from the tonic, major third and augmented fifth. Due to its common use, X(7+5) is a permitted alternative to Xaug(7).
Diminished triad	Xdim	The triad formed from the tonic, minor third and diminished fifth. This is a classical diminished triad, not a diminished seventh, commonly referred to in jazz as a “diminished chord”.
Suspended fourth triad	Xsus4	The triad formed from the tonic, perfect fourth and perfect fifth.
Sixth chord	X(6)	A major triad with added sixth.
Major seventh chord	X(M7)	A major triad with added major seventh.
Minor seventh chord	Xm(7)	A minor triad with added minor seventh.
Seventh (or dominant seventh) chord	X(7)	A major triad with added minor seventh.
Ninth chord	X(9)	A seventh chord with a further added ninth (compound second).
Major ninth chord	X(M9)	A major seventh chord – X(M7) – with a further added ninth.
Minor ninth chord	Xm(9)	A minor seventh chord with a further added ninth.
Flat ninth chord	X(b9)	A seventh chord with a further added flattened ninth.
Diminished seventh chord	Xo7	A diminished triad with added diminished seventh. The notation is a plain text adaptation of $X \circ 7$.
Half diminished chord	X%7	A diminished triad with added minor seventh. The “%” is a plain text adaptation of “ ϕ ”

TABLE 4.1: An explanation of chord type notation. The triads and additions shown here are all acceptable input to the parser, along with several other additions.

4.4.2 Other Forms of Input

The parser currently accepts input only in the textual chord format described in the previous section. However, it could be extended to other forms of input, given a procedure to assign lexical signs to the input. A dummy statistical tagger has been implemented which simply assigns all possible lexical signs to an input chord (see section 4.6). The simplest way to extend the parser to another form of input would be to build another tagger component which assigns signs to the new type of input sequences. This approach is reasonable, since different forms of input will in general require different statistical models as they will often introduce different degrees of ambiguity.

For example, the parser may be made to handle MIDI-style note-level musical input. The tagger component may perform preprocessing steps such as transforming the input into block chords and removing passing notes if necessary. A statistical model can then be used to assign signs to the block chords. In this case, the need for a statistical model will be even greater than for the current system, due to the higher ambiguity of the input. In annotating a chord sequence for input to the current system, a human annotator or the transcriber of the chord sequence (possibly the composer) has performed a certain amount of analysis of the music already. They have identified the roots of the chords; only a small amount of remaining ambiguity of chord roots is recognized by the grammar.

4.5 Grammar Specification

Grammars for the parser are specified in an XML format based on that used by OpenCCG (Baldrige (2007)). The default grammar that the parser uses is called “jazz0.5”, but others may be selected using the `-g` option on the command line. A grammar’s XML files are stored in a directory whose name is the name of the grammar and consist of the following files: a grammar description file `grammar.xml`, a morphological component in `morph.xml`, the lexical entries associated with parts of speech in `lexicon.xml` and the rules that the grammar is allowed to use in `rules.xml`.

Bozşahin et al. (2007) detail the XML format for specifying natural language grammars for OpenCCG. For the most part, the format used for musical grammars is identical. Some parts of the natural language grammars are not required

for the musical grammars for the parser, so are omitted. Some additions to the format specification are required to support specialized notation for musical grammars and these additions are described here. The complete new format is described precisely in a modified set of the OpenCCG XSD files.

The Jazz Parser grammar’s equivalent of a word in natural language grammars is a chord and where “word” is used here it refers to an individual chord in the input. Morphological entries in the grammar are never specified for chords on a particular root, but generalize over all roots and are specific to chord types. Before a chord is looked up in the grammar, it is generalized – its root is replaced by X. A valid input chord will then be in precisely the form of a word in a morphological entry or class.

OpenCCG uses various separate XML files to make up its grammar, the locations of which are specified in `grammar.xml`. Of these, Jazz Parser grammars require only the morph, lexicon and rules components.

4.5.1 morph.xml

Entries in `morph.xml` associate words with parts of speech. For Jazz Parser grammars, morph entries may reference particular words, as in the case of diminished seventh chords, which are associated directly with lexicon family 7. However, I have also introduced word classes. A `class` tag defines a named word class and a set of words that are contained in the class. This allows the grammar to generalize over the four chord classes discussed in section 2.2.1.

The classes are defined as follows:

```
<class name="X" words="X X(M7) X(7) X(9) X(13) X(6)"/>
<class name="Xm" words="Xm Xm(7) Xm(6)"/>
<class name="X7" words="X(7) X(b9) X(b10) X(7+5) Xaug X(7+11)
                                X(7+9) Xaug(7) X X(M7)"/>
<class name="Xm7" words="Xm(7) Xm(9) X%7 Xm"/>
```

Morphological entries may then use a `class` attribute as an alternative to the `word` attribute. They may also use an `optional_minor` attribute, taking values of “major” or “minor”. This sets any 0-indexed (or unindexed) optional minors in the associated lexical entry, when it is instantiated, to be major or minor.

As an example, the entries for category 1 are as follows:

```
<entry pos="1" optional_minor="major" class="X"/>
<entry pos="1" optional_minor="minor" class="Xm"/>
```

The `pos` attribute associates the input word with the lexical entry that has part-of-speech “1” – that is, category 1. The first entry may be used for any chords in the class of major tonic chords and causes category 1, with 0-indexed optional minors set to be major, to be read from the lexicon. The second does the same for major tonic chords, but sets optional minors to be minor.

The entry for category 7 is used only for diminished seventh chords, so does not use a `class` attribute, but simply a `word`:

```
<entry pos="7" word="Xo7"/>
```

4.5.2 `lexicon.xml`

`lexicon.xml` contains the lexical category families that are associated with input chords by their parts of speech. The schema differs from the OpenCCG schema mainly in the specification of logical forms for lexical signs. Categories are built up from `complexcats` and `atomcats`, just as in OpenCCG grammars. `atomcats` use a `type` attribute to describe the atomic category itself. This string consists of a chord root, a chord type and an optional dominant seventh symbol (7). The chord type may be empty, denoting a major category, “m”, denoting minor, or “(m)”, denoting an optional minor. If an optional minor is used in the `type` attribute, a `minor_class` attribute may specify an integer index for the class to which to optional minor belongs – the index that is written as a subscript. If the `minor_class` is not given, class 0 is implied.

The semantics is written using the λ -calculus, unlike the OpenCCG grammars, which use Hybrid Logic Dependency Semantics. A new set of tags is used to specify these logical forms, contained within a sign’s `lf` tag. The following table shows the tags that may be used to build λ -calculus expressions.

Tag	λ -calculus equivalent
<code><variable id="x" /></code>	x
<code><abstraction varid="x">Y</abstraction></code>	$\lambda x.Y$
<code><predicate name="leftonto">Y</predicate></code>	$leftonto(Y)$
<code><application>X Y</application></code>	$(X Y)$
<code><semlit name="I" relchord="false"/></code>	I
<code><semlit name="I" relchord="true"/></code>	I_X

Slash modes, as in OpenCCG, are set using the `mode` attribute of the `slash` tag of a `complexcat`. The only mode that is used in the jazz grammars is “c”.

4.5.3 rules.xml

The file `rules.xml` simply lists the grammatical rules and rule variants that the grammar is allowed to use. For jazz grammars, the usual composition and application rules, with their modifications for cadential slash modes, are allowed, as well as cadence-raising and plagal cadence-raising.

`application` tags permit application rules and have the attribute `dir`, which may be `forward` or `backward`. `composition` tags also have a `dir` attribute and also a `harmonic` attribute, which is `true` to permit harmonic composition and `false` to permit crossing composition.

A new tag `cadenceraising` allows the authentic cadence-raising rule and `plagalcadenceraising` allows the plagal cadence-raising rule. They have two Boolean attributes to allow early versions of the grammar which used earlier versions of these rules to be used with the latest version of the parser. These are `modal_cadences`, which determines whether the cadence-raised category should expect its cadence argument to have a cadential slash mode, and `modulating`, which determines whether the origin of the cadence will be allowed to have a category different from the target. The restriction of modulations to the relative *I*, *IV* and *V* is seen as a temporary limitation for practical purposes, rather than a feature of the grammar, so is enabled or disabled by a Boolean switch in the code.

4.6 Statistical Parsing

4.6.1 Requirement of Statistical Methods

The grammar described in chapter 2 is very ambiguous. If it is used in its pure form by the parser, the degree of ambiguity makes the parser unusable. However, this is not a reason to reject the grammar as a model of human processing of music. As with natural language grammars, a small amount of context around a chord can give a good idea of how it should be interpreted. The high level of lexical ambiguity in spoken languages and the ease with which humans are able to process this lexical ambiguity demonstrate that there is within the human language processing mechanisms an efficient, probably in some sense probabilistic, means for resolving ambiguity on the basis of context. The high ambiguity of the harmonic grammar and promise of still higher ambiguity to come, then, do not

reduce its plausibility as model of human harmonic interpretation. The solution to the difficulties the ambiguity poses for parsing is to apply statistical methods, such as those of Collins (1999) and Clark et al. (2002).

Since CCG is a fully lexicalized grammar, a statistical model can be applied to parsing in the form of a supertagger: a component which takes the context into account and chooses the most likely lexical categories for a chord. This can greatly increase parsing efficiency by reducing the number of signs in the chart. Clark (2002) describes the use of a supertagger applied to CCG for parsing using natural language grammars and Clark et al. (2002) demonstrate the use of such a model for parsing the Penn Treebank.

The difficulty with the use of such a model for the present grammar is that there is no large-scale, consistently and cleanly transcribed database of jazz standards, with associated functional harmonic interpretations, currently available from which the model could be generated. It is possible that a relatively small corpus could provide a simple statistical model that would be sufficient to allow the parser to produce musically meaningful results in reasonable times. Alternatively, a larger corpus could be compiled by hand, or other, more readily available forms of input which require less preparation could be considered. However, this is not the focus of the present study. It is assumed here that shorter parse times could be achieved using a statistical model for lexical tagging and that the full ambiguity of the grammar could be handled in this way.

The current lack of a statistical tagger causes problems for parsing. The interpretation of long sequences using the full grammar becomes completely infeasible. It is therefore necessary to put some artificial restrictions on the parsing process. Certain practical steps taken to make parsing feasible are discussed in section 4.7.

4.6.2 Extensibility of the Parser

The Jazz Parser has been built in such a way as to allow easy extension to incorporate any number of alternative statistical tagger components. As well as the basic CKY parsing routine, a second parsing routine takes as an argument an instance of a tagger component class, whose interface is defined by the class `Tagger` in the module `supertagger.tagger`. The parse begins by requesting a set of the most probable lexical signs for each chord from the tagger component and adding these to the chart. Each sign is returned along with the probability

assigned to this lexical entry by the model. It then performs a full parse on the chart. If too few results are returned from the parse, it may choose to iterate and request more signs from the tagger.

A new tagger can be implemented by building a subclass of the `Tagger` class, which provides the required functions. Two dummy taggers have been implemented: one which returns all possible lexical entries at once, with equal probabilities; and one which returns all of the entries two at a time, again with equal probabilities. Given a suitable statistical model, it would be easy to build a new tagger to use in the place of these. A criterion for iteration with less probable signs is also required, if this is to be allowed.

4.7 Speeding Up Parsing

Using the full grammar results in an unusable parser for anything beyond very short chord sequences. I discuss here some steps taken to speed up the parsing process. These are not theoretically sound techniques and better approaches are suggested, but between them they enable practical parsing and are reasonably satisfactory temporary solutions.

Huge ambiguity is introduced by lifting the restriction on cadences that their target must have the same category as their origin. This has the effect of providing interpretations of modulations via cadences, as explained in section 2.8.4. Constraints may be placed on the interval between the origin and target keys of a cadence that is permitted by the grammar. The precise nature of the constraints is described in section 2.8.4. It is important to note, however, that this is merely a practical constraint required to allow parsing of longer sequences. It is not justified on musical or psychological grounds, except perhaps insofar as the limitation may be considered a very crude approximation to the function performed by a probabilistic process in this respect: it permits the most common modulations – those to the most closely harmonically related keys (left and right in the tonal space) – and rejects any others.

Another technique to reduce ambiguity and speed up parsing is to add an extra check on each sign before it is inserted into the chart to ensure that the time assignments on its predicates and tonal denotations do not violate the monotonicity of time. Time assignments are associated with predicates, tonal denotations and variables. Once a logical form contains time assignments on its predicates

and tonal denotations that are unsequential, no further combination with other logical forms during rule applications will restore sequentiality. Therefore, since any final logical form with unsequential time assignments is nonsensical, no meaningful interpretations will be removed by this process.

Nevertheless, the technique is not entirely satisfactory. The semantic representations produced by the grammar should be a product of the purely syntactic parsing process and should not themselves have an effect of the production of signs. It would be far preferable if, rather than removing these nonsensical signs after their production by reference to their logical forms, the grammar ensured that they were not produced in the first place. A better way to restrict the ways in which signs are combined is by using slash modalities. In natural language CCG grammars modalities serve to limit selectively the ability of categories to be combined by non-order-preserving rules. I have not devised a system of enhanced slash modalities to incorporate these ordering constraints as well as the current cadential modalities, but consider this likely to be a better approach than the temporary solution presently implemented.

Chapter 5

Results and Discussion

5.1 Introduction

In the previous chapters I have described a new CCG grammar for jazz chord sequences. This included, among other things, new interpretations of diminished seventh chords, the use of slash modes to distinguish cadential categories, a new handling of tritone substitutions and a means of producing interpretations of certain kinds of modulations. I have presented a method for musical interpretation of the semantics produced by the grammar based on generalizations within the interpretation model of the Longuet-Higgins tonal space. I then described a parsing system that I implemented which uses grammar and interpretation model. In this chapter I describe the successes and failures of these models applied to chord sequences using the Jazz Parser. I first study examples of 12-bar blues sequences, then jazz standards more generally and some examples of non-jazz annotated chord sequences to demonstrate the broader applicability of the models. In this way I evaluate qualitatively the success of the grammar. I discuss some of its failings highlighted by these studies and suggest future directions for development of the theory.

5.1.1 Quantitative Evaluation

At this stage, analysis of the success of the grammar can only be done at the level of individual chord sequences. It would not be meaningful simply to report statistics such as the acceptance rate of chord sequences, so I do not. The purpose of the grammar is to act as a mechanism to produce harmonic analyses of the

chords. Acceptance of a chord sequence by the grammar by no means implies that its interpretation is musically sane, as is demonstrated by *Blues Changes* example (h) from Coker (1964) (see section 5.2), let alone that it corresponds to the way a listener would hear the music when it is played. The only way to decide whether an analysis is “correct” is to make a rather subjective judgement by listening to the music, or better still playing it, and deciding the function that each chord serves in the music.

Nevertheless, many analyses are clearly incorrect and often several alternatives may be considered reasonable. Of course, the first requirement for a correct interpretation is that the chord sequence is accepted by the grammar. Secondly, if the sequence stands alone musically in that it does not require any sort of introduction or conclusion to make musical sense to a listener, the grammar should produce a result with an atomic category. In cases where an annotator has made a judgement of the key of the sequence and the input has been notated relative to this key (this is true of all the examples examined), we would expect this category to be *I*.

Since these requirements are prerequisites to a correct analysis, it would be meaningful to perform an evaluation on the basis of them (or the first two, depending on the input) as part of an assessment of the grammar. However, there is currently no suitable large corpus of chord sequences in a form that could be easily adapted for use as input to the parser. Furthermore, without the use of statistical parsing methods it is not feasible to run the parser on large numbers of chord sequences of the typical length of whole jazz standards (see section 4.6 for more discussion of this). In future work, a corpus should be constructed, for example, from chord sequences in Elliott (to appear 2008) or *The Real Book* (Hal Leonard Corp. (2006)). The same corpus could be used for training statistical models, using a method such as holdout validation or, better, k-fold cross validation to apply the same corpus to training and evaluation.

In this chapter, I therefore examine qualitatively the degree of success of the grammar by examining interpretations produced when the parser meets the basic criteria and sequence on which it does not. The results reported do not constitute a quantitative evaluation: most of the sequences have been selected manually either as examples of musical constructions that the grammar handles or as examples of its failures (hence the success rate of roughly 50%).

5.1.2 Isolating Chord Sequences Musically

An important characteristic of these chord sequences is that they are intended to be played in a continuous loop, accompanying a particular tune and improvisations on it. There are cases in which a sequence does not make sense when isolated. For example, a cadence at the end may only be resolved when the sequence begins again (*Solar*, for example), or the beginning may not lend itself to musical interpretation without the key having been established by previous iterations (such as *Blues Changes* example (*g*) from Coker (1964)). Often in practice a coda or simply a final chord will be supplied after the last repetition and an introduction may lead in on the first time round. This problem is referred to as *circularity* by Pachet (2000). It is therefore necessary in some cases to introduce an additional initial or final chord, or even several chords, to make the chord sequence stand alone as a musically coherent unit. An example of this can be seen in blues example (*g*) in figure 5.1, where a I must be added at the beginning to allow the grammar to accept the sequence.

A similar situation arises as a result of isolating sections from a larger chord sequence. Since the high ambiguity of the grammar calls for a statistical model for practical parsing, which is not at present available (see section 4.6), it is often not practicable to parse chord sequences for whole songs. However, jazz standards tend to be clearly divisible into sections and a good alternative is to analyze them one section at a time. For the same reasons given above, it is usually necessary to take some chords from before the beginning of a section and after the end to make the section a coherent unit. In other cases, it may be impossible to analyze a section on its own. These decisions must be made by a human annotator before the parser is applied to a sequence.

A further discussion of this problem is found in section 5.6.

(a)	I(M7)	IV(7)	I(M7)	I(7)	IV(7)	IV(7)	I(M7)	I(M7)	V(7)	V(7)	I(M7)	I(M7)
(b)	I(M7)	IV(7)	I(M7)	Vm(7), I(7)	IV(7)	♯IV◦7	I(M7)	VI(7)	IIIm(7)	V(7)	I(M7)	I(M7)
(c)	I(M7)	IV(7)	I(M7)	Vm(7), I(7)	IV(M7)	IVm(7)	IIIm(7)	VI(7)	IIIm(7)	V(7)	I(M7)	I(M7)
(d)	I(M7)	IIIm(7), ♯II◦7	IIIm(7)	Vm(7), I(7)	IV(M7)	IVm(7), ♭VII(7)	IIIm(7)	♭IIIm(7)	IIIm(7)	V(7)	I(M7)	I(M7)
(e)	I(M7)	VII◊7, III(7)	VIm(7), II(7)	Vm(7), I(7)	IV(M7)	IVm(7), ♭VII(7)	♭III(M7)	♭IIIm(7), ♭VI(7)	IIIm(7)	V(7)	I(M7)	I(M7)
(f)	I(M7)	IV(7)	I(M7)	♭IIIm(7), ♭V(7)	IV(M7)	♭Vm(7), VII(7)	IIIm(7)	VI(7)	IIIm(7), V(7)	♭VIm(7), ♭II(7)	I(M7)	I(M7)
(g)	♭II(7), ♭V(7)	VII(7), III(7)	VI(7), II(7)	V(7), I(7)	IV(7)	♯IV◦7	IIIm(7)	♭III(M7)	♭VIm(M7)	♭II(M7)	I(M7)	I(M7)
(h)	Im(6)	II◊7, V(7+5)	Im(6)	I(7)	IVm(6)	IVm(6)	Im(6)	Im(6)	II◊7	V(7+5)	Im(6)	Im(6)
(i)	Im(6)	II◊7, V(7+5)	Im	V◊7, I(7)	IV(7)	♯IV◦7	Im(6)	♭IIIm(7), ♭VI(7)	II◊7	V(7+5)	Im(6)	Im(6)

FIGURE 5.1: 12-bar blues examples, from Coker (1964), appendix C.

<i>1930s</i>	I(7)	IV(7)	I(7)	I(7)	IV(7)	IV(7)	I(7)	I(7)	II(7)	V(7)	I(7)	I(7)
<i>Count Basie</i>	I(7)	IV(7), ♯IV◦7	I(7)	V(7), I(7)	IV(7)	♯IV◦7	I(7)	VI(7)	IIIm(7)	V(7)	I(7)	I(7)
<i>Charlie Parker</i>	I(M7)	VII◊7, III(♭9)	VIm(7), ♭VI(7)	Vm(7), I(7)	IV(7)	IVm(7), ♭VII(7)	IIIm(7), VI(7)	♭IIIm(7), ♭VI(7)	IIIm(7)	V(7)	I	I
<i>Bebop (with final I)</i>	I(7)	IV(7)	I(7)	V(7), I(7)	IV(7)	♯IV◦7	I(7)	IIIm(7), VI(7)	IIIm(7)	V(7)	IIIm(7), VI(7)	IIIm(7), V(7), I
<i>Tritone substitutions (with final I)</i>	I(7)	IV(7)	I(7)	Vm(7), I(7)	IV(7)	♯IVm(7), VII(7)	I(7), VII(7)	♭VII(7), VI(7)	IIIm(7)	V(7), IV(7)	IIIm(7), VI(7)	IIIm(7), V(7), I

FIGURE 5.2: Further 12-bar blues examples, from anon. (accessed Aug 2008).

(a)	I(M7)	IIIm(7), V(7)	I(M7)	IIIm(7), V(7)	I(M7), I(7)	IV(M7), ♯IV◦7	V(7)	I(M7)
(b)	I, VIm(7)	IIIm(7), V(7)	I(M7), VIm(7)	IIIm(7), V(7)	Vm(7), I(7)	IV(M7), IVm(7)	I(M7)	I(M7)
(c)	I(M7)	IIIm(7), ♯II◦7	IIIm(7), VI(7)	IIIm(7), V(7)	Vm(7), I(7)	IV(M7), ♯IV◦7	I(M7)	I(M7)
(d)	I(M7), VI(7)	♭VI(7), V(7)	I(M7), VI(7)	♭VI(7), V(7)	Vm(7), I(7)	IV(M7), IVm(7)	I(M7)	I(M7)
(e)	I(M7), ♭III(7)	II(7), ♭II(7)	I(M7), ♭III(7)	II(7), ♭II(7)	Vm(7), I(7)	IV(M7), IVm(7)	I(M7)	I(M7)

FIGURE 5.3: Rhythm changes A section examples, from Coker (1964), appendix C.

5.2 Coverage of Blues Changes

In the case of 12-bar blues sequences, there is a clear criterion for the success of the grammar. In section 3.3, I describe certain characteristics of the form and propose a general expression which encapsulates these characteristics. The assertion is that any musically meaningful interpretation of a sequence recognized as a 12-bar blues sequence by a listener familiar with the form should correspond to this expression.

Coker (1964) provides a series of nine examples of 12-bar blues sequences (see figure 5.1). Of these, only *d* is not accepted by the grammar. The parser produces *I*-category results from all of the accepted sequences. In section 3.3.2 an underspecified semantic expression was given that intends to encapsulate all of the most important functional and temporal characteristics of the 12-bar blues form. The parser recognized seven of the eight accepted sequences as matching the blues expression. Example *h* is not recognized as a blues because the correct semantic interpretation is not produced: the fault lies in the coverage of the grammar rather than the semantic expression. The problems with examples *d* and *h* are investigated in section 5.7. The table of results reported by the parser is shown in table 5.1.

anon. (accessed Aug 2008) gives a number of examples of different styles of 12-bar blues chord sequences, listed in figure 5.2. The sequences *1930s blues*, *Count Basie blues* and *Charlie Parker blues* are all accepted by the grammar and produce interpretations recognized as 12-bar blues (see table 5.1). *Bebop blues* is accepted by the grammar, but does not produce any correct semantic interpretations and therefore reports no matches to the blues expression. *Tritone substitution blues* is not even accepted by the grammar. The reason for both of these failures is what is introduced and discussed in section 5.7.2 under the name *convergent cadences*.

5.3 Coverage of Rhythm Changes

Although the main focus of generalized semantic interpretation has been on being able to recognize examples of the 12-bar blues, I have also suggested in section 3.4 an underspecified semantic expression to describe the functional characteristics of variations on *Rhythm Changes*. Coker (1964) also includes five examples of

Name	Results	Matches	Atoms	Is
Blues example a	716	2	49	35
Blues example b	593	2	40	18
Blues example c	658	4	44	30
Blues example d	4	0	0	0
Blues example e	195	6	17	12
Blues example f	32	0	2	2
Blues example g (with initial I)	128	4	12	12
Blues example h	260	8	19	19
Blues example i	129	3	9	9
1930s blues	386	2	28	15
Count Basie blues	618	3	40	20
Charlie Parker blues	191	6	17	12
Bebop blues (with final I)	1405	0	69	21
Tritone substitution blues (with final I)	0	0	0	0

TABLE 5.1: The table of results from parsing the 12-bar blues examples from Coker (1964) and anon. (accessed Aug 2008). The column “matches” reports the number of results whose semantics matched the expression for the 12-bar blues. Blues examples *d* and *h* do not produce correct interpretations.

variants of the A section (described as “progressions often used as the A section of a tune having A–A–B–A structure”). These are shown in figure 5.3.

Four of these examples are accepted by the grammar (see table 5.2). All of these produce *I*-category results with a variety of reasonable musical interpretations. Furthermore, each produces interpretations that match the underspecified semantic expression for the form. Example (*c*) is not accepted by the grammar. The reason for this is the same as for blues example (*d*): they have the same opening four chords. The problem is discussed further in section 5.7.1.

5.4 Coverage of Jazz Standards

Blues Changes and *Rhythm Changes* provide good examples of how the parser, with its semantic expression matching, can be used to recognize variants on a chord sequence that maintain a functional and temporal structure. Most jazz standards’ chord sequences do not have the same kind of widespread use as the

Name	Results	Matches	Atoms	Is
Rhythm changes example 1	483	6	33	22
Rhythm changes example 2	252	2	16	16
Rhythm changes example 3	142	0	9	9
Rhythm changes example 4	193	2	16	16
Rhythm changes example 5	77	2	6	6

TABLE 5.2: The table of results from parsing the *Rhythm Changes* examples from Coker (1964). The column “matches” reports the number of results whose semantics matched the expression for *Rhythm Changes*.

basis for variations and new compositions. However, the grammar is able to produce harmonic interpretations of the chord sequences of many other standards.

Some examples of jazz chord sequences taken from Elliott (to appear 2008)¹ and Coker (1964) have been used as input to the grammar and a table of results is shown in table 5.3. Many of these produced multiple *I*-category results with sensible musical interpretations of the sequence.

One illustrative example is *Autumn Leaves*, section A. The chord sequence used as input was I | IVm(7) bVII(7) bIII(7) bVI(7) | II%7 V(7) Im I. The initial I has been added because it is required to establish the key before the first bar, which begins with a cadence returning to I. This chord in fact precedes the section each time it occurs. One of the three *I*-category results produced by the parser was as follows:

$$Im : (I^{\{0\}} + \text{leftonto}(\text{leftonto}(\text{leftonto}(\text{tritone}(\text{leftonto}(\text{leftonto}(I^{\{7\}})))))))$$

That is, the sequence is analyzed as a long extended cadence, including a tritone substitution, returning to the tonic. The full ambiguity of modulations, which would allow almost every chord after the first in the cadence to be analyzed as a new key, is not currently permitted (see section 2.8.4), but as it happens this seems the most plausible interpretation in this case.

About half of the examples chosen as demonstrative input and whose results are shown in the table failed to produce *I*-categories or any results at all. From observing these sequences and others not mentioned here, it becomes clear that

¹Some of the chord sequences are not split into sections in the same way as in Elliott (to appear 2008). *Autumn Leaves* has been split up as AABAC (with B half the length of A). *Pennies from Heaven* is split into AA-BC-AB-D.

Name	Results	Atoms	Is
<i>Ain't Misbehavin'</i> , A	0	0	0
<i>Ain't Misbehavin'</i> , B, with initial turnaround	0	0	0
<i>All of Me</i> , A+B	3	0	0
<i>All of Me</i> , A+C	0	0	0
<i>Autumn Leaves</i> , A (with initial I)	43	3	3
<i>Autumn Leaves</i> , B (with preceding I)	12	1	1
<i>Autumn Leaves</i> , final A (with coda)	120	10	10
<i>Blue and Sentimental</i> , A	23	2	1
<i>Blue and Sentimental</i> , B (with lead-in)	166	13	13
<i>Blue and Sentimental</i> , final B	198	14	4
<i>Embraceable You</i> , A	111	9	2
<i>Embraceable You</i> , B (with lead-in)	0	0	0
<i>Embraceable You</i> , C (with lead-in)	0	0	0
<i>The Girl from Ipanema</i> , A	9	1	1
<i>The Girl from Ipanema</i> , B	0	0	0
<i>Here's That Rainy Day</i> , A	0	0	0
<i>Here's That Rainy Day</i> , B (with lead-in)	566	33	12
<i>Here's That Rainy Day</i> , C (with lead-in)	217	5	0
<i>The Joint is Jumpin'</i> , A	2,201	90	90
<i>Pennies from Heaven</i> , A (with final I)	16	0	0
<i>Pennies from Heaven</i> , A+B+C	448	22	18
<i>Pennies from Heaven</i> , A+B+D	743	27	18
<i>Solar</i> (with final I)	0	0	0
<i>Hey Joe</i>	14	1	1
Coker (1964), App. D (10), without repeats	1,942	124	86

TABLE 5.3: Results table of parses of various jazz standards from Elliott (to appear 2008) and Coker (1964).

most failures can be explained by a handful of shortcomings in the grammar, which are discussed individually in section 5.7.

The interpretation of *Hey Joe* given in Steedman (2004) is produced and is in fact the only atomic interpretation. The associated tonal space positions are correctly returned by the tonal space analysis routine.

5.5 Coverage of Other Music

The theories that underlie functional harmony and the musical justification for the components of the chord grammar are not limited to the interpretation of jazz. In previous works that have used grammar formalisms to describe elements of music (in particular, Longuet-Higgins & Lisle (1989) and Lindblom & Sundberg (1969)), it has been assumed that a musical grammar should describe the compositional process of a certain musical idiom. However, once we have abstracted away from specific issues of harmonic realization to a pure harmonic notation, the differences between the harmonic devices used in different idioms are surprisingly small. The grammar constructed here has focused on modelling conventions used in the body of jazz standards. The musical interpretations that the grammar produces, though, are universal. Therefore, the grammar should describe reasonably well a fairly broad spectrum of Western tonal music and, where it does, its interpretations should be valid. In order to demonstrate this, I examine some examples here of the application of the grammar to music seemingly remote from the jazz standards so far considered.

5.5.1 Beethoven, Op. 2, No. 1

The following chord sequence is a hand-transcription of the first page of Beethoven's first piano sonata, *Op. 2, No. 1* in F minor (Beethoven (1932)). I have transcribed the chords to represent as closely as possible the harmony of the music, notating seventh chords, for example, only where realized in the score. They are written relative to the given key of F minor.

Sequence segment	Results	Atoms	Is
i	3717	251	0 (170 <i>bIIIs</i>)
ii	2733	238	238
iii	2931	156	110

TABLE 5.4: Results table summarizing the parses of the three divisions of the chord sequence from the start of Beethoven's first piano sonata.

Im Im V(7) V(7) | Im V(7) Im,II%7 V | Vm Vm Im VIm(7) |
 bVII(7) bIII IVm,IV(7) bVII | IVm,IV(7) bVII IVm,IV(7)
 bVII | bVII(7) bVII(b9),bIII bVII(7) bVII(b9),bIII |
 bVII(7) bIII,bVII bIII,#VIo7 bVII,#VIo7 | bVII,VIo7
 bIII,VIo7 bIII,bVII bIII,bVII(7) | bIII

Since this sequence is too long to be practically handled by the parser, I have split it into three sections. The first of these is precisely as written above. Towards the end of this first section, the piece modulates into Ab major (bIII, the relative major of the original key). The category that would be expected of a correct analysis of this is therefore *bIII* and the parser must relax the artificial restrictions on modulations in order to be able to produce this. The restriction is in fact relaxed for all three sections. The second and third sections remain in Ab major, so I have written their chords relative to this tonic: this does not affect their acceptability by the grammar, but means that the expected category of a correct analysis is *I*, as has previously always been the case.

(i) Im Im V(7) V(7) | Im V(7) Im,II%7 V | Vm Vm Im IVm(7) |
 bVII(7) bIII

(ii) I IIm,II(7) V | IIm,II(7) V IIm,II(7) V | V(7) V(b9),I
 V(7) V(b9),I

(iii) I | V(7) I,V I,#IVo7 V,#IVo7 | V,IVo7 I,IVo7 I,V
 I,V(7) | I

A summary of the parse results for these sequences is shown in table 5.4. The results for section *i* had either *bVII* or *bIII* categories. The modulation to the key of *bIII* was correctly recognized. The constant movement between closely horizontally related chords, in particular coupled with the freedom of modulation, results in a great many musically reasonable interpretations. One that corresponds to perhaps the most obvious human interpretation is the following:

$$bIII : (I^{\@{0,1}} + \text{leftonto}(I^{\@{4,4}}) + \text{leftonto}(I^{\@{6,6}}) + \\ \text{leftonto}(\text{leftonto}(I^{\@{10,10}})) + \text{leftonto}(\text{leftonto}(bIII^{\@{13,13}})))$$

For section *ii*, the parser produced many *I*-category results, including many valid interpretations. As an example, one of the correct results produced was the following:

$$I : (I^{\@{0}} + \text{leftonto}(V^{\@{2}}) + \text{leftonto}(V^{\@{4}}) + \text{leftonto}(\text{leftonto}(V^{\@{7}})) + \\ \text{leftonto}(I^{\@{8\frac{1}{2}}}) + \text{leftonto}(I^{\@{10\frac{1}{2}}}))$$

Section *iii* is similar and the following is one of the many valid analyses that the parser output. Of particular note are the correct cadential interpretations of the diminished seventh chords.

$$I : (I^{\@{2,0,0,2,0,0}} + \text{leftonto}(I^{\@{3,2,2,3,3,3}}) + \text{leftonto}(V^{\@{4,4,4,4,4,4}}) + \\ \text{leftonto}(V^{\@{5,5,5,5,5,5}}) + \text{leftonto}(I^{\@{6,6,6,6,6,6}}) + \text{leftonto}(I^{\@{7,7,7,7,7,7}}) + \\ \text{leftonto}(I^{\@{8,8,8,8,8,8}}) + \text{leftonto}(I^{\@{9,9,9,9,9,9}}))$$

5.5.2 Bach, BWV 553

In a similar fashion, I have transcribed chords for the first page of the first of J. S. Bach's *Eight Short Preludes and Fugues*, in C major (BWV 553, Bach (1952)). The full score for this opening fragment is reproduced, with the transcribed chord sequence, in appendix B. This example is especially interesting as a demonstration of general applicability of the grammar: it uses long extended cadences of the sort seen commonly in jazz standards and even include cadential use of diminished seventh chords and a cadential downward semitone step by tritone substitution (simultaneously in the example). The full chord sequence is:

$$I \ I \ | \ I \ IV \ | \ II \ V \ | \ III \ VI \ | \ II \ V \ | \ I \ \#IV\flat 7 \ | \ VII\text{Im} \ III\text{Im} \ | \\ VI \ II \ | \ V, II \ V, II \ | \ V \ I, V, I, V \ | \ II\text{sus}4, II \ II, V, II\text{sus}4, II \ | \\ V \ V(7), I$$

Once again, this is too long for the parser to handle. However, it can be made feasible without majorly affecting the interesting results by removing some sequences of repeated chords or repeated *V* – *I* movements. Clearly these can be handled by the grammar and interpreted as *leftonto*(*I*) movements.

$$I \ \dots \ IV \ | \ II \ V \ | \ III \ VI \ | \ II \ V \ | \ I \ \#IV\flat 7 \ | \ VII\text{Im} \ III\text{Im} \ | \\ VI \ II \ | \ \dots \ V \ | \ II \ \dots \ | \ V \ V(7), I$$

An examination of the score reveals that the sequence modulates early on to G major (*V*) and remains in this key until the last bar, as made clear by the $F\sharp$ used throughout the middle section and the final resolution to *V* at the start of

the final bar and at various points before. The ear leaves no doubt as to the new key. In fact, the modulation back to I , ready for the repeat of this section and continuation to the next, takes place only in the pedal introduction at the end of the last bar (transcribed here as $V(7) I$ in accordance with its clear function, though the ambiguity of these notes is immense). There are many analyses that recognize this modulation occurring at different points in the sequence, though the F#s from bar 3 onwards favours those that place it early on.

Even with the limitation on modulations discussed in section 2.8.4 in place, there is a variety of analyses in which cadences modulate to V , II and I at different places. Of these, several correspond to good interpretations of the music. One that is particularly interesting is that in which, after initially establishing the new key of V in bar 3, the piece embarks on an nine-step authentic cadence back to V at bar 9. A tritone jump makes the join between the I and the $\#IVo7$ in bar 6, or between the $\#IVo7$ in bar 6 (really a $IV(b9)$ in this analysis) and the $VIIIm$ in bar 7.

The following result, produced by the parser, delivers this interpretation:

$$I : (I^{\@{0}} + \text{leftonto}(V^{\@{3}}) + \text{leftonto}(\text{leftonto}(\text{leftonto}(\text{leftonto}(\text{tritone}(\text{leftonto}(\text{leftonto}(\text{leftonto}(\text{leftonto}(\text{leftonto}(V^{\@{14}})))))))))) + \text{leftonto}(V^{\@{16}}) + \text{leftonto}(I^{\@{17\frac{1}{2}}}))$$

5.6 Circularity

The problem of circularity was introduced in section 5.1.2 and is discussed by Pachet (2000). He solves the problem by using a circular model of time, allowing his analyses to make use of the end of a sequence before the beginning, or the beginning after the end. This solution is not possible for the present grammatical approach, but a similar solution may be: the parser could make use of some of the circular context of a sequence in order to produce its results. I have not presented a solution, but have instead worked around it by manually supplying additional chords before the beginning or after the end in the input to the parser. A similar problem arises when individual sections are analyzed on their own. In both of these cases it is important that the additional context used by the analysis is not considered a part of the temporal structure of the segment or sequence. The temporary solution of manual insertion of context does not allow for this.

An example is *Blues Changes* example (g) from Coker (1964). A I chord

must be supplied at the beginning in order to establish the key. However, simply adding the chord breaks the blues structure, since the sequence is now 13 bars long and has everything shifted forward by a bar. A human is able to use the context from the previous repetition or the intro to recognize this first section as a cadence in the key of *I* and still recognize the *bIII(7)* as the beginning of a 12-bar sequence. The context needed to isolate a section or whole unrepeated chord sequence must in any solution to this problem be considered a part of the semantic interpretation of the music and must contribute to the semantic structure, but must not be considered part of the temporal structure of the fragment. In future work, a solution must be devised that adheres to these requirements.

5.7 Analysis of Omissions

Although many chord sequences are assigned good musical interpretations by the grammar, many others are either rejected altogether or given nonsensical interpretations. In some cases it is clear from a glance at the chord sequence why the grammar would fail to describe it; others are less obvious.

The parser provides some features that aid the process of establishing the points of failure of the grammar. The interactive mode available after a parse allows the whole chart to be inspected, or individual parts of it. It also allows the inspection of derivation trees of individual results. The general procedure applied to analyzing rejected (or badly interpreted) chord sequences was to perform a derivation of what I believed to be the correct interpretation by hand and then to examine the parser's chart and derivations to find what prevented the parser from finding this interpretation.

In most cases the grammar's omissions can be explained by a small number of musical devices that it does not handle. I introduce these here and point to examples of sequences whose interpretation is blocked by them. In some cases it is helpful to run the parser on smaller segments or slightly modified versions the input to verify that the sequence is accepted but for the identified problem. Where possible I suggest ideas for future development of the grammar and the semantics to cover these common cases.

5.7.1 Blues Changes D and Rhythm Changes C Opening

The opening chords of Blues Changes example (d) and of Rhythm Changes example (c) from Coker (1964) are identical and present a problem for the interpretation of both sequences. The problematic motif is I IIm(7) #IIo7 IIIIm(7). The same opening, but without the #IIo7 chord can be seen in *Pennies from Heaven*. By reference to the common form of all *Blues Changes* examples and that of all *Rhythm Changes* examples we know that this segment must be analyzed in the tonic tonality. The correct interpretation of this passage is not clear and I suggest here several possible ways of looking at it, but do not claim that any particular one is correct.

The original grammar of Steedman (1996) includes a category devised solely to handle this opening passage: category 6, not much discussed up to now. In combination with the original 7 categories for diminished sevenths, this allowed the whole IIm(7) #IIo7 IIIIm(7) passage to be treated as nothing more than a substitute statement for IIm(7). In order to decide whether this is a valid analysis, we must look at the context in which it occurs. In the *Blues Changes* example it is followed by a cadence from Vm(7), making this reading a plausible one. In the *Rhythm Changes* example, however, it is followed by a cadence from VI(7), suggesting an interpretation in which the IIIIm(7) has a dominant function as a part of this cadence. This fits neatly with the new interpretation of diminished sevenths, which suggests that the #IIo7 should also have a dominant function, resolving onto IIIIm(7). A third reading that would fit both cases is that the whole I IIm(7) #IIo7 IIIIm(7) passage is a substitute for I.

Unless this reading is adopted, it is necessary to interpret this identical passage in different ways in these two different contexts. The similar opening in *Pennies from Heaven* leads on to the cadence bIIIo7 IIm(7) V(7) I, supporting the idea of the IIIIm(7) as having a dominant function.

Whatever phenomenon is at work here, an extension to the grammar is required to explain it. The old category 6 is no longer of use as it was originally intended. Allowing the #IIo7 to be interpreted as a very localized dominant, preparing the IIIIm(7), but not being interpreted cadentially, would let category 6 do its job in its current form (with the addition of ⁷s to category 6's definition to permit a dominant category for the IIm(7) #IIo7 IIIIm(7) segment). This would give an interpretation to the *Blues Changes* example.

A different approach would be required for the Rhythm Changes example and *Pennies from Heaven*. This might be solved by treating the $\text{IIIm}(7)$ as a tonic substitute for the I , if sufficient evidence can be found for this process occurring elsewhere.

5.7.2 Convergent Cadences Problem

A great many failures of the grammar to accept sequences can be explained by repeated endings of cadences. For example, a cadence onto I may proceed towards its end with $\text{VI}(7) \text{II}(7) \text{V}(7)$, raising an expectation of the final resolution to I , but then repeat the $\text{II}(7) \text{V}(7)$ before resolving. The repetition may even use different but functionally equivalent chords: $\#\text{Vo}7 \text{V}^{\circ}7$ or $\text{bV}(7) \text{bII}(7)$, for example. This is not limited to the final couple of chords of a cadence: a whole long cadence may in fact be repeated before the resolution is supplied. I shall refer to this as the problem of *convergent cadences*.

An alternative interpretation of any instance of this may be provided by the current grammar, though in most cases it is hidden in parses by the modulation restriction imposed for parsing feasibility reasons (see section 2.8.4). Two sequential cadences with a common resolution may be interpreted by the grammar as a cadence that modulates to the key of its final unresolved dominant followed by a cadence that modulates to the key of the final resolution. For example, $\text{I} \text{bIII}(7) \text{bVI}(7) \text{bII} \text{IIIm}(7) \text{V}(7) \text{I}$ (from the A section of *Here's That Rainy Day*) may be interpreted as a modulation to bII followed by a modulation back to I . However, this is not a good musical interpretation of this sequence. The interpretation in which the bII functions as a bII^7 , expecting a resolution to I which is delayed by the repeated cadence ending $\text{IIIm}(7) \text{V}(7) \text{I}$ seems far more plausible.

A similar example is provided by 12-bar blues example (*f*) from Coker (1964) (see figure 5.1). The ending is:

$$\dots \text{VI}(7) \text{IIIm}(7), \text{V}(7) \text{bVIIm}(7), \text{bII}(7) \text{I}(\text{M}7) \text{I}(\text{M}7)$$

We know from the structure of the 12-bar blues that the whole final section is in the tonic tonality, so the analysis of the ending that the grammar permits in which a cadence modulates to V , before cadencing back to I , cannot be correct. The grammar must have a way to interpret what seems like two cadences resolving onto the same chord as a single cadence with a repeated ending.

A valuable insight comes from the linguistic construction of coordination, which has its own grammatical rule in CCG. A cadence like VI(7) IIm(7) V(7) bVIIm(7) bII(7) I is rather like the sentence *Marcel proved and I disproved completeness*. Just as the constituents *Marcel proved* and *I disproved* raise the expectation of objects, resolved in the common object *completeness*, the cadences VI(7) IIm(7) V(7) and bVIIm(7) bII(7) share their resolution in the common “object” I. The linguistic constituents signify their common expectations in their categories S/NP and S/NP ; so do the cadences in categories VI^7/cI^7 and $bVIIm^7/cI^7$. We do not require that the cadences to have a common beginning, signified by the result (left) part of their categories.

It is possible that a new grammatical rule, a *convergent cadences rule*, such as the following, could be called for. This uses a new semantic predicate *converge*.

$$X^7/cZ^7 : x \quad Y^7/cZ^7 : y \quad \Rightarrow \quad W^7/cZ^7 : \lambda\alpha.converge((x, y), \alpha)$$

This could be suitably extended to handle more than two cadences with a convergent resolution. Such a rule could produce a semantics like the following for blues example (d) from Coker (1964), which our semantic expression with trivial modifications could recognize as a 12-bar blues semantics.

$$I + L(T(L(IV))) + converge((\lambda x.L(L(L(L(L(x))))), \lambda x.L(T(L(x))), I)$$

5.7.3 Movements to Relative Majors and Minors

A very common and coherent harmonic transition is from a major chord to its relative minor ($I \rightarrow VIIm$), or from a minor chord to its relative major ($Im \rightarrow bIII$ – that is, $VIIm \rightarrow I$). Haagsmans (1916) introduces *parallel chords* (based on the less intelligible Riemann (1895)) as the functional interpretation of minor chords in major keys and major chords in minor keys. The triads of VIm, IIm and IIIIm are described as the secondary triads to I, IV and V respectively, since they are so closely related. When viewed as a parallel chord, a IIIIm chord is seen as having the same function as a V chord – a dominant function with respect to the tonic I.

In jazz, movements between major and minor chords on the same root and through many levels of dominant function chords are common. A IIIIm chord is likely to have a tonic function or a dominant function with respect to VI. Nevertheless, the relationship between a major chord and its relative (or parallel)

$\sharp V^-$	$\sharp II^-$	$\sharp VI^-$	$\sharp III$	$\sharp VII$	$\sharp\sharp IV$	$\sharp\sharp I$	$\sharp\sharp V^+$	$\sharp\sharp III^+$
III^-	VII^-	$\sharp IV^-$	$\sharp I$	$\sharp V$	$\sharp II$	$\sharp VI$	$\sharp III^+$	$\sharp VII^+$
I^-	V^-	II^-	VI	III	VII	$\sharp IV$	$\sharp I^+$	$\sharp V^+$
$\flat VI^-$	$\flat III^-$	$\flat VII^-$	IV	I	V	II	VI^+	III^+
$\flat IV^-$	$\flat I^-$	$\flat V^-$	$\flat II$	$\flat VI$	$\flat III$	$\flat VII$	IV^+	I^+
$\flat\flat II^-$	$\flat\flat VI^-$	$\flat\flat III^-$	$\flat\flat VII$	$\flat IV$	$\flat I$	$\flat V$	$\flat III^+$	$\flat VI^+$

FIGURE 5.4: The tonal space movement (arrow) corresponding to a root movement between a major chord and its relative minor or vice versa. The III and I (squares) are common to both chords. The VI is only in the VI^m chord and the V is only in the I chord (dashed squares).

minor is very strong and movements between such chords are frequent. Such movements can occur without causing any great perturbation of the functional sequence.

Consider for example the A section of *Embraceable You*:

I $\flat III^o7$ $II^m(7), V(7)$ $III^m(7), VI(7)$ $II^m(7)$ $II^{\%}7, V(7)$ I

This is accepted by the grammar only by interpreting it as modulating to V in bar 3 and back to I by the end. However, this interpretation overlooks the close relationship between the $V(7)$ chord and the $III^m(7)$ chord. The $III^m(7)$ should not need to disturb the dominant function of the $V(7)$ and require the interpretation of V as a new tonality. Instead, by virtue of the parallel relationship between V and III^m , the cadence should be able to be continued from the $III^m(7)$ through to the end, allowing the whole section to be interpreted as a long extended cadence.

There are numerous examples of where this sort of movement prevents the parser from accepting or producing a good musical interpretation of a sequence: for instance in *Here's That Rainy Day*, *All of Me* and *Bewitched, Bothered and Bewildered*.

A possible way to handle these movements would be to add some simple

categories to the grammar such as those below. The *relative-minor* predicate is introduced to denote the north-westerly movement in the tonal space and *relative-major* the south-easterly movement (see figure 5.4).

$$X := I_X/VI_Xm : \lambda x.relative-minor(x)$$

$$Xm := I_Xm/bIII_X : \lambda x.relative-major(x)$$

5.7.4 Movements Down a Major Third

In the previous section, I discussed the view of minor chords as being able to take on the same function as their parallel major chords, as presented in Haagmans (1916) and Riemann (1895). It is not uncommon to see movements from a minor chord X to its bVI_Xm chord. We have seen that it is possible to view downward semitone steps between dominant function chords as a tritone step combined with a left step in the tonal space, by using the tritone substitution, which allows the a chord to function in the same way as the chord rooted on its tritone. The assertion by Haagmans (1916) that a minor chord may act as a substitute for its parallel major suggests that the minor chord III_m or $III_m(7)$ might act as a substitute for $V(7)$, the dominant seventh. This entails an explanation of a $Im \rightarrow bVI$ movement as a left movement combined with a movement to the relative major (in either order).

An example where this may be applied is the B section of *The Girl from Ipanema*:

$$\begin{array}{c} I(M7) \mid \#I(M7) \ \#IV(7) \mid \#Im(7) \ VI(7) \mid II_m(7) \ bVII(7) \mid \\ III_m(7), VI(7+11) \ II_m(7), V(7+11) \mid I(M7) \end{array}$$

One way of looking at this section is as a long extended cadence from the initial $\sharp I$, through II , returning to I as the A section is restarted. This can join into a single cadence if we have a means of interpreting movements from a minor chord to its bVI major as a part of a cadence. This occurs twice during the section: $\#Im(7) \ VI(7)$ and $II_m(7) \ bVII(7)$.

A category that allows the grammar to apply the interpretation of a left step combined with a parallel minor substitution is as follows:

$$X := I_Xm/bVI_X : \lambda x.relative-minor(leftonto(x))$$

This allows the following semantics to be assigned to the sequence:

$$I + relative-minor(L(L(relative-minor(L(T(L(L(L(I))))))))))$$

Alternatively, there may be no need to combine the whole sequence into a single cadence. A different approach to this example is suggested in section 5.7.6.

This *III*m – *I* movement is far less common than the relative minor/major movement of the previous section and the alternative explanation of *The Girl from Ipanema* may be preferable.

5.7.5 Pennies Ending Problem

Although *Pennies from Heaven*, A+B+D (i.e. second half), appears from the results table to have been successfully described by the grammar, there are no correct analyses of it among the results. The chord sequence is as follows:

$$\begin{array}{c} \text{I, IIm(7) IIIIm(7), bIIIo7 IIm(7) V(7) | Vm(7) I(7) IV |} \\ \text{IV(7) bVII(7) I VI(7) | IIm(7) II(7), V(7) I} \end{array}$$

The motif IV(7) bVII(7) featured here occurs elsewhere and causes a problem for interpretation with the grammar. *All of Me* features a cadence ending with ...IIm IV bVII(7) I. *Bewitched, Bothered and Bewildered* contains the phrase I IIIaug(7) IV bVII(7) I.

The grammar allows a *IV* relative to its preceding chord to be ignored. It is common to see a phrase like I IV I, functionally remaining on the tonic, but with the IV substituted for colouration. This is described in Pratt (1984) under the guise of a “pedal $\frac{6}{4}$ ”. One possible interpretation of this IV bVII I motif is that, in the same way as the *IV* may be used as a substituted *I* without changing the tonal centre, the bVII may be introduced as a second-level *IV* substitute. In this case, the whole motif could be simply a *I*-substitute.

However, an explanation that I prefer from playing these examples is that the bVII has a dominant function, resolving to the I. It is entirely plausible that a bVII chord should act as a substitute for a Vm(7), since other than the root they contain all the same notes. Similarly, a bVII(7) could substitute for a Vm(b9). Applying this approach to both chords, a IV may function as a substitute for IIm(7) and a bVII(7) as a substitute for Vm(b9). In *Pennies from Heaven*, this interpretation makes the passage IV(7) bVII I a two-step cadence onto *I*. There is no doubt the by the end of the B section, where a IV is held for two bars, this has become the tonic. The song then modulates back to the original tonic using this substituted cadence, IV IV(7) bVII(7) I, which becomes a substitute for IV IIm(b9) Vm(b9) I. The *IV* – *IIm* transition here is also closely linked by the major to relative minor relation. These two substitutions could be recognised by the grammar by adding a new lexical category for each.

In *All of Me*, the IIIm IV bVII(7) I can be seen as a substitute for the simpler cadence ending $\text{IIIm IIIm(7) Vm(b9) I}$. The cadence from *Bewitched, Bothered and Bewildered* can be viewed similarly. It is made slightly more complicated by the use of an augmented chord, not yet discussed in relation to the grammar. Augmented chords possess a property of root ambiguity similar to diminished seventh chords, so this apparent III chord can be seen as a $\sharp V$ or a $\sharp VII$, and an appropriate addition can be made to the lexicon to handle this ambiguity. Like the diminished triad, the augmented triad generally has a dominant function. Assuming the cadential interpretation proposed for the IV bVII(7) , the IIIaug(7) could have two functions. One is as a chromatic colouration of the preceding I , taking its true root to be I and its function to be tonic. The other is as the start of the cadence, having the true root $\sharp V$ and a dominant function, moving with delayed resolution to the (substituted) IIIm(7) , in the same way that the grammar allows for other dominant chords during cadences.

5.7.6 Abrupt Modulations

The grammar currently has one device by which modulations may be produced: the authentic cadence. This is by far the most common device used to modulate: even when modulating to a remote key, it is most common to prepare the new key using at least one level of dominant to the new tonic. Any such modulation can be recognized by the grammar. However, new keys may be reached without this sort of preparation.

One interpretation of the B section of *The Girl from Ipanema* has already been presented, in which the whole section functions as a slow extended cadence back to the main tonic. However, a listener may feel, as a result of the length of the sustained chords, that a new key has been reached at the beginning of the section and that yet another new key is reached halfway through. This gives the succession of keys $\text{I} - \sharp\text{I} - \text{II} - \text{I}$ in relation to the main tonic. The start of the B section certainly has a feeling of a fresh start somewhere new and unexpected.

The new key halfway through the section can be recognized by the grammar, since it is prepared by its dominant seventh. However, the new key at the beginning of the section, a semitone above the previous key, is not introduced in this way. This form of *abrupt* modulation up one semitone without any device to establish the new tonality is heard commonly. It is unlikely to be seen within the

(a) $\text{Im Im Vm}(7) \text{ I}(7) \mid \text{IV}(M7) \text{ IV}(M7) \text{ IVm}(7) \text{ bVII}(7) \mid$
 $\text{bIII bIIIIm}(7), \text{bVI}(7) \text{ bII II}\%7, \text{V}(7)$

(b) $\text{Im Im Vm}(7) \text{ I}(7) \mid \text{IV}(M7) \text{ IV}(M7) \text{ IVm}(7) \text{ bVII}(7) \mid$
 $\text{bIII bIIIIm}(7), \text{bVI}(7) \text{ bII Im}$

FIGURE 5.5: (a) The chord sequence for *Solar*, from Elliott (to appear 2008). (b) A slightly modified sequence which can be interpreted as a blues by the parser. The Im resolution is added at the end and the reiteration of the end of the cadence is removed.

chord sequences of the style of music that this study is mainly concerned with, but this example could perhaps be interpreted with a category that allows for such modulations.

5.8 Is Solar a Blues?

Pachet (2000) discusses the question of whether any formal system of analysis can recognize Miles Davis' jazz standard *Solar* as a 12-bar blues. He suggests that the main problem with analyzing *Solar* as a blues is that “there is no reasonable theory of the Blues available.” In this work, I have presented a precise expression of the most important characteristics that identify a chord sequence as an example of the 12-bar blues. However, there remain two problems in analyzing *Solar* as a blues using this qualification.

Firstly, there is the problem of circularity, which is addressed in Pachet (2000) and which I have discussed in section 5.6. The important point to note is that, whilst circularity presents a practical difficulty for analysis, it is commonly used in jazz in particular and presents no problem for a human listener. Chords in a sequence that occur outside – just before or just after – an analyzed section form a part of the understood harmonic structure (denoted by our logical forms) but not the temporal structure of the section. In the case of *Solar*, this means that the omission of the final tonic resolution does not prevent a listener from recognizing it as a blues. The Im occurs at the beginning and is supplied as the resolution to the final cadence on the repeat, as well as serving as the start of the sequence. I avoid this problem in much the same way as I have done for other chord sequences – by placing an additional Im at the end of the sequence. In this case, I incorporate it into the final bar so that it does not affect the length of the sequence. A better solution could be developed, but for the time being this

solution mimics closely enough what a human listener would do.

Secondly, the sequence presents an example of the convergent cadences problem, introduced in section 5.7.2. This is a simple example in which the last two function steps of a cadence are repeated before the final resolution. A rule such as that proposed could handle this common device and allow the B section of the sequence to be recognized as a single extended cadence returning to the tonic.

Apart from these two problems, *Solar* conforms to the definition of the blues given in this study. Neither of the problems presents a serious reason why the form of analysis used here to recognize examples of the 12-bar blues should not recognize *Solar*. The circularity problem is one that faces many jazz sequences and certainly does not discredit the suggested definition of the blues. The convergent cadences problem is also a common one and a solution to this is even required to be able to recognize one of the blues examples in Coker (1964). A modified version of the *Solar* chord sequence is given in figure 5.5, which works around both of these problems. The repeated cadence ending is removed and a final tonic is supplied. This sequence is accepted by the grammar and the following interpretation is among the nine recognized as 12-bar blues:

$$Im : (I^{\{0\}} + leftonto(IV^{\{4\}}) + leftonto(leftonto(tritone(leftonto(I^{\{11\}}))))))$$

The formulation here of a precise expression of the characteristics of the 12-bar blues in terms of the semantic analyses presents a partial solution to the problem of Pachet (2000). Both of the remaining barriers to an interpretation by the grammar of *Solar* as a blues can be solved by further enrichment of the grammar. The ability to express and automatically recognize perceptually significant characteristics of chord sequences is an important contribution to the plausibility of the model. The technique demonstrated by this grammar seems a promising approach to computational harmonic analysis, both for the development of practical systems and as models of human interpretation.

Chapter 6

Future Work and Conclusion

6.1 Future Work

The grammar developed here is an improvement on that of Steedman (1996), both in terms of coverage of jazz standards and correspondence to music theory. In section 5.7, problems that have been observed with the grammar's analyses of jazz chord sequences were discussed and possible approaches that future work might take to solving them were suggested. In particular, the grammar focuses heavily on analyzing authentic cadences and other musical devices need to be considered; the grammar will need to be extended to handle these and suitable tonal space movements devised to go with them.

One major subject requiring further study is the use of statistical models to increase parsing efficiency and to allow still further ambiguity of the grammar to be handled by the parser. The basic routines for statistical parsing are already included in the parser, though no statistical model has been implemented. The lack of large corpora suitable for building statistical models of harmony must be addressed. It is possible that even a simple model could greatly improve parsing efficiency without a great loss in the accuracy of harmonic analyses.

I have demonstrated the application of the grammar to music outside the jazz idiom, in order to emphasize the generality of the harmonic analyses produced by the grammar. A more detailed study would be required to establish in what ways the grammar would need to be extended in order for it to serve as a good model for harmonic interpretation for a broader spectrum of music.

A subject not touched upon in this study is the inference of structure in chord sequences. Division into sections has been done manually, but there re-

mains the open question of how this might be automated. As the system stands, any attempt to tackle this task would be hindered by the limitation of the size of sequences that the parser can feasibly handle. However, with a statistical model, longer sequences could be interpreted and the tonal space semantics of the grammar would be likely to prove valuable in locating section boundaries and recognizing repeated and varied sections. This problem is related to those discussed in section 5.6, where it was noted that perceived temporal structure of a chord sequence need not coincide precisely with the actual temporal structure of the chords. If a sequence is to be divided into sections, this must incorporate a way of handling the fact that the semantics of the sections may rely on chords from outside their boundaries.

The grammar processes textual input of notated chord sequences. This is sufficient to demonstrate the process of musical interpretation and is particularly suited to the domain of jazz harmony. If the concept is to be applied to a wider range of music, it would be highly advantageous to be able to handle other forms of input. Furthermore, this leads to interesting questions about how to handle the much greater ambiguity of harmonic interpretation and the subtler details of timing and other aspects of realization currently ignored by the abstraction of chord sequences. Although this work has dealt only with harmonic interpretation, other aspects of music, such as rhythm, as handled by Longuet-Higgins (1979), must be considered to approach a fuller model of musical interpretation.

6.2 Conclusion

In this study, I have carried out extensive development of the CCG chord grammar for jazz chord sequences proposed in Steedman (1996). I have analyzed interpretations the grammar produces of examples of chord sequences for jazz standards. A new language of underspecified semantic expressions allows the expression of generalizations over harmonic interpretations. This has been applied successfully to the problem of recognizing examples of the 12-bar blues forms on the basis of constraints on their movements in Longuet-Higgins' tonal space.

A parser has been built that can interpret chord sequences using the grammar. It is able to test results against underspecified semantic expressions and includes various tools to aid testing grammars as models for interpretation. Some solutions to problems of parsing efficiency were implemented and the more promising

solution of statistical parsing methods has been discussed, but not implemented.

Although a quantitative evaluation of the parser is not at present possible for reasons outlined in section 5.1.1, the success of the grammar has been evaluated qualitatively in terms of the musical accuracy of the analyses it produces and the coverage of common musical devices. It was found that it was able to recognize examples of the 12-bar blues form and could provide good musical analyses of many jazz standards. In section 5.7, I have suggested directions for future development of the grammar and specific ideas to extend the its coverage of jazz chord sequences.

There is good reason to believe that the grammar constitutes a plausible model of human understanding of harmony, particularly in the light of the possibility of making semantic generalizations over musical constructs such as those commonly made by musicians. The grammar treats the language of harmony in a way comparable to the treatment of spoken and written language by similar grammars. It demonstrates the great potential that Chomskian grammars hold for expressing models of the perception of music. I have demonstrated that the grammar is not limited to jazz music and the close relationship between the grammar and the harmonic function of chords suggests that the model should be far more widely applicable to harmonic analysis of tonal music.

Appendix A

Chord Sequences for Jazz Standards

Fragments of chord sequences from the body of jazz standards used as examples of input to the parser are included where appropriate in chapter 5. Below are the complete chord sequences for all of the standards referred to in table 5.3. Most of these sequences are a key-relative adaptation of the sequences given in Elliott (to appear 2008), expressed in the textual notation used for parser input and for chord sequences throughout this paper. The two non-jazz examples are not reproduced here, since they are written in their full form in section 5.5.

A.1 Ain't Misbehavin'

Ain't Misbehavin' has AABA form. Each A section ends with a different turnaround, which is omitted in the isolated A section. The appropriate turnaround is prepended to the isolated B section.

Full sequence:

I, VI(7) IIm(7), V(7) I, III(7) IV, bVII(7) | IIm(7), VI(7)
IIm(7), V(7) I, VI(7) IIm(7), V(7) |
I, VI(7) IIm(7), V(7) I, III(7) IV, bVII(7) | IIm(7), VI(7)
IIm(7), V(7) I, IVm(6) I, III(7) |
VIIm IV(7) VIIm VI(7) | V, III(7) VIIm(7), II(7) IIm(7) VI(7)
IIm(7) V(7) |
I, VI(7) IIm(7), V(7) I, III(7) IV, bVII(7) | IIm(7), VI(7)
IIm(7), V(7) I

A section:

I, VI(7) IIm(7), V(7) I, III(7) IV, bVII(7) | IIIIm(7), VI(7)
IIm(7), V(7) I

B section:

I, IVm(6) I, III(7) | VIm IV(7) VIm VI(7) | V, III(7) VIm(7),
II(7) IIIIm(7) VI(7) IIm(7) V(7) | I

A.2 All of Me

All of Me has ABAC form. A cannot be easily isolated from B or C, so AB and AC are used as input sequences.

Full sequence:

I VII%7 III(7) | III%7 VI(7) IIm |
VII%7 III(7) VIm | II(7+11) IIm(7) V(7) |
I VII%7 III(7) | III%7 VI(7) IIm |
IV bVII(7) I VI(7) | IIm(7) V(7) | I

A+B sections:

I VII%7 III(7) | III%7 VI(7) IIm |
VII%7 III(7) VIm | II(7+11) IIm(7) V(7) | I

A+C sections:

I VII%7 III(7) | III%7 VI(7) IIm |
IV bVII(7) I VI(7) | IIm(7) V(7) | I

A.3 Autumn Leaves

Elliott (to appear 2008) splits *Autumn Leaves* into an AABC form. It seems to make more sense to split it into AABAC – essentially AABA with a short B section and a coda. When isolating sections as input, A, B and A+C are used. An initial key tonic is prepended to the A and B sections.

Full sequence:

IVm(7) bVII(7) bIII bVI | II%7 V(7) Im I(7) |
IVm(7) bVII(7) bIII bVI | II%7 V(7) Im |
II%7 V(7) Im |
IVm(7) bVII(7) bIII bVI | II%7 V(7) Im(7), VIIo7 bVIIIm(7),
bIII(7) | bVI(7+9) V(7+9) Im I(7)

A section:

Im | IVm(7) bVII(7) bIII bVI | II%7 V(7) Im

B section:

Im | II%7 V(7) Im

A+C sections:

Im | IVm(7) bVII(7) bIII bVI | II%7 V(7) Im, VIIo7 bVIIIm(7),
bIII(7) | bVI(7+9) V(7+9) Im

A.4 Blue and Sentimental

Blues and Sentimental has an AABA form. The final A has an additional coda.

Full sequence:

I, bVII(7) VI(7) | II(7) V(7) | II(7) V(7) | I V(7+5) |
I, bVII(7) VI(7) | II(7) V(7) | II(7) V(7) | I I(7) |
IV IVo7 | I I(7) | IV IVo7 | I V(7) |
I, bVII(7) VI(7) | II(7) V(7) | II(7) V(7) | I, VII(7) bVII(7),
VI(7) | II(7) V(7) | I V(7+5)

A section:

I, bVII(7) VI(7) | II(7) V(7) | II(7) V(7) | I

B section:

I I(7) | IV IVo7 | I I(7) | IV IVo7 | I V(7) I

Final B section:

I, bVII(7) VI(7) | II(7) V(7) | II(7) V(7) | I, VII(7) bVII(7),
VI(7) | II(7) V(7) | I

A.5 Embraceable You

Embraceable You has an ABAC form.

Full sequence:

I bIIIo7 IIm(7), V(7) IIIIm(7), VI(7) | IIm(7) II%7, V(7) I
VII%7, III(7) |
VIIm, VIIm(7) #IV%7, VII(7) IIIIm, IIm(7) #Io7, Im(6) | VIIIm(7),
III(7) VIIm(7), II(7) IIm(7) V(7) |
I bIIIo7 IIm(7), V(7) IIIIm(7), VI(7) | IIm(7) II%7, V(7) I
Vm(7), I(7) |
IV VII%7, III(7) VIIm(7) #IV%7, IVm(7), bVII(7) | I IVm(7),
bVII(7), IIm(7), V(7) I

A section:

I bIIIo7 IIm(7), V(7) IIIIm(7), VI(7) | IIm(7) II%7, V(7) I

B section:

I VII%7, III(7) | VIm, VIm(7) #IV%7, VII(7) IIIIm, IIIIm(7) #Io7,
Im(6) | VIIIm(7), III(7) VIm(7), II(7) IIm(7) V(7) | I

C section:

I Vm(7), I(7) | IV VII%7, III(7) VIm(7) #IV%7, IVm(7), bVII(7)
| I IVm(7), bVII(7), IIm(7), V(7) I

A.6 The Girl from Ipanema

The Girl from Ipanema has an AABA form in which B is twice as long as A.

Full sequence:

I(M7) II(7) | IIm(7), bII(7) I(M7), bII(7) |
I(M7) II(7) | IIm(7), bII(7) I(M7) |
#I(M7) #IV(7) | #Im(7) VI(7) | IIm(7) bVII(7) | IIIIm(7),
VI(7+11) IIm(7), V(7+11) |
I(M7) II(7) | IIm(7), bII(7) I(M7), bII(7) |

A section:

I(M7) II(7) | IIm(7), bII(7) I(M7), bII(7) | I(M7)

B section:

I(M7) | #I(M7) #IV(7) | #Im(7) VI(7) | IIm(7) bVII(7) |
IIIIm(7), VI(7+11) IIm(7), V(7+11) | I(M7)

A.7 Here's That Rainy Day

Here's That Rainy Day has an ABAC form.

Full sequence:

I bIII(7) bVI(7) bII | IIm(7) V(7) I V(7), I(7) |
IVm(7) bVII(7) bIII(7) bVI(7) | IIm(7) V(7) I, VI(7) IIm(7),
V(7) |
I bIII(7) bVI(7) bII | IIm(7) V(7) I V(7), I(7) |
IV IIm(7), V(7) IIIIm(7), VIm(7) II(7), bIIIo7 | IIm(7) V(7) I,
VI(7) IIm(7), V(7)

A section:

I bIII(7) bVI(7) bII | IIm(7) V(7) I

B section:

I V(7), I(7) | IVm(7) bVII(7) bIII(7) bVI(7) | IIm(7) V(7) I,
VI(7) IIm(7), V(7) | I

C section:

I V(7), I(7) | IV IIm(7), V(7) IIIIm(7), VIm(7) II(7), bIIIo7 |
IIm(7) V(7) I

A.8 The Joint is Jumpin'

The Joint is Jumpin' has an ABCA form, but only the A section was transcribed and used as input during this study.

A section:

I, #Io7 IIm(7), V(7) I, #Io7 IIm(7), V(7) | I, I(7) IV, IVo7 I,
V(7) I

A.9 Pennies from Heaven

Pennies from Heaven was written as an ABAC form in Elliott (to appear 2008). It has been treated here as AA-BC-AB-D (with D of double length), taking full advantage of the repetition of the short A section. Sections A+B+C and A+B+D were isolated for use as input.

Full sequence:

I, IIm(7) IIIIm(7), bIIIo7 IIm(7) V(7) | I, IIm(7) IIIIm(7),
bIIIo7 IIm(7) V(7) |
Vm(7) I(7) IV IV | VIm(7) II(7) IIm(7) V(7) |
I, IIm(7) IIIIm(7), bIIIo7 IIm(7) V(7) | Vm(7) I(7) IV |
IV(7) bVII(7) I VI(7) | IIm(7) II(7), V(7) I, VI(7) IIm(7),
V(7)

A+B+C sections:

I, IIm(7) IIIIm(7), bIIIo7 IIm(7) V(7) | Vm(7) I(7) IV IV |
VIm(7) II(7) IIm(7) V(7) | I

A+B+D sections:

I, IIm(7) IIIIm(7), bIIIo7 IIm(7) V(7) | Vm(7) I(7) IV | IV(7)
bVII(7) I VI(7) | IIm(7) II(7), V(7) I

A.10 Solar

Solar is a blues and, since it is short, has not been divided into sections.

Im Im Vm(7) I(7) |
IV(M7) IV(M7) IVm(7) bVII(7) |
bIII bIIIIm(7), bVI(7) bII II%7, V(7)

A.11 Hey Joe

Hey Joe is a Rock standard made famous by Jimi Hendrix. It is included here because it is used as an example of an extended plagal cadence in Steedman (2004).

I(7) bVI bIII bVII IV I I(7)

A.12 Coker, App D., 10

Example chord sequence 10 from appendix D of Coker (1964) has been used as example input in its full form.

I(6) I(6) IV(7) IV(7) II(7) II_m(7), V(7+5) I(6) I(6) | V_m(7)
I(7) IV(6) IV(6) II(7) II(7) V(7) V(7) | I

Appendix B

Score for Bach Example

The following is the opening of the first of the Eight Short Preludes and Fugues by J. S. Bach, reproduced under license by permission of the publishers. The chord names are added and correspond to the chords used in the analysis of section 5.5.2.

The musical score is presented in two systems, each with three staves. The first system (measures 1-3) features a treble staff with a melody starting on G4, a bass staff with a bass line starting on C3, and a grand staff with a common time signature. The first measure is marked with a dynamic of *mf*. Chord labels C, C, F, D, and G are placed above the treble staff. The second system (measures 4-6) continues the piece, with chord labels E, A, D, G, C, and F#07 placed above the treble staff. The notation includes various musical symbols such as slurs, accents, and dynamic markings.

The image displays two systems of musical notation for a piece by J.S. Bach. Each system consists of three staves: a treble clef staff, a bass clef staff, and a lower bass clef staff. The first system begins at measure 7 and features a sequence of chords: Bm, Em, A, D, G, D, G, D. The second system begins at measure 10 and features a sequence of chords: G, C G, C G, D^{sus4}, D G, D G, D^{sus4}, D G, G⁷, C. The notation includes various rhythmic patterns, including sixteenth and thirty-second notes, and rests.

J.S. Bach: Organ Works Book 1: Eight Short Preludes and Fugues

Edited by John Dykes Bower and Walter Emery. (NOV010018)

©Copyright 1952 Novello & Company Limited. Copyright Renewed 1980.

All Rights Reserved. International Copyright Secured.

Reprinted by Permission.

Bibliography

- anon. (accessed Aug 2008). Blues chord progressions and variations.
http://www.jazzguitar.be/jazz_blues_chord_progressions.html.
- Bach, J. S. (1952). *Organ Works, Book 1: Eight Short Preludes and Fugues*.
Novello Publishing Ltd.
- Baldrige, J. (2007). The OpenCCG Homepage. Accessed Aug 2008.
URL <http://openccg.sourceforge.net/>
- Baldrige, J., & Kruijff, G. (2003). Multi-modal combinatory categorial grammar.
Proceedings of the 10th Annual Meeting of the EACL, Budapest, 1, 211–218.
- Beethoven (1932). *Complete Pianoforte Sonatas, Volume 1*. The Associated
Board of the Royal Schools of Music.
- Bozşahin, C., Kruijff, G. M., & White, M. (2007). Specifying grammars for
openccg: A rough guide.
URL [http://comp.ling.utexas.edu/wiki/lib/exe/fetch.php/openccg/
openccg-guide.pdf](http://comp.ling.utexas.edu/wiki/lib/exe/fetch.php/openccg/openccg-guide.pdf)
- Chomsky, N. (1957). *Syntactic Structures*. Mouton de Gruyter.
- Clark, S. (2002). Supertagging for combinatory categorial grammar. *Proceeding
of the Sixth International Workshop on Tree Adjoining Grammar and Related
Frameworks*, (pp. 101–106).
- Clark, S., Hockenmaier, J., & Steedman, M. (2002). Building deep dependency
structures with a wide-coverage ccg parser. *Proceedings of the 40th Annual
Meeting of the Association for Computational Linguistics (ACL)*, (pp. 327–
334).
- Clarke, J. (188?). *Catechism of the Elements of Harmony*. Robert Cocks & Co.

- Coker, J. (1964). *Improvising Jazz*. Simon & Schuster.
- Collins, M. (1999). Head-driven statistical models for natural language parsing. *PhD thesis, University of Pennsylvania*.
- Cork, C. (1996). *The New Guide to Harmony with LEGO Bricks*. Tadley Ewing Publications.
- Elliott, J. (to appear 2008). Observations on harmony with lego bricks. <http://www.dropback.co.uk/>.
- Haagmans, D. (1916). *Tonal Functions: Harmony, Scales and Intervals*. J. Fisher & Bro.
- Hal Leonard Corp. (2006). *The Real Book, Sixth Edition*. Hal Leonard Europe.
- Koelsch, S., Gunter, T. C., Wittfoth, M., & Sammler, D. (2005). Interaction between syntax processing in language and in music: An erp study. *Journal of Cognitive Neuroscience*, 15, 1565–1577.
- Komagata, N. (2004). A solution to the spurious ambiguity problem for practical combinatory categorial grammar parsers. *Computer Speech and Language*, 18(1), 91–103.
- Lindblom, B., & Sundberg, J. (1969). Towards a generative theory of melody. *Swedish Journal of Musicology*, 10(4), 53–86.
- Longuet-Higgins, H. C. (1979). The perception of music. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 205(1160), 307–322.
- Longuet-Higgins, H. C., & Lisle, E. R. (1989). Modelling musical cognition. *Contemporary Music Review*, 3, 15–27.
- Pachet, F. (2000). Computer analysis of jazz chord sequences: Is *Solar* a blues? In E. Miranda (Ed.) *Readings in Music and Artificial Intelligence*. Harwood Academic Publishers.
- Patel, A. D. (1998). Syntactic processing in language and music: Different cognitive operations, similar neural resources? *Music Perception*, 16(1), 27–42.
- Patel, A. D. (2003). Language, music, syntax and the brain. *Nature Neuroscience*, 6(7), 688–691.

- Pratt, G. (1984). *The Dynamics of Harmony*. Open University Press.
- Riemann, H. (1895). *Harmony Simplified, or, the Theory of the Tonal Functions of Chords*. Augener & Co. Originally *Harmonielehre*, trans. H Bemerunge.
- Schenker, H. (1906). *Harmony*. The University of Chicago Press. Trans. E. Borgese.
- Steedman, M. (1984). A generative grammar for jazz chords sequences. *Music Perception*, 2, 52–77.
- Steedman, M. (1996). *The Blues and the Abstract Truth: Music and Mental Models*, (pp. 305–318). Erlbaum.
- Steedman, M. (2000). *The Syntactic Process*. Cambridge, MA, USA: MIT Press.
- Steedman, M. (2004). Formal grammars for computational musical analysis: The blues and the abstract truth. Slides for a talk, Music Informatics and Cognition Workshop, Sonic Arts Research Centre, Queen’s University Belfast, April 2004.
- van Rossum, G. (n.d.). Python Programming Language – Official Website. Accessed Aug 2008.
URL <http://www.python.org/>