Universidad Autónoma de Madrid
Escuela politécnica superior

Double Degree in Computer Science and Mathematics

B.Sc. Thesis

ANALYSIS AND SYNTHESIS OF MUSICAL HARMONIES BASED ON FORMAL GRAMMARS

Author: María Rojo González
Advisor: Simone Santini

February 2019

Universidad Autónoma de Madrid

Escuela politécnica superior

Doble Grado en Ingeniería Informática y Matemáticas

Trabajo Fin de Grado

ANÁLISIS Y SÍNTESIS DE ARMONÍAS MUSICALES BASADAS EN GRAMÁTICAS FORMALES

Autor: María Rojo González
Tutor: Simone Santini

Febrero 2019

ANALYSIS AND SYNTHESIS OF MUSICAL HARMONIES BASED ON FORMAL GRAMMARS

Author: María Rojo González
Advisor: Simone Santini

Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Febrero 2019

Abstract

Music is one of the most fundamental and universal forms of expression of humanity, being part of the human society since its origins. As any artistic manifestation, it is a cultural product and its evolution and use in all kind of scenarios show us how important it is for the humanity. The relation between music and mathematics has been a close one, investigated since the Pythagoreans, and one of the main products of this relation is tonal music, considered one of the greatest contributions of western civilization. So not only is music one of the forms of expression that most directly speak to the emotions, it is also the one whose foundations are more firmly mathematical, making music an ideal testing ground for analyzing the formalization of emotionally charged forms of expression, and to explore the limits of such formalization.

Music is not subject to simple deterministic rules, but there is an obvious structure underneath it. Investigating and formalizing such structure is important to better understand music and even to develop new ways of composition using computers as a synergistic partner or as an independent composer. Thus, in this work, we consider the usefulness and the limits of applicability of formal grammars. The use of grammars can give important insights into the harmonic structure of a piece of music, and a lot of researchers have developed grammars and applied them to different styles of music.

In musical theory there are harmonic concepts that are not well-defined in a mathematical sense. In this work we considered several of these concepts and see how, and whether, grammars can be modified to accommodate them. However, the ambiguous nature of music limits the possibility of formal analysis of harmonic sequences: using the grammars we could not precisely define (and, consequently, recognize) some musical concepts, such as modulation (the change of key within a musical piece). Trying to formalize modulation leads to unavoidable ambiguities in the grammar. Consequently we tried using other methods to formalize these concepts, and we successfully could determine modulations using a numerical cost-based method. Then we used grammars to analyze separately the non-modulating segments in which our algorithm has divided the music. The combination of different methods and technologies developed in this work appears to be a very bright perspective for analyzing musical pieces.

Key words

Formal grammars, numerical cost-based methods, automated composition, harmonic analysis, tonal music.

Resumen

La música es una de las formas de expresión más fundamentales y universales de la humanidad, formando parte de las sociedades humanas desde sus orígenes. Como toda manifestación artística, es un producto cultural, y su evolución y uso en todo tipo de situaciones nos muestra lo importante que es para el ser humano. Desde los tiempos de la Escuela pitagórica ya se empezó a investigar la relación entre la música y las matemáticas y unos de los principales productos de esta relación es sin duda la música tonal, considerada una de las mayores contribuciones de la civilización occidental. Por tanto, la música no es sólo una de las formas de expresión más cercana a nuestros sentimientos y emociones, también tiene una sólida base matemática. Esto permite que sea un campo ideal para analizar la formalización de formas de expresión con carga emocional, y para explorar los límites de dicha formalización.

La música no está sujeta a un conjunto de reglas simples y deterministas, sin embargo, es fácil darse cuenta que está basada en estructuras. Es importante investigar y formalizar dichas estructuras, ya que nos permite entender mejor la música e incluso desarrollar nuevas formas y métodos de composición. Por tanto, en este trabajo consideramos la utilidad y los límites de las gramáticas formales para la formalización de la música. El uso de gramáticas nos ofrece diferentes perspectivas de la estructura armónica de una pieza musical. Por ello, muchos investigadores y musicólogos han desarrollado gramáticas y las han aplicado a diferentes estilos musicales.

En la teoría musical hay algunos conceptos de la armonía que no están bien definidos desde un punto de vista matemático. En este proyecto hemos considerado algunos de estos conceptos y hemos visto si la gramática podría modificarse para representarlos y cómo. Sin embargo, la ambigüedad inherente a la música limita la formalización de secuencias armónicas: no hemos podido definir algunos conceptos musicales usando las gramáticas, como por ejemplo, la modulación (el cambio de tonalidad en una pieza musical). La formalización de la modulación conlleva la introducción de ambigüedades en la gramática. En consecuencia, hemos intentado usar otros métodos para formalizar estos conceptos, y hemos conseguido detectar modulaciones con éxito usando un método numérico basado en costes. A continuación, hemos usando las gramáticas para analizar por separado las secuencias no modulantes en los que nuestro algoritmo ha dividido el fragmento musical. Por estos motivos, consideramos que la combinación de distintos métodos y tecnologías presenta un futuro prometedor para el campo del análisis musical automático.

Palabras Clave

Gramáticas formales, métodos numéricos basados en costes, composición automática, análisis armónico, música tonal.

Special thanks

I would like to thank my advisor Simone, who has believed in me and has helped me with this non-conventional project since the beginning. Special thanks to my friend José Alberto and my brother Javier for all the musical advice and to my friends Iñaki and Mario. Without them, the project would not be what it is today.

Also, a big thank you to all my friends that have been supporting me through this stage of my life with their encouraging words. It has been tough, but you were right: I could do it.

Contents

List of Figures

List of Tables

1
Introduction

1.1 Motivation

Music is one of the most fundamental and universal forms of expression of humanity. Even before the migrations of the first human groups from Africa (over 50.000 years ago), people started creating sounds and organizing them into rhythmical structures using prehistoric musical instruments. It is an universal cultural manifestation, born along with human beings, essential for and at the same time product of the human cultural evolution. Music has been used to communicate, to provoke emotions, for religious and mating rituals and for group work.

Since the ancient Greeks began a systematic study of mathematics, in the VI century B.C., the relation between music and mathematics has been a close one. The Pythagoreans were the first to investigate the expression of musical scales in terms of numerical ratios, discovering that sounds with a pitch ratio of 1:2 are perceived as the same note in different octaves, and that the ratio of 3:2 describes a pure perfect fifth, pleasing to the ear. The relation between music and mathematics was further investigated in medieval universities, where music was taught along with arithmetic and geometry as part of the mathematical curriculum. In the Middle Ages, cultivated music in Europe was mostly developed in the Roman Catholic Church, used for liturgical rites.

One of the main products of the relation between music and mathematics is without a doubt *tonal music*. It was created between the XVI and the XVII century and it is considered one of the greatest contributions of western civilization. *The Art of Fugue*, by Johann Sebastian Bach, exhibits the theoretical principles of counterpoint and tonal music and is considered one of the masterpieces of the entire history of music. Almost all modern music, from opera to pop songs, is tonal, and even the forms that explore and try to break its boundaries, such as modern jazz, have to come to terms with tonality.

This is what makes music so unique: not only it is intrinsic to human beings, being one of the forms of expression that most directly speaks to the emotions, but it is also the one whose foundations are more firmly mathematical. As computer scientists and mathematicians, this characteristics gives us the opportunity to use music as the ideal testing ground for analyzing the formalization of emotionally charged forms of expression, and to explore the limits of such formalization.

1.2 Scope and objectives

Music is the combination of harmony, melody and rhythm; in this project, we are going to limit our focus to the harmonic part of the music. Musical composition is a creative activity, and it is not subject to simple, deterministic rules. If it were, by now, after 400 years of tonal music, we would probably have run out of new songs. Still, harmony is not arbitrary: there is an obvious structure in harmonic sequences, and this has soon awakened the appetite of computer scientists for possible models of formalization of harmonies. A lot of this work uses techniques similar to those used in the analysis of natural language, because like language, music is generative: a relatively small set of rules can be used to generate infinite sentences of arbitrary complexity.

Thus, in this work, we consider the usefulness and the limits of applicability of formal grammars. The use of grammars can give important insights into the harmonic structure of a piece of music, and a lot of researchers have developed grammars and applied them to different styles of music. We shall consider some of their work in the next chapter.

One of the challenges in the application of formal grammars to musical theory is that the rules are not quite as strict as they are in language. The existence of various tonalities creates an inherent ambiguity in the interpretation of the harmonic structure that cannot be resolved at the syntactic level, as we shall discuss in detail in the following chapters. There is a linguistic parallel that may help understand the problem. Consider the sentence:

I vitelli dei romani sono belli

If we read it as an Italian sentence, it means: "The cows of the romans are beautiful", but if we read the same sequence of symbols as a Latin sentence, its meaning is: "Go, Vitellius, at the war sound of the Roman god". In language, examples like this are extremely rare, and can be dismissed as quirks. Not so in music, where tonality and structure are intertwined and can easily create ambiguity.

In fact, in musical theory there are harmonic concepts that are not well-defined in a mathematical sense, thus they cannot be precisely defined using grammars. When one tries to apply formal grammars to these concepts the result is often either an ambiguous grammar or a grammar that generates undesired structures, which do not correspond to any construct of musical theory.

In this work we shall consider several of these concepts and see how, and whether, grammars can be modified to accommodate them. In particular, we shall pay special attention to modulation, that is, the change of key within a musical piece. We shall see that trying to formalize modulation leads to unavoidable ambiguities in the grammar. To solve this problem, we shall develop a new, hybrid approach: we develop a numerical, cost-based approach to detect modulations and then use grammars to analyze separately the non-modulating segments in which our algorithm has divided the music.

1.3 Structure of the document

This work is organized as follows:

The first chapter provides a brief introduction to the motivation, scope and objectives of this work paying special attention to the relation between music and languages. The second chapter presents the context of the project: the interest on formalizing music and a brief, partial history of the automatic composition research. Furthermore, the formal grammar that we take as the starting point of this work is presented and explained.

In the third chapter we discuss some mathematically ill-defined harmonic concepts and whether the grammar can be modified to accommodate them or not. At the end of the chapter some extensions and limitations of the formalization of harmony using the grammar are presented.

The fourth chapter considers the harmonic concept of modulation. A numerical approach to detecting modulations is exposed and we present some examples which explain its behavior.

In the fifth chapter we show how the formal grammar and the modulation approach work together. We analyze two famous fragments of classical musical pieces using first the modulation detection algorithm and then the formal grammar to complete the analysis. The sixth chapter finished the work with some conclusions and future work lines.

Additionally, there are two annexes that further explain some music concepts needed to completely understand the project, and that report all the cost tables that we used in our algorithm.

2
State of the art

2.1 Music

Music is as old as humankind. As any artistic manifestation it is a cultural product and its evolution and use in all kind of scenarios show us how important it is for the humanity. Innumerable music styles have been created and developed in each of the different existing cultures. The music is different depending on period and location. Even within the same culture and in the same period of time we can find different musical expressions; for instance, nowadays in Europe, jazz, pop music and classical music are different musical styles that live at the same time.

The differences between musical styles are of harmonic, melodic or rhythmic nature, or any combination of the three. We can distinguish between tonal and modal music, there are a variety of musical scales and even the sounds (the musical notes) can vary from one culture to another: for example, arab music use quarter tones, while in western music the fundamental unit is the semitone (two quarter tones).

Over the years, a lot of research has been done with the objective of formalizing music [2]. These studies usually focus in a particular music style [19] due to the big differences between music styles, which makes a general formalization an almost impossible task. In this project we are going to focus on the tonal musical harmony of the XVII, XVIII and early XIX century, whose theoretical basis is studied in [16]. A brief explanation of some basic harmonic theory can be found in Annex A.

The connection between the computer and music is even older than the computer itself. In the early XIX century, when Charles Babbage designed his analytical engine [4], some thought was already given for possible applications of computers to music. Lady Ada Lovelace wrote in her notes:

> "[The Analytical Engine] might act upon other things besides number, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine... Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent." [9] [13]

Interest on automated composition and the integration of computers in music has since then naturally evolved with new research and technologies [2] [14].

2.2 Automated composition

Today, computers already have plenty of applications in music related work: from sound synthesis, which includes Fourier synthesis, frequency modulation (FM) syntheses and VOSIM systems, to signal processing. There has also been a lot of research in the development of both hardware and software for sound synthesizing to describe instrumental timbres [8]. But, despite having less visible results, computer music composition has always had a great theoretical interest, being a heavily researched topic.

The interest in the study of musical structure has a long history in computing, dating at least from the 1950s [2]. In 1956 Klein and Bolitho created "Push Button Bertha" using a computer called DATATRON, which composed melodies using random sampling and testing. The same year, using Markov chains, Hiller and Isaacson used the ILLIAC computer to compose the "Illiac Suite". Another collaborator of Hiller's, Baker, conceived a composing utility called MUSICOMP and they used it to create "Computer Cantata" in 1962. Automated compositional procedures were used in Europe in the sixties: Barbaud documented his procedures in "Initiation à la composition musicale automatique"; Xenakis composed with his "stochastic music program" "Pithoprakta" and "Achorripsis"; and Koenig created "Übung fur Klavier" using statistical methods. Other technologies employed in music composition were segmented line graphs and hierarchic procedures, by Tenney in his "Stochastic String Quarted" of 1963; and graphic methods, by Myhill in "Scherzo a Tre Voce" of 1965. Other composers, as DeLio, merged manual (to compose the form) with automatic composition (to compose the details). Lastly, Gill used backtracking tools to compose "Variations on a Theme by Berg" in 1963.

Papadopoulos and Wiggins [14] categorized and reviewed different AI methods that have been used for music: mathematical models, such as stochastic processes and Markov chains, knowledge based systems, grammars, evolutionary methods such as genetic algorithms, system which learn such as artificial neural networks and machine learning, and hybrid systems, which use a combination of techniques.

As we have mentioned before, music is the combination of harmony, melody and rhythm. Automatic composition approaches normally focus in one of the three elements; usually harmony or melody. So we can differentiate the study of harmonic composition from the study of melodic composition. For example, Ames [3] used Markov chains for the automated composition of melodies. He divided the music elements of a melody (notes and rests) in a set of characteristics: duration, articulation, register and chromatic degree. Each one has a matrix of transition probabilities that defines a Markov chain. The composition consists of a loop, where in each iteration all the Markov chains determine together the next element. The main advantage of Markov's models is that they can be adapted to any system of probabilities. In fact, this approach could be used in two different ways: "composing" the probabilities of the matrices or deriving the probabilities from the analysis of existing music material. On the other hand, the main disadvantage is that Markov chains cannot provide structure nor hierarchy to the music composition: it is just a linear generator of music elements. Ames proposed a blend of approaches, using Markov chains along with other methods.

Other approaches have been used as well: Gillick, Tang and Keller [7] generated jazz melodies using machine learning and Ortega, Sánchez and Alfonseca [6] used grammatical evolution to create melodies: the melodies are created by a grammar and then genetic algorithms evaluate the quality of the results.

The similarities between musical syntax and natural language syntax make Chomsky formal grammars a good candidate for formalization and for automatic composition. In addition to that, some authors use the computer as a synergistic partner for the human composer instead of an independent composer. For instance Keller and Morrison [10] used probabilistic context-free grammars to generate jazz melodies automatically in real-time. They used this approach to implement a software tool, "Impro-Visor" for assisting musicians in creating jazz solos. This approach generates melodies based on a previous existing harmony and they manifested having problems with the rhythm of the melody, an observation that confirms the need of separating the three elements of the music. McCormack [12] adapted grammars based on L-systems into a system for composition of melodies to help composers. To do this, he merged different methods as L-systems, stochastic grammars, parametric numerical parameters and hierarchical and context sensitive grammars. Perchy and Sarria [15] used stochastic context-free grammars for generating melodies with the purpose of aiding musicians in the compositional work.

Lastly, Holtzman [8] creates a Generative Grammar Definition Language (GGDL) compiler to describe grammars for automated composition. He used it to formalize compositional languages from past research.

All these studies focus on the automated composition of melodies. Work has also been done on the analysis and synthesis of harmonies. As we mentioned before there is an analogy between music and language, so there are rules that determine the musical structure, analogous to rules of the grammar of languages. This, along with the recursive character of the music harmony, makes generative grammars a proper formalism for describing it. Steedman [19] created a generative grammar that describes the harmony of the jazz 12-bar blues. Rohrmeier [17] [18] made a more generic generative grammar for tonal music. Steedman's grammar is context sensitive whereas Rohrmeier's grammar is context-free.

To make the rules of these grammars it is necessary to characterize them first, in other words, we need to analyze the harmonic structure of a music style first in order to describe it with grammar rules. Once we have the rules, we could use them in a generative grammar to compose new music. As we said when we talked about music, there are numerous music styles and they are harmonically different. So we need different rules to describe each of them. For this reason, a general formalization does not make much sense, so musicians and scientists focus on a music style in particular.

2.3 A grammar for harmonic analysis

In this work we have taken as a basis a specific grammar of music harmony, one created by Martin Rohrmeier [18]. The reasons for this choice are manifold. Firstly, unlike other grammars, it is not related to a specific genre, but allows the modeling of arbitrary harmonic structures; in particular, it can be applied to the analysis of baroque, romantic and modern music, which is the focus of this work. Secondly, Rohrmeier's grammar has shown great potential for analysis as well as for music generation [18].

The grammar defined in [18] represents the hierarchical properties of harmony. It applies to a large set of tonal harmonic progressions but it does not cover all tonal harmony, a task which would require a large number of style-specific rules. In this work, we shall use this grammar as a starting point. In order to make this work self-contained, in this section we shall spend some time analyzing this grammar.

The grammar is based in two principles: (1) each chord in a chord sequence depends on its preceding or following chord or chord group, and (2) chords are classified into functional categories depending on their tonal function. Consequently, the grammar is based in the tonal functions of the chords (tonic, dominant, subdominant) rather than on the chords themselves. This allows us to separate the structure of a phrase, which is independent of tonality, from the actual chords. Tonal functions are mapped to specific chords by a tonality. For example, the subdominant (IV) is mapped to F if the tonality is C, to C if it is G, to $E\flat$ if it is B♭, etc.

The grammar is hierarchical, structured on four levels: phrase, functional, scale degree and surface. The productions are correspondingly hierarchical: each production transforms a non-terminal into a sequence of non-terminals at the same level or at lower levels.

- The non-terminals of the phrase level:

$$\mathbb{P} = \{piece, P\}$$

- The non-terminals of the functional level:

$$\mathbb{R} = \{TR, SR, DR\}$$
$$\mathbb{F} = \{t, s, d, tp, sp, dp, tcp\}$$

  At this level, the harmony is composed of a collection of regions with a tonic (TR), dominant (DR) or subdominant (SR) value. These generate various tonic (t, tp, tcp), dominant (d, dp) or subdominant components.

- The non-terminals of the scale degree level:

$$\mathbb{S} = \{I, II, ..., VII, V^I, V^{II}, ..., VII^I, VII^{II}, ...\}$$

  Each symbol here corresponds to a single chord, expressed in reference to the tonality of a region.

- The terminals, which are at the surface level and represent each existing chord:

$$\mathbb{O} = \{C, Cm, C^0, ...\}$$

Each non-terminal symbol implicitly carries the key property, i.e., each non-terminal symbol defined above has an attribute: a key in $\mathbb{K}$.

$$\mathbb{K} = \{C, Cm, C\sharp, C\sharp m, D\flat, D\flat m, ...\}$$

Where $\mathbb{K}$ is the set of all the keys of the tonal harmony.

The start symbol is: *piece*. The rules of the grammar are, as we already mentioned, defined for each level, and form a hierarchy.

Phrase level. A piece is formed by phrases and the head of each phrase is a tonic region (TR).

$$piece_{key=x\in\mathbb{K}} \longrightarrow P^+ \quad (2.1)$$

$$P \longrightarrow TR \quad (2.2)$$

Functional level. This level defines the relationships between functions and keys on an abstract level. The rules are divided in different sets with different objectives: to expand functional sequences, to substitute functional elements and to modulate or change the key.

- Functional expansion rules:

$$TR \longrightarrow DR \quad t \quad | \quad TR \quad DR \quad | \quad TR \quad TR \quad | \quad t \quad (2.3)$$

$$DR \longrightarrow SR \quad d \quad | \quad DR \quad DR \quad | \quad d \quad (2.4)$$

$$SR \longrightarrow SR \quad SR \quad | \quad s \quad (2.5)$$

- Substitution rules:

$$t \longrightarrow tp \quad | \quad tcp \quad (2.6)$$

$$s \longrightarrow sp \quad (2.7)$$

$$d \longrightarrow dp \quad (2.8)$$

- Modulation rules:

$$X_{key=y} \longrightarrow TR_{key=\psi(X,y)} \qquad \text{for any } X \in \mathbb{F} \text{ and } y \in \mathbb{K} \quad (2.9)$$

$$X_{key=y\ maj/min} \longrightarrow X_{key=y\ min/maj} \qquad \text{for any } X \in \mathbb{F} \text{ and } y \in \mathbb{K} \quad (2.10)$$

  Where $\psi(f, k) \in \mathbb{K}$ is the function that determines the new key.

Scale degree level. At this level the functional symbols are transformed into scale degrees symbols, which are a key-independent representation of the harmonic functions (viz., each chord is represented by its tonal function in the key).

- Secondary dominant rules

$$X \longrightarrow D(X) \quad X \qquad \text{for any } X \in \mathbb{S} \quad (2.11)$$

$$X \longrightarrow \Delta(X) \quad X \qquad \text{for any } X \in \mathbb{S} \quad (2.12)$$

$$D(X) \longrightarrow \begin{cases} V^{VI^X} \quad | \quad VII^{VI^X} & \text{if } X \text{ refers to a diminished triad} \\ V^X \quad | \quad VII^X & \text{otherwise} \end{cases} \quad (2.13)$$

  Where $D(X)$ is the dominant of X (scale degree of a perfect fifth above X) and $\Delta(X)$ is the scale degree of a fifth above X.

- Function-scale degree interface

$$t \longrightarrow I \quad | \quad I \quad IV \quad I \tag{2.14}$$

$$s \longrightarrow IV \tag{2.15}$$

$$d \longrightarrow V \quad | \quad VII \tag{2.16}$$

$$tp \longrightarrow \begin{cases} VI & \text{if key is major} \\ III & \text{if key is minor} \end{cases} \tag{2.17}$$

$$dp \longrightarrow VII \quad \text{if key is minor} \tag{2.18}$$

$$sp \longrightarrow \begin{cases} II & \text{if key is major} \\ VI \quad | \quad \flat II & \text{if key is minor} \end{cases} \tag{2.19}$$

$$tcp \longrightarrow \begin{cases} III & \text{if key is major} \\ VI & \text{if key is minor} \end{cases} \tag{2.20}$$

Surface level. At this level, scale degrees are transformed into chords, given the key. There are 24 keys (12 major and 12 minor) and 7 degrees in each one, so there are 24x7 = 168 rules of the type:

$$I_{key=C} \longrightarrow C$$
$$VI_{key=G} \longrightarrow Em$$
$$I_{key=Dm} \longrightarrow Dm$$
$$V_{key=Am} \longrightarrow E$$
$$...$$

In addition, the following rule allows any chord to be repeated.

$$X \longrightarrow X \quad X \qquad \text{for any } X \in \mathbb{O} \tag{2.21}$$

The complete grammar is summarized in Figure 2.1.

Non-terminal symbols:

$$\mathbb{P} = \{piece, P\}$$
$$\mathbb{R} = \{TR, SR, DR\}$$
$$\mathbb{F} = \{t, s, d, tp, sp, dp, tcp\}$$
$$\mathbb{S} = \{I, II, ..., VII, V^{I}, V^{II}, ..., VII^{I}, VII^{II}, ...\}$$

Terminal symbols: $\mathbb{O} = \{C, Cm, C^0, ...\}$
Symbols' attributes: $\mathbb{K} = \{C, Cm, C\sharp, C\sharp m, D\flat, D\flat m, ...\}$
Start symbol: *piece*
Productions:

$$
\begin{aligned}
piece_{key=x\in\mathbb{K}} &\longrightarrow P^{+} \\
P &\longrightarrow TR \\
TR &\longrightarrow DR \quad t \quad | \quad TR \quad DR \quad | \quad TR \quad TR \quad | \quad t \\
DR &\longrightarrow SR \quad d \quad | \quad DR \quad DR \quad | \quad d \\
SR &\longrightarrow SR \quad SR \quad | \quad s \\
t &\longrightarrow tp \quad | \quad tcp \\
s &\longrightarrow sp \\
d &\longrightarrow dp \\
X_{key=y} &\longrightarrow TR_{key=\psi(X,y)} \qquad \text{for any } X \in \mathbb{F} \text{ and } y \in \mathbb{K} \\
X_{key=y\ maj/min} &\longrightarrow X_{key=y\ min/maj} \qquad \text{for any } X \in \mathbb{F} \text{ and } y \in \mathbb{K} \\
X &\longrightarrow D(X) \quad X \qquad \text{for any } X \in \mathbb{S} \\
X &\longrightarrow \Delta(X) \quad X \qquad \text{for any } X \in \mathbb{S} \\
D(X) &\longrightarrow \begin{cases} V^{VI^{X}} \quad | \quad VII^{VI^{X}} & \text{if } X \text{ refers to a diminished triad} \\ V^{X} \quad | \quad VII^{X} & \text{otherwise} \end{cases} \\
t &\longrightarrow I \quad | \quad I \quad IV \quad I \\
s &\longrightarrow IV \\
d &\longrightarrow V \quad | \quad VII \\
tp &\longrightarrow \begin{cases} VI & \text{if key is major} \\ III & \text{if key is minor} \end{cases} \\
dp &\longrightarrow VII \quad \text{if key is minor} \\
sp &\longrightarrow \begin{cases} II & \text{if key is major} \\ VI \quad | \quad \flat II & \text{if key is minor} \end{cases} \\
tcp &\longrightarrow \begin{cases} III & \text{if key is major} \\ VI & \text{if key is minor} \end{cases} \\
X &\longrightarrow X \quad X \qquad \text{for any } X \in \mathbb{O}
\end{aligned}
$$

Transformational surface level rules

Figure 2.1: Rohrmeier's grammar [18].

3

Analysis and modification of the grammar

In this section, we present an analysis of the grammar that we introduced in the previous chapter and propose some modifications. Our initial objective was to see whether it was feasible to implement it as an LR grammar, as this would entail a simple and efficient parsing algorithm [1]. During this analysis we observed that the grammar is ambiguous, meaning that a sequence of chords might have several different parse trees. When we studied the ambiguities of the grammar, we observed that they are of different natures. Some of them derive from the way the grammar is defined, and they could be eliminated through formal transformations of the grammar. Other ambiguities are inherent to the structure of music and they are related to the fact that each chord can be interpreted in any tonality and in each one has a different harmonic function [16]. For example, the sequence

$$G\ C$$

can be interpreted as a dominant-tonic progression in the tonality of C ($V\quad I$), a tonic chord followed by a subdominant one in G ($I\quad IV$), a subdominant chord followed by a dominant one in F ($II\quad V$), and so on. Because of this, modulation (changes from one tonality to another, rules (2.9) and (2.10)) introduces an ambiguity that cannot be eliminated solely with formal means. In this chapter we analyze these ambiguities and their nature.

3.1 Formal ambiguities

In this section we analyze the formal ambiguities encountered in the grammar presented in [18]. In order to do this in a simpler setting, in this section we disregard the modulation, the secondary dominant and the surface level rules, so that the terminals of the grammar are: $\{I, II, III, IV, V, VI, VII, \flat II\}$ and the non-terminals are: $\{piece, P, TR, SR, DR, t, s, d, tp, sp, dp, tcp\}$. We also assume that the key is major. Although in the original grammar the chord $\flat II$ is considered exclusive to the minor mode, it also can be used in the major mode [16], so we are going to include it here.

A simple example will show that the grammar is ambiguous. Consider the chords $I\quad IV\quad V\quad I$ (a very common progression in both classical and popular music). This sequence can be parsed in two different ways as seen in Figure 3.1.

It must be noted that, from a musical point of view, these two structures are very different. In the tree on the left, the first three chords form a phrase ($I\quad IV\quad V$ form a half-cadence), which is separated from the last chord (I). On the other hand, in the tree on the right the last three chords ($IV\quad V\quad I$) form a perfect cadence which is prepared by the first chord (I). In fact, the tree on the left represents two phrases separated by a cadence and the tree on the right is a phrase that ends in a cadence.

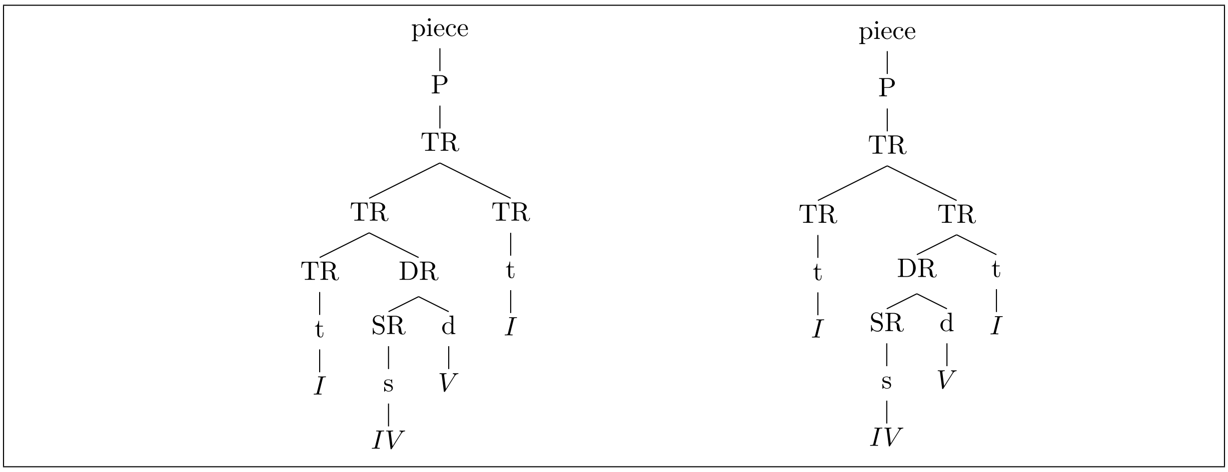


Figure 3.1: Two different parse trees of the chord sequence $I \quad IV \quad V \quad I$ that show the ambiguity of the grammar.

Consider the top-level rules:

$$\begin{aligned} piece &\longrightarrow piece \quad P \quad | \quad P \\ P &\longrightarrow TR \\ TR &\longrightarrow TR \quad TR \end{aligned}$$

These three rules generate an ambiguity because one could add TR terms using the first rule or the third rule. To avoid this problem we eliminate the non-terminal P and we substitute the first two rules with the following one:

$$piece \longrightarrow TR$$

With this modification, the repetition of TR is done always with the third rule. The next rules also produce ambiguities, as shown in the Figure 3.1:

$$TR \longrightarrow DR \quad t \quad | \quad TR \quad DR \quad | \quad TR \quad TR \quad | \quad t$$

To eliminate this ambiguity, we add a new non-terminal symbol, CTR, and replace the rule above with the two rules:

$$\begin{aligned} TR &\longrightarrow CTR \quad | \quad CTR \quad TR \quad | \quad CTR \quad DR \\ CTR &\longrightarrow DR \quad t \quad | \quad t \end{aligned}$$

Analogously the following rules produce ambiguities:

$$\begin{aligned} DR &\longrightarrow SR \quad d \quad | \quad DR \quad DR \quad | \quad d \\ SR &\longrightarrow SR \quad SR \quad | \quad s \end{aligned}$$

To eliminate them, we add the non-terminal symbols CDR and CSR to the grammar and replace the two rules above with the following ones:

$$\begin{aligned} DR &\longrightarrow CDR \quad | \quad CDR \quad DR \\ CDR &\longrightarrow SR \quad d \quad | \quad d \\ SR &\longrightarrow CSR \quad | \quad CSR \quad SR \\ CSR &\longrightarrow s \end{aligned}$$

The following function-scale degree interface rules generate ambiguities as well:

$$\begin{aligned} d &\longrightarrow V \quad | \quad VII \\ dp &\longrightarrow VII \end{aligned}$$

The ambiguity generated by these rules is trivial, as the symbol VII could be obtained by $d \to VII$ or $d \to dp \to VII$. To solve it we decide to eliminate the rule $d \longrightarrow VII$.
Lastly, the rules

$$t \longrightarrow I \quad | \quad I \quad IV \quad I$$

make the grammar nondeterministic. In other words and input symbol (I) does not uniquely determine a production (it could be $t \longrightarrow I$ or $t \longrightarrow I \quad IV \quad I$). In this case, rather than replacing the rules, we decided to use the GLR (Generalized LR) parser of Bison to handle this grammar. In conclusion, we obtain the grammar of Figure 3.2.

Non-terminal symbols: $\{piece, TR, SR, DR, CTR, CSR, CDR, t, s, d, tp, sp, dp, tcp\}$
Terminal symbols: $\{I, II, III, IV, V, VI, VII, \flat II\}$
Start symbol: *piece*
Production rules:

$$\begin{aligned}
piece &\longrightarrow TR \\
TR &\longrightarrow CTR \quad | \quad CTR \quad TR \quad | \quad CTR \quad DR \\
CTR &\longrightarrow DR \quad t \quad | \quad t \\
DR &\longrightarrow CDR \quad | \quad CDR \quad DR \\
CDR &\longrightarrow SR \quad d \quad | \quad d \\
SR &\longrightarrow CSR \quad | \quad CSR \quad SR \\
CSR &\longrightarrow s \\
t &\longrightarrow tp \quad | \quad tcp \\
s &\longrightarrow sp \\
d &\longrightarrow dp \\
t &\longrightarrow I \quad | \quad I \quad IV \quad I \\
s &\longrightarrow IV \\
d &\longrightarrow V \\
tp &\longrightarrow VI \\
dp &\longrightarrow VII \\
sp &\longrightarrow II \quad | \quad \flat II \\
tcp &\longrightarrow III
\end{aligned}$$

Figure 3.2: Modified grammar created from the modifications of the grammar on [18].

3.2 Secondary dominants and descending fifths sequences

In the previous section, we have modified the grammar ignoring secondary dominants and, consequently, the descending fifths sequences related to them. In this section, we analyze these issues, with special reference to the ambiguities that they generate. We take as our starting point the modified grammar without ambiguities from the last section, represented in Figure 3.2.

3.2.1 Secondary dominants

The productions (2.11) and (2.13) of the grammar, introduced in Section 2.3 (p. 6), define the secondary dominants. They can generate secondary dominants at arbitrary depth, which, from a musical point of view, does not make much sense. Therefore, we consider one level of depth: a secondary dominant is associated with one of the basic degrees of the key: I, II, III, IV, V, VI or VII. More depths (for instance, a secondary dominant of V^{II}) would lead us very far from the key and are not very common in music.

So, to add this rule to our modified grammar, we need to add the terminals

$$\{V^{II}, V^{III}, V^{IV}, V^{V}, V^{VI}, V^{VII}, VII^{II}, VII^{III}, VII^{IV}, VII^{V}, VII^{VI}, VII^{VII}\}$$

In production (2.13) there is a distinction depending on whether the chord is a diminished triad or not. In major keys, the only degree that is diminished is VII. The diminished triads do not determine any particular key, so they are not the tonic chord of any key and therefore we cannot define the dominant of these chords. Consequently, we are not going to consider secondary dominant of diminished triads (VII) or of the tonic I (because its dominant is V).

In production (2.11), X stands for any of the non-terminals $\{I, ..., VII, V^{I}, V^{II}, ..., VII^{I}, VII^{II}, ...\}$ of the original grammar, which are terminals in our modified grammar. As there cannot be terminal symbols in the left-hand side of productions, we add the following non-terminals:

$$\{dI, dII, dIII, dIV, dV, dVI, dVII\}$$

and the following productions:

$$\begin{aligned} dI &\longrightarrow I \\ dII &\longrightarrow II \\ dIII &\longrightarrow III \\ dIV &\longrightarrow IV \\ dV &\longrightarrow V \\ dVI &\longrightarrow VI \\ dVII &\longrightarrow VII \end{aligned}$$

The secondary dominant productions we are considering are:

$$\begin{aligned} dII &\longrightarrow V^{II} \quad II \quad | \quad VII^{II} \quad II \\ dIII &\longrightarrow V^{III} \quad III \quad | \quad VII^{III} \quad III \\ dIV &\longrightarrow V^{IV} \quad IV \quad | \quad VII^{IV} \quad IV \\ dV &\longrightarrow V^{V} \quad V \quad | \quad VII^{V} \quad V \\ dVI &\longrightarrow V^{VI} \quad VI \quad | \quad VII^{VI} \quad VI \end{aligned}$$

Since in minor mode there is a VII chord that is not diminished, the dominant rule of the VII chord is added:

$$dVII \longrightarrow V^{VII} \quad VII \quad | \quad VII^{VII} \quad VII$$

As we added some new terminals and non-terminals we need to modify some productions the following way:

$$\begin{aligned} t &\longrightarrow dI \quad | \quad dI \quad dIV \quad dI \\ s &\longrightarrow dIV \\ d &\longrightarrow dV \\ tp &\longrightarrow dVI \\ dp &\longrightarrow dVII \\ sp &\longrightarrow dII \quad | \quad \flat II \\ tcp &\longrightarrow dIII \end{aligned}$$

3.2.2 Irregular resolutions

The previous productions consider the regular resolutions of secondary dominants (i.e., when the secondary dominant is followed by its tonic). Even though this is the most common case, the secondary dominants could be followed by other chords, generating what is known as an *irregular resolution*.

Our first approach was to try to generalize the secondary dominant resolutions, introducing functional level symbols in the productions. For instance, the production

$$dV \longrightarrow V^V \quad V$$

which is a regular resolution of the dominant of the V in the V, could be generalized as

$$dV \longrightarrow V^V \quad TR$$

But this introduces ambiguities as it breaks the hierarchy of the grammar: the chords that follow the secondary dominant could be in the secondary dominant level or in a higher level. For example, using the production above, there are two ways to analyze the sequence $I \quad IV \quad V^V \quad I \quad V$, as we can see in Figure 3.3.

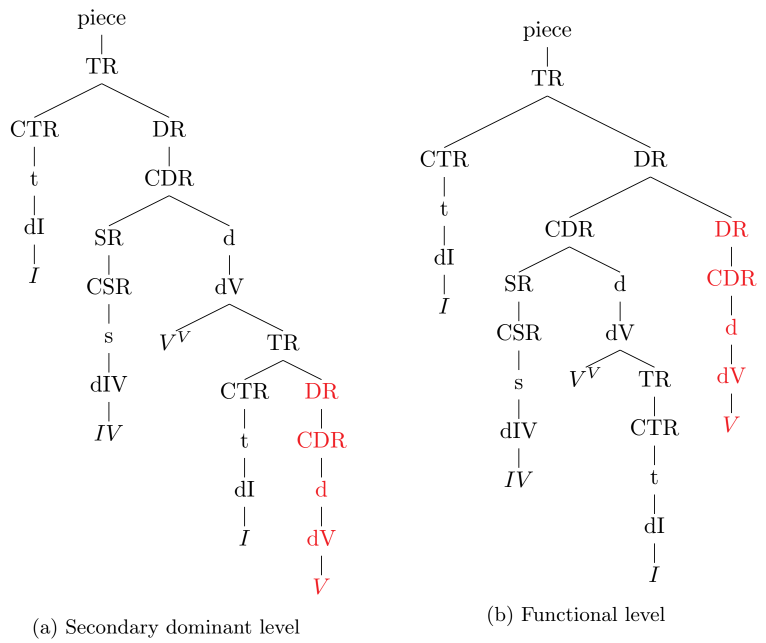


Figure 3.3: Introducing more general secondary dominant productions generates ambiguities as the two different parse trees of the sequence $I \quad IV \quad V^V \quad I \quad V$ show us.

So, we decided to expand the grammar with individual cases of irregular resolutions, instead of generalizing the grammar (this will be discussed in the extensions and limitations section 3.6). Thus, we add to the grammar the productions of Table 3.1. They are the rules for each of the possible irregular resolutions explained in [16].

Irregular resolution	**Grammar production**
V^{VI} - IV	$dIV \longrightarrow V^{VI} \quad IV$
V^{III} - I	$dI \longrightarrow V^{III} \quad I$
V^V - III	$dIII \longrightarrow V^V \quad III$
V^V - I	$dI \longrightarrow V^V \quad I$
V^{II} - V	$dV \longrightarrow V^{II} \quad V$
V^{IV} - V^{II} - II	$dII \longrightarrow V^{IV} \quad V^{II} \quad II$
V^V - V^{IV} - IV	$dIV \longrightarrow V^V \quad V^{IV} \quad IV$

Table 3.1: Secondary dominant irregular resolution productions.

3.2.3 Descending fifths sequences

The production (2.12) of page 6 contains the idea of descending fifths sequences or progressions. The expression "descending fifths" refers to a progression where each chord is a fifth above the next one, and where all the chords are in the same key. An example of descending fifths is the sequence

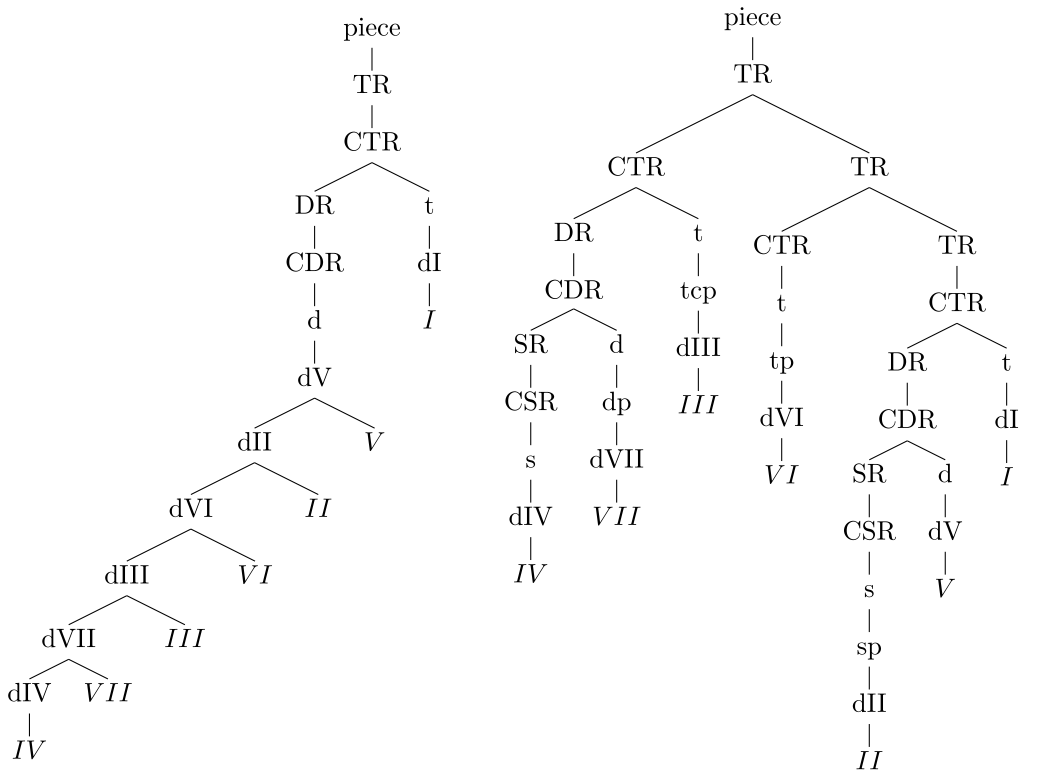
$$Em\ Am\ Dm\ G\ C$$

in the tonality of C: here each chord is the fifth above the chord that follows it. In production (2.12), X stands for any of the non-terminals I,...,VII so the actual productions are:

$$\begin{aligned} dI &\longrightarrow dV \quad I \\ dII &\longrightarrow dVI \quad II \\ dIII &\longrightarrow dVII \quad III \\ dIV &\longrightarrow dI \quad IV \\ dV &\longrightarrow dII \quad V \\ dVI &\longrightarrow dIII \quad VI \\ dVII &\longrightarrow dIV \quad VII \end{aligned}$$

The first symbol in the right-hand side of each production is a non-terminal, allowing the recursive extension of the progression. The second one is a terminal because if it were not, elements that do not belong to the sequence could be generated in the middle of it.

These rules make the grammar ambiguous. For example, the succession $IV\ \ VII\ \ III\ \ VI\ \ II\ \ V\ \ I$ from Figure 4 of [18] can be analyzed two different ways, one that uses the descending fifths sequence (Figure 3.4a) and one that does not (Figure 3.4b).



(a) Using descending fifths sequence

(b) Using the rest of the grammar

Figure 3.4: These two different parse trees of a descending fifths sequence prove that the production (2.12) makes the grammar ambiguous.

In Figure 4 of [18] we can see another way of analyzing the succession using a shorter sequence of fifths.

The descending fifths sequence introduces two new problems in the grammar. First, any sequence of chords that can be analyzed as a descending fifths sequence can also be analyzed without this progression, thus introducing an ambiguity. Secondly, the definition of the descending fifths sequence is, like many concepts in music, a fuzzy one. The pair of chords $V \; I$ can be seen as a descending fifths sequence of length two, although no musical theorist would see it this way as it is understood that a sequence of two chords is not enough to be considered as a progression. It is not clear what is the minimum number of chords needed to determined the existence of a descending fifths sequence, nor whether it makes sense to talk of such a minimum, independent of the harmonic context in which the sequence is placed.

In this work we are not taking on this problem and we shall simply disregard descending fifths sequences, dropping the corresponding production from the grammar. We shall hint at some possible ways of analyzing descending fifths in the final chapter.

3.3 Major and minor mode

In this section we analyze the function-scale degree interface productions ((2.14)-(2.20), p. 7). The productions (2.17) to (2.20) of the original grammar depend on the mode of the key: major or minor. The key value and the mode are not non-terminals of the grammar, but attributes of non-terminals. We have therefore productions that do not depend only on the non-terminal on the left-hand side but, also, on its attributes. This is a situation that lead us out of the scope of our syntactic analysis. Furthermore, productions (2.16) and (2.18), and (2.17), (2.19) and (2.20) produce ambiguities. The ambiguities of the productions (2.14) and (2.15) have been discussed before. To solve these problems we are going to replace these rules with others that do not produce ambiguities and do not depend on the key mode.

In [16] there are various observations on the relations between the major and the minor keys. At first sight, major and minor modes of a key (e.g. C major and C minor) might seem very different (after all, C major has no accidentals, while C minor has three flats). Nevertheless, their harmonic functions are similar: the dominant is the same, and the tonic differs in a semitone. So, functionally, both modes of the key are very similar and some composers of the XVIII and XIX century even consider the two modes as two different aspects of the same key. We are going to follow them, considering that a key is the union of its major and its minor modes, and therefore that the productions are the same in both modes.

As mentioned in Chapter 2, there are three tonal functions: tonic, dominant and subdominant. As specified in [16], the tonal chords (I, V and IV) have the tonic (t), dominant (d) and subdominant (s) function respectively. The II chord could be considered a tonal chord, but weaker than I, IV and V, and it usually has a subdominant function (sp). The VII is a dominant chord, so its function is dominant (dp). Finally, the III and the VI chord are modal chords. Since they share two notes with the I, in this grammar we attribute to them a tonic function (tcp and tp).

Based on these considerations, we simplify the function-scale degree interface productions the following way:

$$
\begin{aligned}
t &\longrightarrow dI \quad | \quad dI \quad dIV \quad dI \\
s &\longrightarrow dIV \\
d &\longrightarrow dV \\
tp &\longrightarrow dVI \\
dp &\longrightarrow dVII \\
sp &\longrightarrow dII \quad | \quad \flat II \\
tcp &\longrightarrow dIII
\end{aligned}
$$

3.4 Modulation and attributes

In this section, we complete our analysis of the base grammar by looking at the modulation rules ((2.9) and (2.10), p. 6). These rules use the key property of each symbol, as modulation is, by definition, the process of changing the key of a musical piece. The idea is that each functional region ($\mathbb{F}\backslash\{t\}$) may become a new local tonic region (TR) in the corresponding key.

In all rules other than the modulation rules, the key property of the right-side symbols of the productions is the same as the key property of the left-side symbols. In the modulation productions, a function changes the key property as the result of the modulation. The difference between the two modulation rules is that the first one determines a modulation (or a tonicization; in this grammar they are indistinguishable) and the second one determines a change of mode (from major to minor or vice-versa).

This formalization of modulation works very well if we are using the grammar to generate harmonies: one applies the rule, changes the key and starts generating a tonal region in the new key. If, on the other hand, we are using the grammar to analyze a sequence of chords, this rule turns out to be too general and creates ambiguities: it allows us to analyze each chord of a sequence as a different tonal region or as a part of the same tonal region. For example the sequence $C\ F\ G\ C$ admits the following valid interpretations:

- $I_C\ IV_C\ V_C\ I_C$
- $I_C\ I_F\ I_G\ I_C$

where the subscript is the key of the corresponding scale degree. This formalization captures the different processes of modulation (diatonic, enharmonic and chromatic), but it does not focus on the process of the modulation itself (how the change of key is made). So, as in the example above, the parsing process could introduce modulations that are theoretically incorrect.

In this case, we are in the presence of an ambiguity of a different nature than the ones we have seen so far; one at a more fundamental level. This ambiguity is not due simply to the way we wrote the grammar, so it cannot be eliminated using the methods of formal grammar manipulation. Here the ambiguity resides in the very concept that we are trying to model. Essential aspects of the concept, such as the difference between a modulation and a tonicization or the position of the chord where we move from one key to another, cannot be formally defined.

The concept is defined in music theory in a qualitative way that makes its formalization in the grammar all but impossible. Musical theory recognizes such a fact:

> "Any chord, or any group of tones, can be interpreted in any key. [...] The ambiguity of the single chord [...] is the basis of the technique of modulation." (W. Piston, *Harmony*, p. 77 [16])

Our solution is to separate the determination of the key from the grammar, using an alternate method to determine the key and then analyzing each non-modulating component using a simpler grammar without modulation. Therefore we remove the key property of the symbols and the modulation rules from the grammar. The originality of our solution consists of mixing two paradigms: we use a form of numerical computing to model the concepts that cannot be formalized, and use the grammar to model the portion of musical theory that allows formalization.

3.5 Resultant grammar

The resultant grammar of the analysis and modifications exposed in last sections is shown in Figure 3.5. We are going to use it to analyze harmonic sequences (i.e., sequences of chords). To perform this analysis we have implemented the first two phases of a compiler: the lexical analyzer and the syntax analyzer [1].

We have used Flex [11] to implement the lexical analyzer, which reads and converts the input into a stream of tokens to be analyzed by the syntax analyzer. A valid input is a sequence of chords, written as scale degrees of a tonality (I, II,...), separated by spaces. As we have separated the modulation from the grammar, all the chords of the input must be in the same tonality. The tokens correspond to the terminal symbols of the grammar of Figure 3.5.

For the syntax analyzer, the unambiguous context-free grammar of Figure 3.5 is converted into a deterministic GLR parser using Bison [11]. The parser obtains a string of tokens (the sequence of chords) from the lexical analyzer and verifies that the string (sequence) can be generated by the grammar. In that case, we obtain the parse tree of the chord sequence, that shows its structure.

Non-terminal symbols:
$\{piece, TR, SR, DR, CTR, CSR, CDR, t, s, d, tp, sp, dp, tcp, dI, dII, dIII, dIV, dV, dVI, dVII\}$
Terminal symbols:
$\{I, II, III, IV, V, VI, VII, \flat II, V^{II}, V^{III}, V^{IV}, V^{V}, V^{VI}, V^{VII}, VII^{II}, VII^{III}, VII^{IV}, VII^{V}, VII^{VI}, VII^{VII}\}$
Start symbol: *piece*
Productions:

$$
\begin{aligned}
piece &\longrightarrow TR\\
TR &\longrightarrow CTR \quad | \quad CTR \quad TR \quad | \quad CTR \quad DR\\
CTR &\longrightarrow DR \quad t \quad | \quad t\\
DR &\longrightarrow CDR \quad | \quad CDR \quad DR\\
CDR &\longrightarrow SR \quad d \quad | \quad d\\
SR &\longrightarrow CSR \quad | \quad CSR \quad SR\\
CSR &\longrightarrow s\\
t &\longrightarrow tp \quad | \quad tcp\\
s &\longrightarrow sp\\
d &\longrightarrow dp\\
t &\longrightarrow dI \quad | \quad dI \quad dIV \quad dI\\
s &\longrightarrow dIV\\
d &\longrightarrow dV\\
tp &\longrightarrow dVI\\
dp &\longrightarrow dVII\\
sp &\longrightarrow dII \quad | \quad \flat II\\
tcp &\longrightarrow dIII\\
dI &\longrightarrow V^{III} \quad I \quad | \quad V^{V} \quad I\\
dII &\longrightarrow V^{II} \quad II \quad | \quad VII^{II} \quad II \quad | \quad V^{IV} \quad V^{II} \quad II\\
dIII &\longrightarrow V^{III} \quad III \quad | \quad V^{V} \quad III \quad | \quad VII^{III} \quad III \quad | \quad VII^{V} \quad III\\
dIV &\longrightarrow V^{IV} \quad IV \quad | \quad V^{VI} \quad IV \quad | \quad VII^{IV} \quad IV \quad | \quad VII^{VI} \quad IV \quad | \quad V^{V} \quad V^{IV} \quad IV\\
dV &\longrightarrow V^{V} \quad V \quad | \quad V^{II} \quad V \quad | \quad VII^{V} \quad V \quad | \quad VII^{II} \quad V\\
dVI &\longrightarrow V^{VI} \quad VI \quad | \quad VII^{VI} \quad VI\\
dVII &\longrightarrow V^{VII} \quad VII \quad | \quad VII^{VII} \quad VII\\
dI &\longrightarrow I\\
dII &\longrightarrow II\\
dIII &\longrightarrow III\\
dIV &\longrightarrow IV\\
dV &\longrightarrow V\\
dVI &\longrightarrow VI\\
dVII &\longrightarrow VII
\end{aligned}
$$

Figure 3.5: Resultant grammar from the analysis of the grammar of [18].

To show how the grammar works, we have analyzed two sequences of chords from the Prelude 1 in C major (BWV 846) of J.S. Bach:

- Sequence 1: $I \ \ II \ \ V \ \ I \ \ VI \ \ V^V \ \ V \ \ I$
- Sequence 2: $VI \ \ V^V \ \ V \ \ VII^{II} \ \ II \ \ VII \ \ I \ \ IV \ \ II \ \ V \ \ I$

Their resulting parse trees are represented in Figures 3.6a and 3.6b respectively. We can see that the sequence 1 is divided in three tonic regions with different harmonic components:

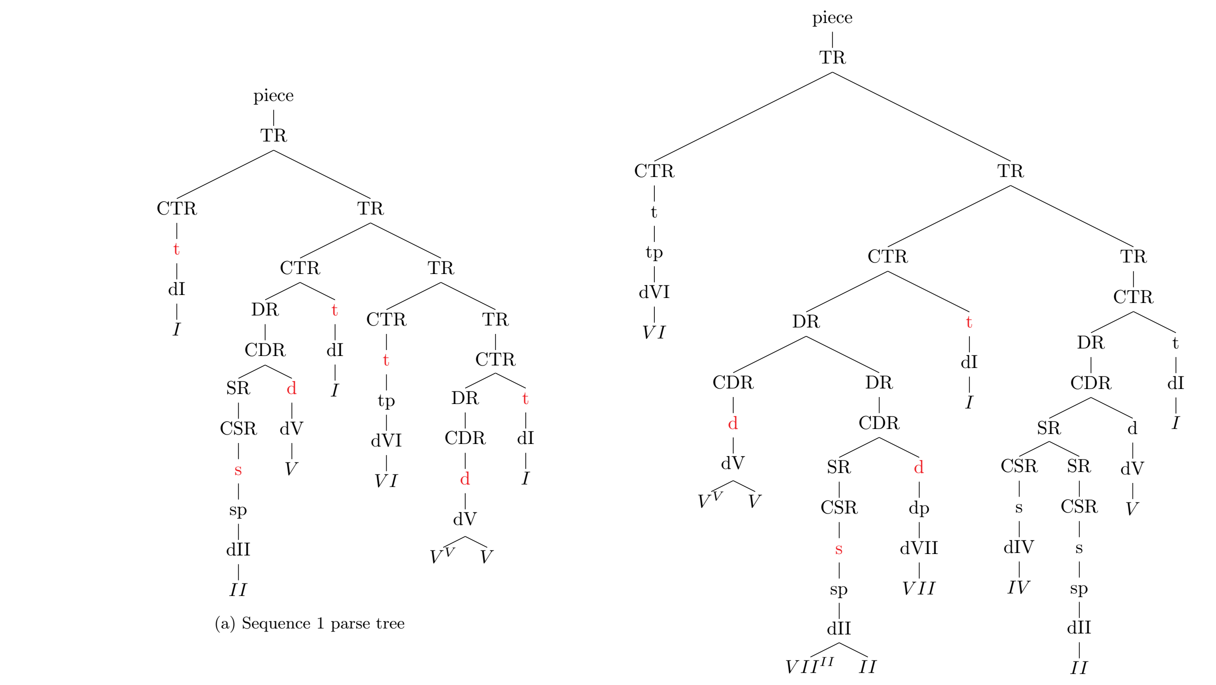
- $I \rightarrow t$
- $II \ \ V \ \ I \rightarrow s \ \ d \ \ t$
- $VI \ \ V^V \ \ V \ \ I \rightarrow t \ \ d \ \ t$

According to the grammar, the subdominant (s) is linked to the following dominant (d) and the dominants are linked to tonics (t). As these three regions are tonic regions, they end in a tonic chord. In the sequence 2 we can see other different harmonic progressions, as:

- $V^V \ \ V \ \ VII^{II} \ \ II \ \ VII \ \ I \rightarrow d \ \ s \ \ d \ \ t$

In both sequences there are secondary dominants with regular resolutions.

In fact the syntax analysis of chord sequences allows us to know the functions of each chord of the sequence and the relationships between them.



(a) Sequence 1 parse tree

(b) Sequence 2 parse tree

Figure 3.6: J. S. Bach, Prelude 1. Parse trees.

3.6 Extensions and limitations

Most of the limitations of the grammar in [18] are linked to the ambiguous nature of music. Musical harmony does not follow some set of strict rules, so it is extremely difficult to generalize all the possible progressions in an unique and unambiguous grammar. It is quite probably impossible to create a grammar that analyzes all harmonic progressions that musicians have used, even if we limit our scope to a specific period - for example, classical music of the XVII and XVIII centuries, which is where most of our examples come from. We must accept that the grammar explains the most comon progressions, while some of the most unusual ones will be left out. In this section we are going to analyze some of these limitations, and propose some extensions that could improve the grammar.

Subdominant chords. According to the grammar, the subdominant regions (formed by the chords IV, II and $\flat II$) are always followed by a dominant chord. This is determined by the production:

$$CDR \longrightarrow SR \quad d$$

The chord IV could also appear in the structure $I \quad IV \quad I$, illustrated in the production:

$$t \longrightarrow dI \quad dIV \quad dI$$

In most cases, subdominant chords appear in these structures defined in the grammar. But there are some cases, as $I \quad II \quad I \quad IV \quad I \quad II \quad I \quad V \quad I$ [1], where the subdominants are used in a different way. That sequence cannot be produced by the grammar, as the subdominant chord II does not precede a dominant chord. These cases are really uncommon in the XVII and XVIII century tonal harmony, and trying to include all of them in the grammar can lead to a disappearance of its hierarchy and structure. So, as we mentioned before, we are going to consider only the common cases, leaving the uncommon sequences to future work.

Secondary dominants. As we explained in previous sections, the grammar of [18] only considers the regular resolutions of secondary dominants. We have expanded the grammar adding some rules for the common irregular resolutions [16], but this approach is still not complete. Because there are not strict rules on resolutions, sometimes a resolution may not be immediate, meaning that, the secondary dominant chord does not resolve in the immediately following chord but in the next one. In this case there is a chord, called a "paso" chord, between the secondary dominant and its resolution. In other cases, the resolution has a complex structure for which it is not easy to give general rules. These special situations are not reflected in the grammar.

When we tried to generalize the secondary dominant rules introducing functional level symbols in the productions, the grammar became ambiguous, because the hierarchy was broken (as we can see in Figure 3.3). So in order to do it, it is necessary to change the location of the secondary dominants in the hierarchy. First, we have to distinguish the dominants that have a resolution:

$$TR \longrightarrow DR \quad t$$

from the ones that do not:

$$TR \longrightarrow TR \quad DR$$

And then, we have to introduce the secondary dominants and a generalization of their resolution at that level. But music does not follow some set of strict rules, it only has "common" progressions, thus a generalization will lead to the disappearing of the music structure of the grammar. Because our objective is to analyze the structure of the music, we have decided to focus in the most common cases, leaving a further study of the secondary dominant resolutions generalization for future work.

Irregular resolutions of dominants. The grammar defines the most common resolutions of the dominants, the regular resolutions, for example the progression V - I. In this section, we study the less common ones, the irregular resolutions. In table 3.2 the most important irregular resolutions [16] are classified depending if they belong or not to the grammar.

[1] from the Song Without Words op. 102 n° 2 of F. Mendelssohn

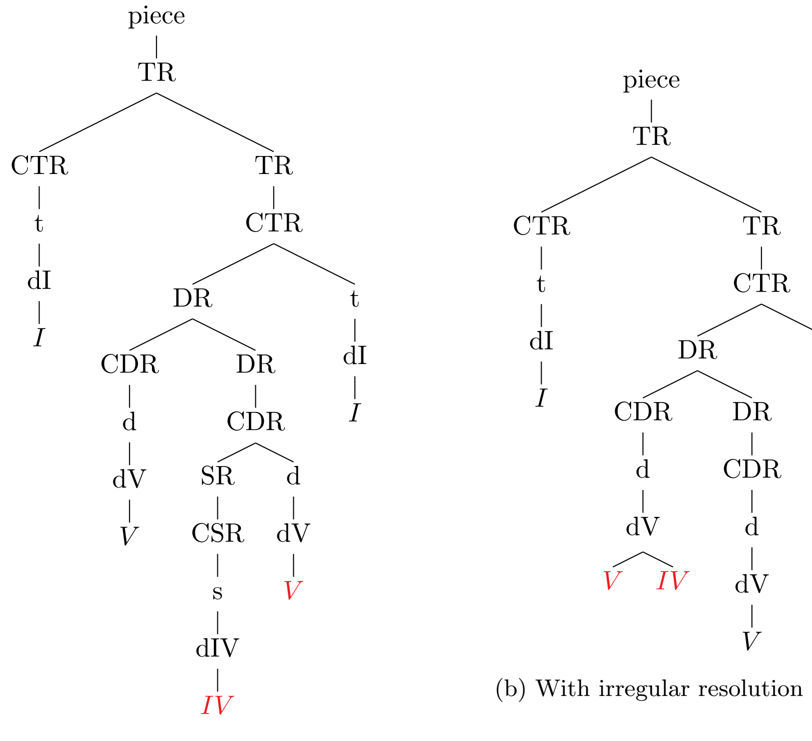
Belong to the grammar	**Do not belong to the grammar**
V - VI V - III	V - IV V - II V - V^V V - V^{IV} V - V^{VI}

Table 3.2: Some irregular resolutions of dominants.

When we analyze a sequence of chords ending with a progression of the second column of Table 3.2, the parser reports a syntax error. So, our first approach was to add new productions, as we did with the secondary dominants:

$$dV \longrightarrow V \quad IV \quad | \quad V \quad II \quad | \quad V \quad V^V \quad | \quad V \quad V^{IV} \quad | \quad V \quad V^{VI}$$

But these productions make the grammar ambiguous. For example, consider the sequence $I \;\; V \;\; IV \;\; V \;\; I$ where the irregular resolution is in the middle. We can analyze it without using the new productions, as shown in Figure 3.7a, where the IV chord is connected to the next one. Or we can analyze it using the irregular resolution production, as shown in Figure 3.7b, where the IV chord is the resolution of the previous one.



(a) Without irregular resolution

(b) With irregular resolution

Figure 3.7: The new productions of the irregular resolutions of dominant chords make the grammar ambiguous: these are two different parse trees of the sequence $I \;\; V \;\; IV \;\; V \;\; I$.

From the theoretical musical analysis point of view, the interpretation of the previous sequence will be different depending on the context and sometimes both interpretations are equally good. In a context-free grammar, such as the one we are considering, we cannot differentiate between the two cases. So to avoid the ambiguity, we decided not to add the previous productions to the grammar, so the grammar will always analyze the ambiguous sequences the way represented in Figure 3.7a. This is a limitation of the grammar, as some sequences cannot be analyzed and others have a theoretically awkward interpretation.

"Paso" chords. When an author links two chords using ornamental notes, a new chord is formed between them, which is called a "paso" chord. It does not have an important function in the harmony as its main function is to embellish and give color to the music. Sometimes, a "paso" chord forms a common harmony with the others and the sequence can be analyzed with the grammar. Other times it forms an uncommon sequence that does not belong to the language generated by the grammar. With our approach we cannot distinguish between a "paso" chord that makes the harmony uncommon or and just an uncommon harmony. The identification and analysis of "paso" chords is left as future work. Once we identify them, they can be deleted from the sequence in order to analyze the chords that have an important function using the grammar.

Harmonic sequences. The harmonic sequence is the systematic transposition of a melodic, rhythmic and harmonic pattern [16]. The length of the pattern varies from a single chord to a long phrase, but the most common are the patterns with two chords. To form a sequence it is necessary to have at least two transpositions of the original pattern. We can classify the harmonic sequences into non-modulating and modulating sequences.

In modulating sequences, each transposition is in a different key and there is no return to the original key. The most common embrace three keys, where the first one is the original key, the last one is the modulation key and the second one is a *passing modulation* that links the first key to the last one. As we have separated the modulation from the grammar, each transposition is analyzed separately.

However the grammar does not identify non-modulating harmonic sequences. So the resulting analysis (if possible) is not the most correct one as it links chords from different transpositions. For example, the harmonic sequence $I\ \ IV\ \ II\ \ V\ \ III\ \ VI\ \ IV\ \ V\ \ I$, whose pattern is $I\ \ IV$ and has two transpositions ($II\ \ V$ and $III\ \ VI$), is analyzed as shown in Figure 3.8. We can see that the parse tree does not show the harmonic sequence structure.

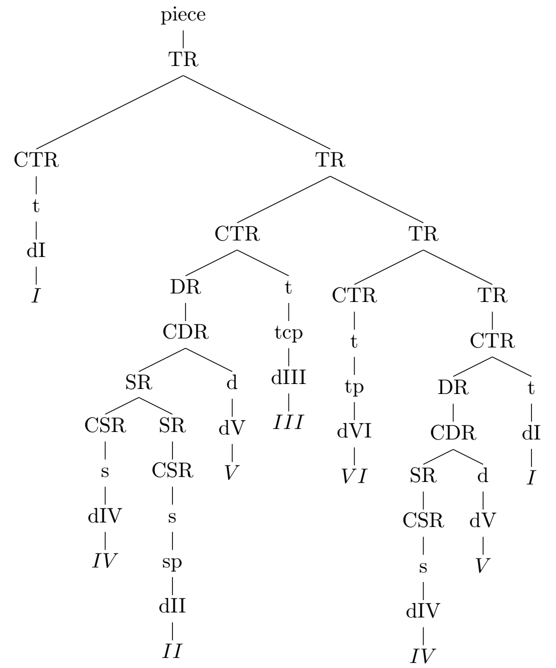


Figure 3.8: The parse tree of the harmonic sequence $I\ \ IV\ \ II\ \ V\ \ III\ \ VI\ \ IV\ \ V\ \ I$ does not show its sequential structure.

There can also be harmonic sequences of secondary dominants, such as

$$V^{II}\ \ V^{V}\ \ V^{(I)}\ \ V^{IV}\ \ V^{VII}\ \ V^{III}\ \ V^{VI}\ \ VI$$

that cannot be analyzed by the grammar because the secondary dominants do not resolve in any chord (except the last one). Thus, we decided to leave the detection and analysis of harmonic sequences aside from this work. We will further discuss this issue in the conclusions.

4

Modulation

In Section 3.4 we arrived at the conclusion that we should separate the detection of the modulations of a chord sequence from the analysis of its harmonic structure. In this chapter we shall explain the musical properties of modulation looking for a guide on how to detect it, then we are going to expose the detection method and finally we shall give some examples.

4.1 Musical properties of modulation

Modulation is the process of changing the key (i.e., changing the tonal center of a musical fragment) and it takes place in three stages: first a tonality has to be made clear to the hearer, then the composer changes the key and finally the new tonality is made clear to the hearer [16]. To make clear a tonality the author may use several and repeated tonic and dominant chords, as they are the ones that most determine a tonality.

One of the most common ways of changing the key is using a *pivot chord*, i.e., a chord that belongs to both keys. For example, in the sequence $C\ F\ G\ C\ Am\ D\ G\ Am\ D\ G$, shown in Figure 4.1, the pivot chord is the fifth chord, Am, which is the VI of C and the II of G.

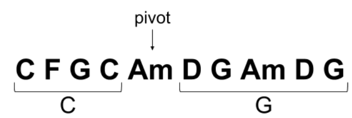


Figure 4.1: Example of a pivot chord that links two tonalities.

The three stages are all equally important: if there is a key change but the new tonality is not well-established (e.g., the duration of the new tonal center is short and/or the chords of the new tonality are weak), the result is not a modulation but a *tonicization*. The main difference between a modulation and a tonicization is the duration of the new tonal center, although sometimes it is difficult to distinguish them: depending on the context, the musical analyst could consider that either a modulation or a tonicization took place. This is one of the reasons why modulation is a mathematically ill-defined concept: there is no clear, unambiguous definition of whether a key change is a modulation or a tonicization.

Note that when the new tonality is the other mode of the original tonality (e.g., when the original tonality is A minor and the new tonality is A major), the process is not called modulation, but *change of mode*. It is also common to use chords of both modes of the key equally, resulting in what is called *mode mixture*. For these reasons and given the relations between major and minor modes exposed in

Section 3.3, we are going to merge the two modes of the same key (e.g, A major and A minor) into a common key (A).

We have seen that each chord can be interpreted in any key, and in each key it has a different harmonic function. This is what makes the process of modulation inherently ambiguous, as each fragment has many (exponentially many, in fact) interpretations. As an example, the fragment in Figure 4.1 can be interpreted as:

C	F	G	C	Am	D	G	Am	D	G		
I_C	IV_C	V_C	I_C	VI_C/II_G	V_G	I_G	II_G	V_G	I_G	-	With modulation
I_C	IV_C	V_C	I_C	VI_C	V_C^V	V_C	VI_C	V_C^V	V_C	-	Without modulation

or even with more unlikely combinations such as:

C	F	G	C	Am	D	G	Am	D	G
V_F	I_F	III_E	VI_E	IV_E	I_D	V_D	I_{Am}	V_G	I_G

where the subindex is the key of the corresponding scale degree.

From the point of view of a formal grammar, all these solutions are valid, and each one of them generates a formally correct harmonic structure. From the musical point of view, not all these interpretations are plausible. The ear "hears" quite clearly the modulation of the first interpretation, and it might possibly hear the tonal continuity of the second interpretation. Nobody would hear the crazy interpretation of the third example. It is true that all chords can be interpreted in all tonalities, but not all interpretations are equally good. A piece of music can exhibit a few chords far away from the tonal center without constituting a modulation. But when there are many chords that have a more natural interpretation in a new tonality than in the current one we are in the presence of a modulation. Note that, in abstract term, the interpretation of modulation is reminiscent of an optimization problem: given a sequence of chords, we look for the tonality with the biggest value, as it will be the more "natural" interpretation of it.

A second ambiguity of the modulation process is the exact location of the chord where the key changes. Many times there is a sequence of common chords in both tonalities, and all of them are a good option to be the point of modulation.

In the next section we use the harmonic functions of the chords in each tonality to determine the best key for each sequence. Using that, we will be able to determine modulations of chord sequences and to identify pivot chords.

4.2 Modulation detection algorithm

Given a sequence of chords, our objective is to determine if there is any modulation and where, and to identify the keys in each fragment of the sequence. Although modulation is an ambiguous process, not all the keys are a good match for a sequence of chords. Usually, there is a key that is the best interpretation or there are two or three keys that are reasonably good. In the latter case we have to decide what is the best interpretation depending on the context. To do this we are going to use the relations between chords and keys.

4.2.1 Chords and keys

Even though any chord could be interpreted in any key [16], given a key, there are chords that are more related to it than others. For example, Table 4.1 shows the relation between a number of chords and the key of C. The highlighted chords are the most "natural" for the key. Note that, in principle, any possible chord has some harmonic relation with any key. However, some of these relations are so remote that, in practice, only a limited number of chords can be considered to characterize the key. We decided that in our method we would only take into account the chords that are used the most, because they are the ones that determine the key. These are the following:

- The chords from the major scale (*C*, *F*, *G*, ...)
- The chords from the melodic and harmonic minor scales (without the augmented chords, because they are rarely used [16]) (*Cm*, *Dm*, *E♭*, ...)
- The secondary dominants of the major, melodic and harmonic minor scale chords (if applicable) (*A*, *B*, *D*, ...)
- The dominant seventh chords (*G7*, *A7*, *B♭7*, ...)
- The Neapolitan chord (*D♭*)

		I	II	III	IV	V	VI	VII	♭II
C (B♯)	**Major**	C	Dm	Em	F	G(7)	Am	B°	
	Secondary dominants	-	A(7)	B(7)	C7	D(7)	E(7)	-	
Cm	**Ascending Melodic**	-	Dm	-	F	-	-	-	D♭
	Harmonic minor	Cm	D°	-	Fm	G(7)	A♭	B°	
	Descending Melodic	-	-	E♭	-	Gm	-	B♭	
	Secondary dominants	-	-	B♭7	C7	D(7)	E♭7	F7	

Table 4.1: Set of chords considered to be closer to the tonality of C. The colors represent the weight assigned to each chord in the tonality of C: the green color means a weight of 5.0; the orange, a weight of 3.0; the yellow, a weight of 2.0; and the white, a weight of 1.0.

The rest, such as the augmented chords of the minor scales, the diminished sevenths chords or the augmented sixths chords will not be taken into account here; they could be studied as an extension of the current work.

Each chord of that set has a function within the key that determines its importance in the key; given a key, we use it to assign a *weight* to each chord depending on its importance in the key. If the chord is very important and its appearance determines strongly the key, then it will have a bigger weight that a less important chord. The function and importance of each chord in a key are derived from musical theory [16]. Based on these data, we determine a weight that we assign, for each key, to the relevance chords for that key. We have tested different weights in non-modulating sequences and the more suitable ones are reflected in Table 4.1. The secondary dominants and the Neapolitan chord are the least important chords from the set defined above, so we assign them a weight of 1.0 (In Table 4.1 they are the ones with white background). The tonic (*I*) and the dominant (*V*) are the most important chords of the key, so they have the largest weight: 5.0 (These are *C*, *Cm*, *G* and *G7*; in green background). The rest of the chords (chords of the major, melodic and harmonic minor scales) have and intermediate weight (2.0 in yellow background or 3.0 in orange background) that depends on how often a chord is used within that key. If a chord does not appear in the table of a key, we consider that it has a weight of 0.0 in that key. The sets of related chords for each existing key can be found in Annex B.

4.2.2 Detecting the key of a sequence of chords

We use the weights of the chords in each key to determine the key of a sequence of chords. It is important to note that it makes no sense to determine the key of a single chord: without a context, any chord can be interpreted in any key. If we have two chords, we have a clearer indication of the possible keys, but we still need more chords to have a clear predominant key.

To figure out the key of a sequence of chords, we consider the weight of each chord in each of the twelve keys. The value of a key at the end of the sequence is the sum of the chords' weights in that key. The key with the largest value at the end of the sequence of chords will be the key of the fragment.

For example, given the sequence *C F C Dm G7 C G7 C*, Table 4.2 shows the process of detecting the key. Each column represents the weight of the chord in each of the twelve keys (e.g. *C* has a weight of 5.0 in the key of C, 3.0 in the key of G, 2.0 in the key of D, etc.). Each row represents the weights that each chord of the sequence has in a key. The last column represents the sum of all the weights in each row, that is, the total weight of the chord sequence in each key. We can see that C has the largest total weight (36.0), so we can determine that the chord sequence is in the key of C.

		Chords								Total
		C	**F**	**C**	**Dm**	**G7**	**C**	**G7**	**C**	
Keys	**C**	5	3	5	3	5	5	5	5	**36**
	G	3	2	3	2	1	3	1	3	18
	D	2	2	2	5	1	2	1	2	17
	A	2	2	2	3	1	2	1	2	15
	E	2	1	2	0	1	2	1	2	11
	B	1	0	1	0	0	1	0	1	4
	G♭/F♯	0	1	0	0	0	0	0	0	1
	D♭/C♯	1	1	1	0	0	1	0	1	5
	A♭/G♯	1	1	1	0	1	1	1	1	7
	E♭/D♯	1	1	1	0	1	1	1	1	7
	B♭	1	5	1	2	1	1	1	1	13
	F	5	5	5	3	1	5	1	5	30

Table 4.2: Detecting the key of the chord sequence *C F C Dm G7 C G7 C* adding the weights of the chords in each key.

4.2.3 Detecting modulations

In music, chords sound sequentially, one after another. So, modulation will take place sequentially as well: there will be a sequence of chords in a key and at some point the following chords will be in another key. For example, in Table 4.3 we can see that the sequence *C F C Dm G7 C G7 C C C Fm* E^o *Fm* E^o *Fm* is divided in two overlapping parts: the first part is in the key of C (*C F C Dm G7 C G7 C C*) and the second part is in the key of F (*C C Fm* E^o *Fm* E^o *Fm*). Also, the last chord of the C subsequence is at the same time the first chord of the F subsequence; such chord belongs to both keys. This intermediate chord is the *pivot chord* and it is the locus where the modulation takes place. It is important to note that there are three *C* chords in a row, and each one of them could be considered the pivot chord (example of the ambiguity of modulation).

	C	**F**	**C**	**Dm**	**G7**	**C**	**G7**	**C**	**C**	**C**	**Fm**	**E^o**	**Fm**	**E^o**	**Fm**
C:	I	IV	I	II	V7	I	V7	I	I						
								Fm:	V	V	I	VII7	I	VII7	I

Table 4.3: Harmonic analysis of the sequence *C F C Dm G7 C G7 C C C Fm* E^o *Fm* E^o *Fm*.

If we take a subsequence of chords within the first part of the chord sequence (that is, in the key of C) and we analyze it, the predominant key will be C. The same happens if we take a subsequence of chords within the second part (the F key part): when we analyze it, the dominant key will be F. On the other hand, if we take a subsequence overlapping the two parts (i.e., the first half of the chords in C and the second half in F), when we analyze it, both keys (C and F) will have similar weight.

So, the idea is to have a *sliding window* in the sequence of chords. At each step the sliding window will define a subsequence that we analyze and whose key we determine using the method of Section 4.2.2. At each step, we slide the window one chord forward and repeat the analysis: if the predominant key in adjacent windows is the same then there is no modulation in that fragment, if there is a change of key, then it is possible that a modulation has taken place. Depending on the length of the sliding window (called *window length* in the following and indicated as W) we obtain different subsequences from the original chord sequence. Given the chord sequence example above, and choosing a window length equal to 8, we obtain eight subsequences of the original sequence, as shown in the rows of Table 4.4. The first one goes from the first to the eighth chord, the second, from the second to the ninth chord, and so on.

We then analyze the subsequences, determining the predominant key of each one. This is shown in Table 4.5, where in each column is the total weight of the subsequence for each key. We can see that the predominant key of the first five subsequences is C (with the key weights 36, 36, 38, 36 and 33 respectively) and the key of the last three subsequences is F (with 34, 32 and 36 respectively). Figure 4.2 is a representation of the first and last row (C and F) of Table 4.5: the variation of C and F weights through time. The x axis is the position of the sliding window of the chord sequence and the y axis is the key weights. The curves have been smoothed to give a better sense of the time variation.

Subsequence	**Chords**														
1: 1-8	C	F	C	Dm	G7	C	G7	C							
2: 2-9		F	C	Dm	G7	C	G7	C	C						
3: 3-10			C	Dm	G7	C	G7	C	C	C					
4: 4-11				Dm	G7	C	G7	C	C	C	Fm				
5: 5-12					G7	C	G7	C	C	C	Fm	E^o			
6: 6-13						C	G7	C	C	C	Fm	E^o	Fm		
7: 7-14							G7	C	C	C	Fm	E^o	Fm	E^o	
8: 8-15								C	C	C	Fm	E^o	Fm	E^o	Fm

Table 4.4: Representation of the sliding window of the sequence *C F C Dm G7 C G7 C C C Fm E^o Fm E^o Fm* with $W = 8$.

		Subsequences							
		1-8	**2-9**	**3-10**	**4-11**	**5-12**	**6-13**	**7-14**	**8-15**
Keys	C	**36**	**36**	**38**	**36**	**33**	31	26	24
	G	18	18	19	16	14	13	10	9
	D	17	17	17	15	12	11	11	10
	A	15	15	15	13	10	9	7	6
	E	11	11	12	10	10	9	7	6
	B	4	4	5	4	4	4	3	3
	G♭/F♯	1	1	0	0	0	0	0	0
	D♭/C♯	5	5	5	6	6	8	7	9
	A♭/G♯	7	7	7	9	9	11	10	12
	E♭/D♯	7	7	7	9	9	11	10	12
	B♭	13	13	9	10	8	9	8	9
	F	30	30	30	30	30	**34**	**32**	**36**

Table 4.5: Complete analysis of the chord sequence *C F C Dm G7 C G7 C C C Fm E^o Fm E^o Fm* with a window length of $W = 8$. In bold, the biggest value of each subsequence, which determines its tonality.

In the first five subsequences the predominant key remains the same, which implies that there is no modulation. In the sixth subsequence the predominant key changes to F, and the following subsequences remain in F, which means that there is a modulation from C to F in a chord of that subsequence. In Figure 4.2 the modulation point (the pivot chord) is at the intersection of C and F curves. At first, C has higher values but when the modulation point is closer, the C values decrease and the F values increase. That is because the window starts containing F chords, so the key C has less weight and the key F has more weight. When the subsequences have more F chords than C chords, the graph of F starts having higher values than C. Note that after the modulation point, the C values keep decreasing while the F values keep increasing, thus reinforcing the establishment of the new key.

We have implemented this modulation detection algorithm in the programming language of C. The input is a sequence of chords and the output is the key weights for each subsequence. We then represent this data in an area chart like the one in Figure 4.2.

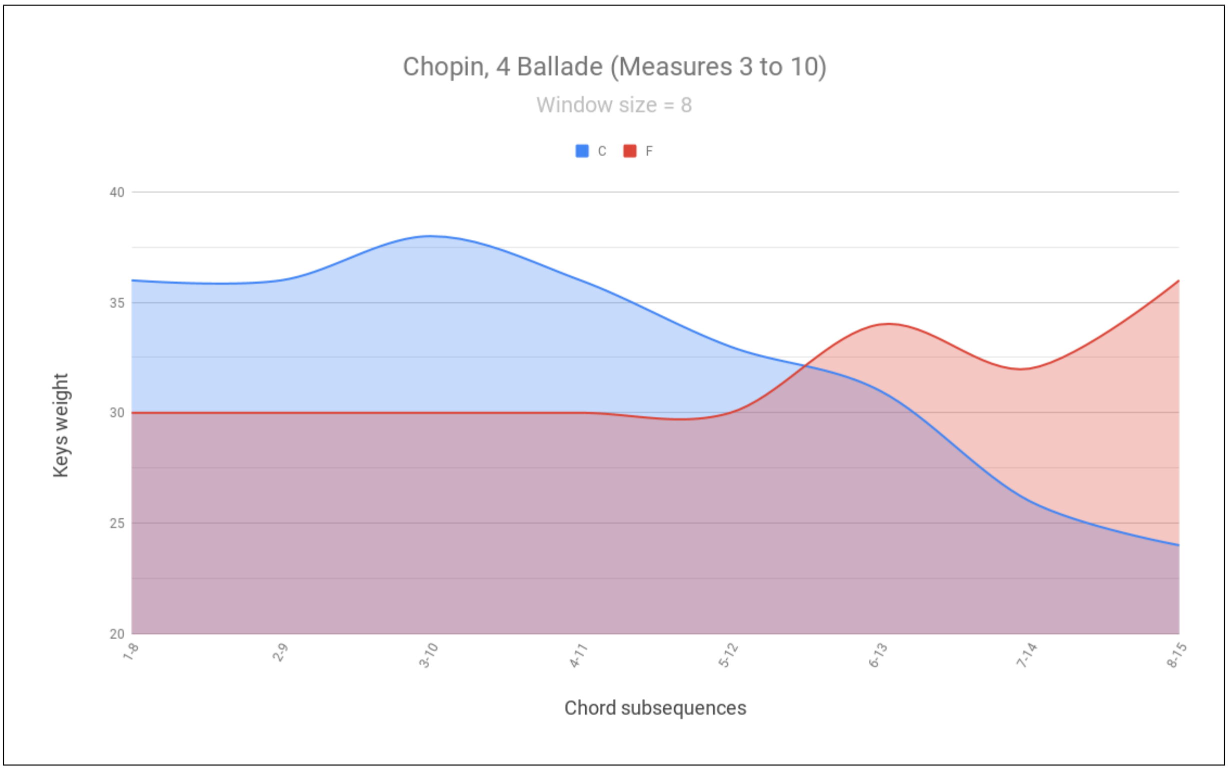


Figure 4.2: Representation of the results of the tonalities of C (blue curve) and F (red curve) from Table 4.5, where the x axis is the position of the sliding window and the y axis is the resulting weight of the subsequence in a tonality.

4.3 Adjustments

One crucial parameter in our method is the window length, W. Depending of it, the results are different because we will be analyzing more or less chords in each window. Also, a point that we have left open is the location of the pivot chord. In this section, we shall first analyze the effect of the window length on the detection of modulation, then turn our attention to the individuation of the pivot chord.

4.3.1 Window size

As we mentioned already, a single chord does not determine a key. Rather, a set of chords and the relationships among them is what determines a key. So a window length $W = 1$ makes no sense. A window length too small ($W \in \{2, ..., 4\}$) is not useful, because there is not enough information in each subsequence to determine a predominant common key over time in a non-modulant sequence of chords. For example, the sequence $C\ Dm\ F\ G\ C$, which is in C ($I\ II\ IV\ V\ I$), analyzed with a window length of two has different predominant keys, whereas there is only one predominant key with window length 5, as shown in Table 4.6. That is because two chords, for example $G\ C$, have a good interpretation in many keys: $V\ \ I$ in C, $I\ \ IV$ in G, $I\ \ V$ in F, etc.

On the other extreme, a window too long could prevent us from detecting some modulations. That is because if we analyze a lot of chords at a time, short modulations followed by a return to the same key could fall within a window and be "averaged out" by the chords in the dominant key, thus being lost.

Short windows are useful when the chord sequence is short or the harmonic rhythm is fast (i.e., there are a lot of keys and few chords in each one). The Chorale 320 (*Gott sei uns gnädig*) of Bach is an example of such a short chord sequence (22 chords). The result of its analysis with a small window length (6) and a large window length (10) is represented in Figure 4.3. If we analyze the chorale using music theory we obtain that the chorale starts in A major and it modulates to F♯ minor in the second third. In Figure 4.3a we can see clearly that at the beginning the predominant key is A and at the end it is F♯. However, when we analyze it with a larger window length (Figure 4.3b), the predominant key is

		W = 2				W = 5
		1-2	**2-3**	**3-4**	**4-5**	**1-5**
Keys	C	**8**	6	**8**	**10**	**21**
	G	5	4	7	8	15
	D	7	7	5	5	14
	A	5	5	4	4	11
	E	2	1	3	4	7
	B	1	0	2	3	4
	G♭/F♯	0	1	2	1	2
	D♭/C♯	1	1	1	1	3
	A♭/G♯	1	1	2	2	4
	E♭/D♯	1	1	2	2	4
	B♭	3	7	6	2	10
	F	**8**	**8**	6	6	19

Table 4.6: Analysis of *C Dm F G C* with a window length of $W = 2$ and $W = 5$.

A almost throughout the piece, and at the end there is an ambiguity between B and F♯. This happens because the window size (10 chords) is quite large with respect to the length of the piece (22 chords), so even the last subsequence has F♯ chords and A chords analyzed together, so the key is not as clear as if there were only F♯ chords (that happened with a smaller window length as in the Figure 4.3a).

An example of fast harmonic rhythm is the Allegretto of the Sonatina Op. 36 n° 2 of Clementi, represented in Figure 4.4. Theoretically, the sequence of modulations is G - D - A - G with D and E tonicizations within the last key. With a small window length (Figure 4.4a) we can see clearly all the modulations and all the predominant keys in each part of the piece. However, when the window is bigger (Figure 4.4b), the first key and some tonicizations disappear. The reason is that each key has but a few chords, so if the window is too big, then in each subsequence there are chords from two or three keys, affecting the final values.

On the other hand, long sliding windows are useful when the chord sequence has many chords and the harmonic rhythm is slow or when we want to identify the main keys in an ambiguous fragment. The beginning of the Waldstein Sonata of Beethoven is harmonically ambiguous, as we can see in Figure 4.5a: the key values oscillate rapidly, several keys have the same total weight, and we see a lot of very rapid changes of the dominant key. When we use a longer window (Figure 4.5b), we can appreciate that the dominant key of the beginning of the Waldstein Sonata is C.

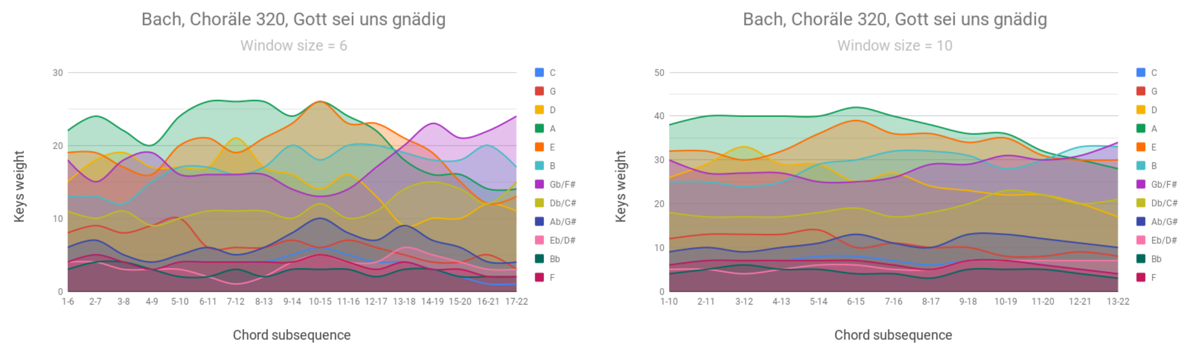
The optimal window length depends on several things: the length of the chord sequence, the harmonic rhythm, the harmonic ambiguities and what we want to analyze. After analyzing several chord sequences using different window lengths, we conclude that the best window length is in the range from 6 to 12 chords. Depending if we want to see all the harmonic regions (modulations and tonicizations) or if we only want to see the main ones (the modulations), we will choose a small or a big window length of the range.

The sequence behaves as a low-pass filter: the bigger W is, the more it filters, i.e., only the main keys appear. When W is small, more keys appear, as the tonicizations and passing modulations are not filtered.

4.3.2 Location of the modulation: pivot chord

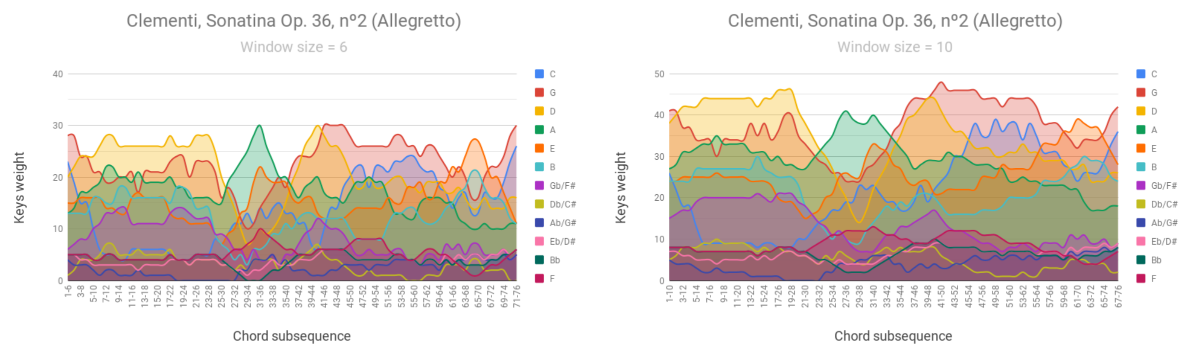
As we mentioned in Section 4.1, one key aspect of the modulation is the determination of the exact location where the change of the key takes place. Depending on the type of modulation there could be a pivot chord (i.e., a common chord between the two keys that separates them), several pivot chords (all of them equally valid) or none (one chord is in a key and the next one is in the other key). The most common situation is having several chords common to both keys, being one of them the "natural" pivot chord. So, determining the pivot chord is a difficult and ambiguous task.

In the modulation algorithm elaborated before, the modulation is detected when a subsequence has more chords of a different key than the current one and this is when half of the subsequence has chords in one key and the other half has chords in the other one. So, in order to simplify it, we decided to take



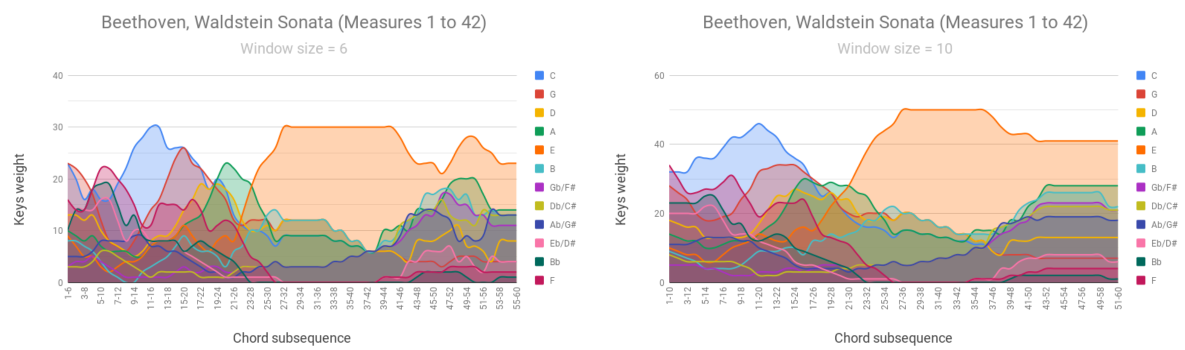
(a) Window length = 6 (b) Window length = 10

Figure 4.3: J. S. Bach, Chorale 320, "Gott sei uns gnädig". Graphs from the analysis of the modulations using different window lengths.



(a) Window length = 6 (b) Window length = 10

Figure 4.4: M. Clementi, Sonatina Op. 36, nº 2. Graphs from the analysis of the modulations using different window lengths.



(a) Window length = 6 (b) Window length = 10

Figure 4.5: L. Beethoven, Waldstein Sonata. Graphs from the analysis of the modulations using different window lengths.

as the pivot chord the chord in the middle of the first subsequence with a change of key: that is to say, we take the chord in the position $\lceil W/2 \rceil$ of the subsequence.

We tested this approximation with 55 modulations from different musical pieces and different window sizes. The results are shown in Figure 4.6. We can see that almost half of the time the estimated pivot chord coincides with the theoretical one and more than 75% of the time it is at a distance no greater than 1 from the theoretical pivot chord.

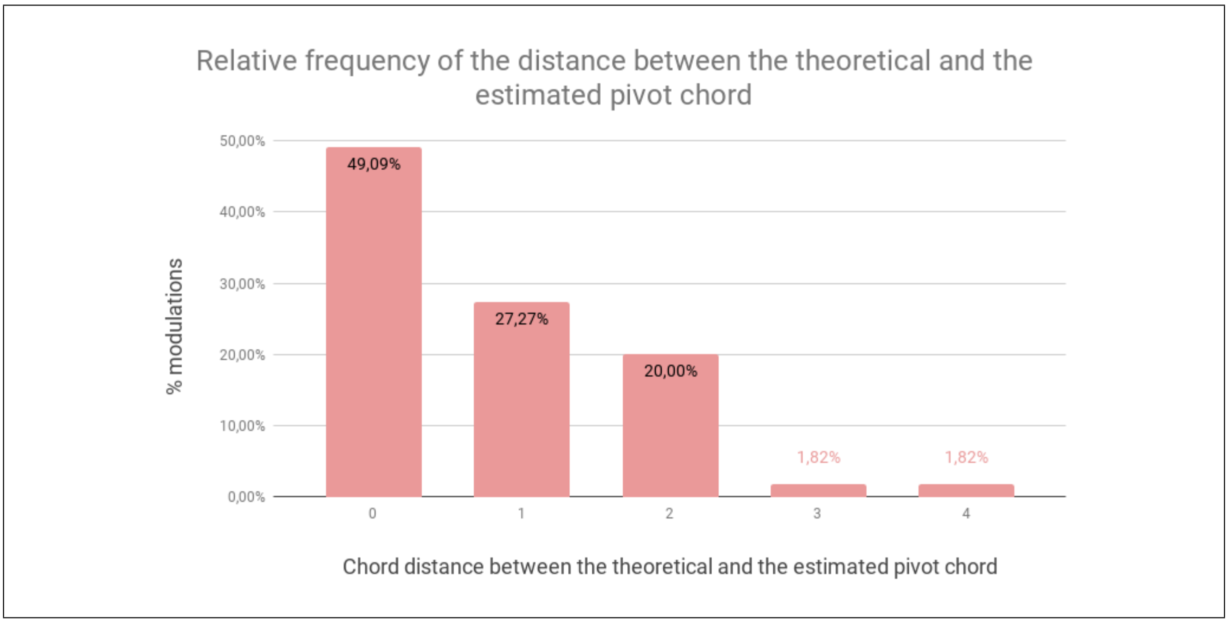


Figure 4.6: Graph that shows the accuracy of the pivot chord estimation. The data is from a sample of 55 modulations from different musical pieces and different window sizes.

4.4 Detections of different types of key changes

A key change does not necessarily imply a modulation. There is a modulation when the keys before and after the change are both clearly established. There are two elements to establish a key: the harmony (using tonic and dominant harmony) and the time (a key is more clear if the chord subsequence in that key is longer). Using our method, keys are established faster if the corresponding segment contains a lot of dominant and tonic chords, as these chords have the largest weights. If a long sequence of chords is in the same key, our method will have that key as the winner in several consecutive windows, and that implies that the key is well-established. We can see an example of modulation from C to F in the area 1 of Figure 4.7. The piece starts in C and stays in C for quite some time (establishment of the first key). When the modulation takes place, C starts having less weight and F more weight, so F starts being the predominant key. The C weight continues decreasing and F is established for a long time (the second key is made clear).

Most of the times is the music time (the length of the chord subsequence in the key) what determines whether a key change is a modulation or not. When there is a key change and the new key is predominant for a short period of time making way for a third key, then we are not in the presence of a modulation. If the third key is the same as the first one, the change is called a tonicization. If the third key is different from the first then it is called a false or passing modulation. The area 2 and 3 of Figure 4.7 are examples of passing modulation and tonicization respectively. In the area 2 we can see a progression from A♭ to D♭ to B♭. A♭ and B♭ are predominant for a significant amount of time, but D♭ is predominant for a short period of time having the function of a nexus between A♭ and B♭. In the area 3 we can see that within the B♭ key there are a few chords of F. F is predominant for a short period of time and B♭ weight does not decrease a lot: when F is predominant, B♭ is the second predominant key, then B♭ starts being predominant again. This usually happens when there are more harmonies related with a chord of the key that is not the tonic. In the example, F is the *V* of B♭, and in that fragment there are more chords related to F, but not with the purpose of modulating to F but to have more dominant harmony in B♭.

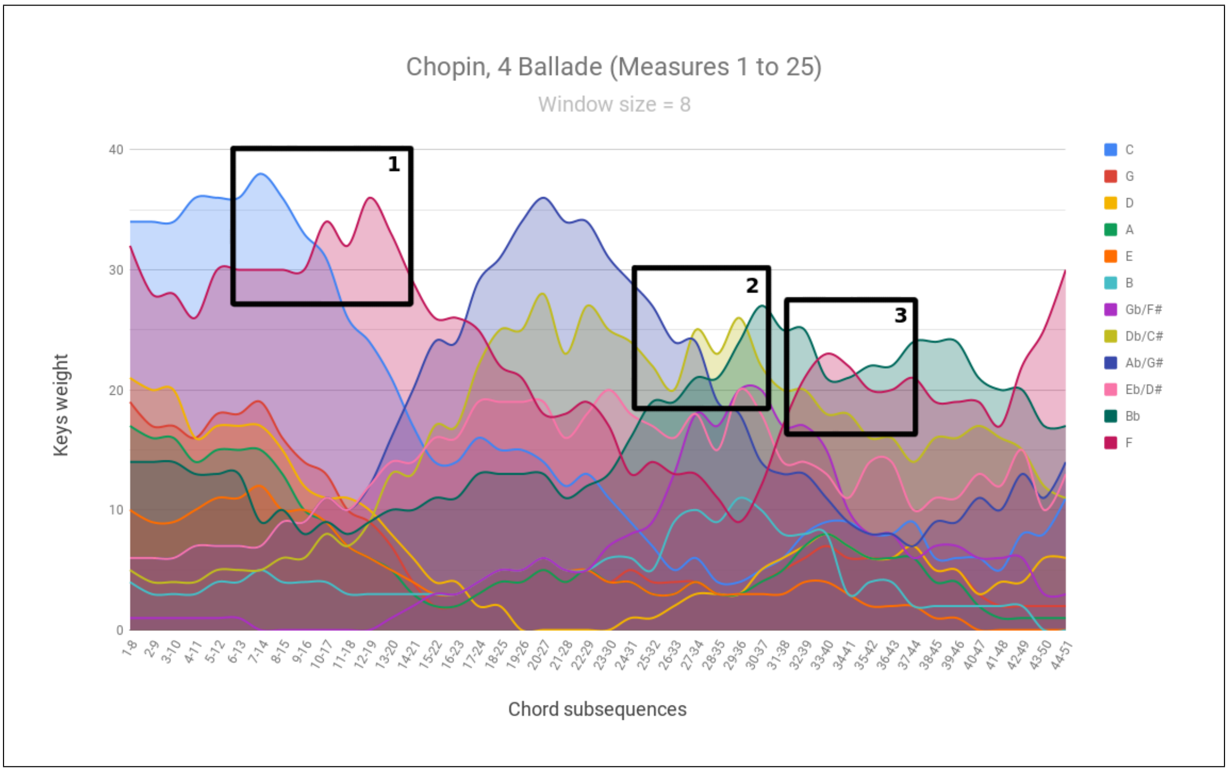


Figure 4.7: F. Chopin, 4 Ballade. Graph from the analysis of the modulations. The boxes point out different types of modulations.

4.5 Examples

In this section we shall give a few examples of results of the above method and further explain other examples exposed in last sections.

Figure 4.8 is an example of analysis of the Pathetique Sonata of Beethoven. The piece starts with a tonicization in F, the *IV* degree of C. We know that because C has high values until chord 50, even in the fragments where F is predominant. Then, there are several modulations: from C to G, from G to A♭, from A♭ to B♭ and from B♭ to E♭. The last change of key (from E♭ to D♭) is theoretically analyzed as a tonicization. The reason why it appears to be a modulation is that it is a long fragment of tonicization of the *VII* of E♭: D♭. This is a good sample of the ambiguity of modulation.

In the Bach chorale of Figure 4.3a we can see that the key E is at first a tonicization (when its values are equal to A values) and then a passing modulation (as a nexus between A and F♯). In the Figure 4.4a we can see that even though the movement of the Sonatina of Clementi is in G (starting and ending in G), it has several modulations and two tonicizations in the last G part: to D, which is the *V* of G and to E that is the *VI* of G. We have already talked about the ambiguity of the beginning of the Sonata Waldstein, represented in Figure 4.5. We can see the particular predominant keys in each part in Figure 4.5a and the main key (C) in Figure 4.5b. In both we can see that A is an passing modulation between C and E.

In fact, this method allows us to identify the predominant key in each part. Even though it does not always coincide with the theoretical modulation that we can obtain analyzing by hand, it indicates the actual harmony that we are listening. Because, even if we have classified modulations by type, that classification is ambiguous and depends largely on context - the predominant key (the key closest to the existing chords) will always be the same.

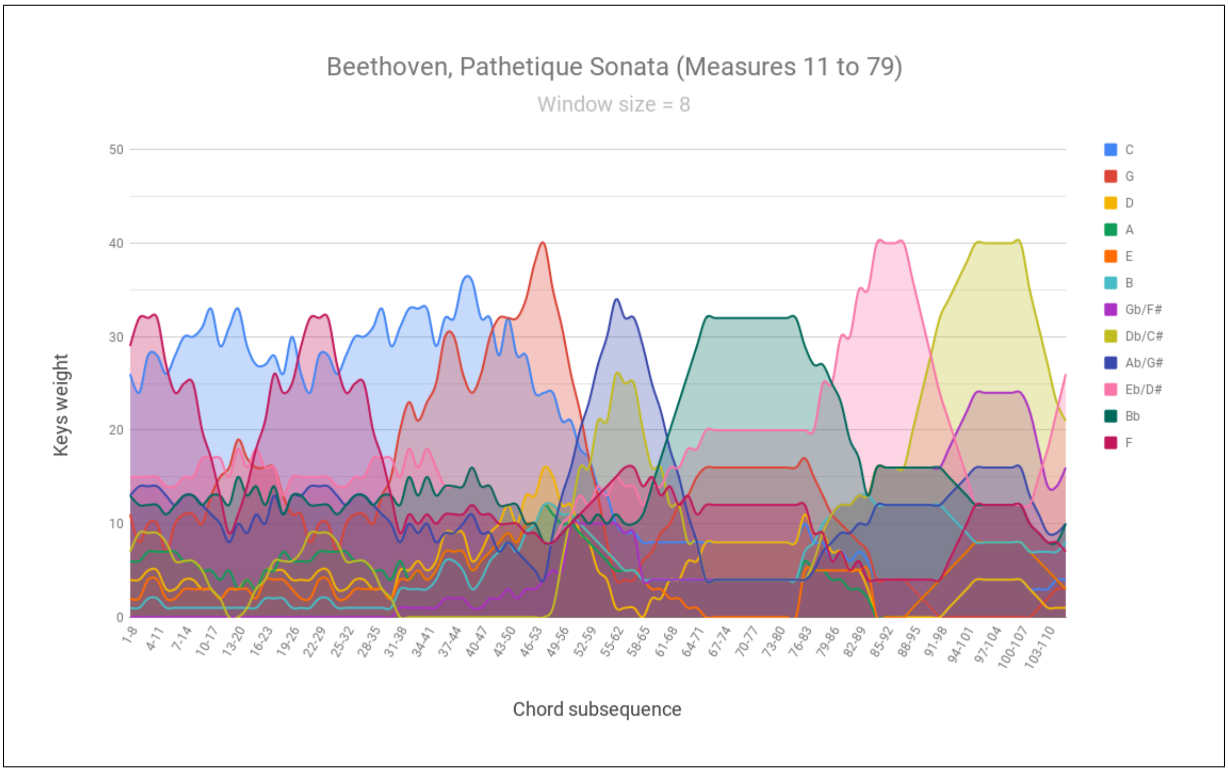


Figure 4.8: L. Beethoven, Pathetique Sonata. Graph from the analysis of the modulations.

5

Case studies

In this section we are going to expound a complete analysis of a few pieces. First we are going to use the modulation detection algorithm of Chapter 4 to detect the modulations in the input sequence and then we are going to analyze each non-modulant subsequence with the grammar defined in Chapter 3.

5.1 Wolfgang Amadeus Mozart: Eine kleine Nachtmusik

For this analysis we have chosen a fragment from Eine kleine Nachtmusik, a famous piece of Mozart. The fragment is from measure 18 to 29 and a theoretical analysis of it can be found in the example 14-3 of [16]. The sequence of chords is the following:

G D G D G C F♯° G D G C♯° D A D A D Em E A D A D A D B Em A

When we analyze it with the modulation detection algorithm we obtain that the sequence begins in G and then modulates to D, as we can see in Figure 5.1. As the grammar analyzes sequences without modulations, we need to divide the chord sequence in two parts: the part that has G as the predominant key and the part in D. So we need to set a modulation point in the chord sequence to divide it. The first subsequences have G as the predominant key, the seventh subsequence (from chord 7 to 14) has G and D with the same weight and the rest of the subsequences have D as the predominant key. So we are going to use the seventh subsequence (represented in red in Figure 5.2) to approximate the modulation point: the pivot chord. As explained before, we are going to take as the pivot chord the one in the middle of the subsequence, which in this case is the fourth one (G, the one in a box in Figure 5.2).

Since the pivot chord is common to both keys, when we divide the sequence, that chord belongs to both subsequences: it is the last chord of the first one and the first chord of the second one (represented in red in Table 5.1). We also need to translate the English notation of the chords into the Roman numerals notation, which is what the grammar requires. Both the division of the sequence and the change of notation is represented in Table 5.1.

When we use the grammar to analyze the sequences of Table 5.1, we obtain the parse trees of Figures 5.3 and 5.4. We can clearly see in Figure 5.3 that the G sequence is divided in tonic regions. Each tonic region is below a CTR symbol (in red) and all of them end in a relaxation point with the I. There are different progressions: with a single chord (I), a dominant-tonic progression ($V \;\; I$) and a subdominant-dominant-tonic progression ($IV \;\; VII \;\; I$). Moreover, the parse tree of the D sequence (Figure 5.4) has tonic and dominant regions. Most of them are tonic regions below the symbols CTR, but we can observe that the sequence ends with a dominant region below the CDR symbol. This region, ending with a tension point with a V chord, has also a secondary dominant of the II. Both the secondary dominant

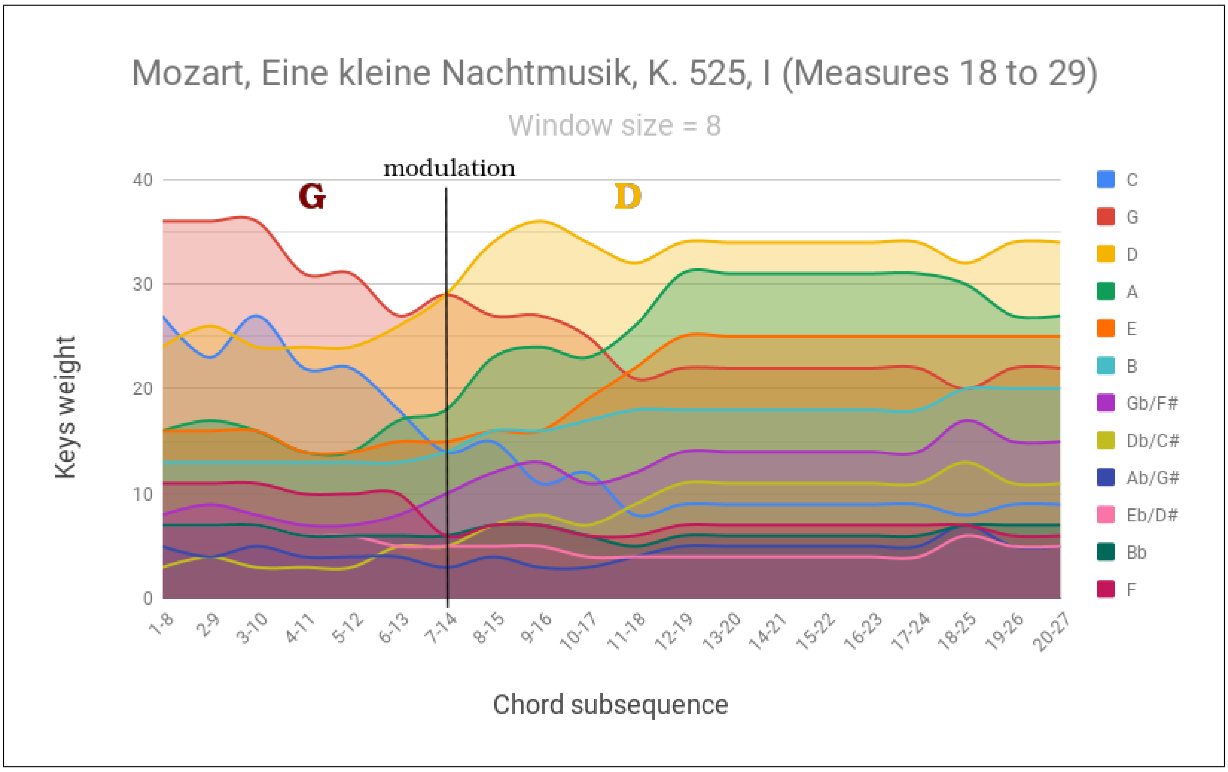


Figure 5.1: W. A. Mozart, Eine kleine Nachtmusik, K. 525, I. Graph from the analysis of the modulations.

G D G D G C **F♯º G D G C♯º D A D** A D Em E A D A D A D B Em A

Figure 5.2: Modulation area (in red) of the Eine kleine Nachtmusik's fragment.

and the II form a subdominant region that prepare the next chord: V. The tonic regions are similar to the ones of the G sequence. There is a secondary dominant of the V in the sequence $II \quad V^V \quad V \quad I$, where V^V and V form a dominant region that is resolved in the following I.

5.2 Johann Sebastian Bach: Chorale 320, "Gott sei uns gnädig"

The second example that we are going to analyze is the Chorale 320 "Gott sei uns gnädig" of J.S. Bach, whose sequence of chords is the following:

F♯m E A C♯7 D E A F♯m Bm E A A E B E Bm F♯m C♯ F♯m F♯7 Bm F♯m

When we analyze the sequence using the modulation detection algorithm, we notice that we can divide the piece in three parts, as we can see in Figure 4.3:

- A sequence in the key of A, represented in green.
- A sequence in the key of F♯, represented in purple.
- A passing modulation if the key of E, that links the A fragment with the F♯ fragment, represented in orange.

G:	G	D	G	D	G	C	F♯°	G	D	G
	I	V	I	V	I	IV	VII	I	V	I

D:	G	C♯°	D	A	D	A	D	Em	E	A	D	A	D	A	D	B	Em	A
	IV	VII	I	V	I	V	I	II	V^V	V	I	V	I	V	I	V^{II}	II	V

Table 5.1: Division of the Eine kleine Nachtmusik's fragment into two non-modulating sequences and change of notation. The pivot chord is represented in red.

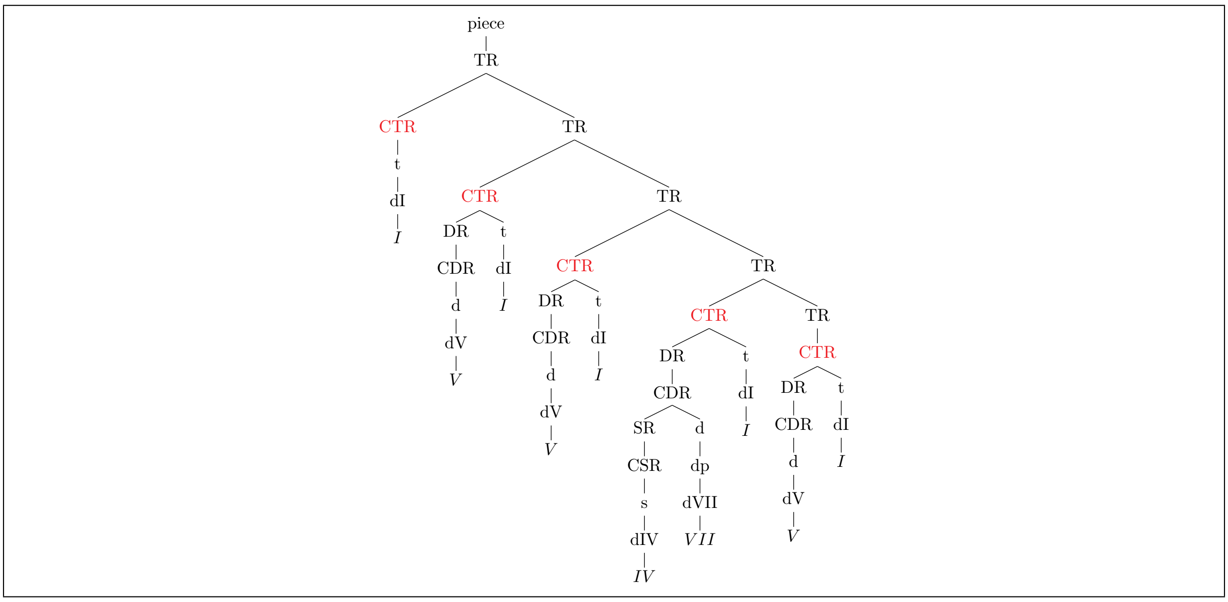


Figure 5.3: G sequence of Eine kleine Nachtmusik. Parse tree.

Using the results of a window size equal to 6 and selecting the pivot chord as described in previous sections, we obtain the following non-modulating sequences (note that the pivot chords are in the two sequences that they link):

- Key of A: *F♯m E A C♯7 D E A F♯m Bm E A A E B*
- Key of E: *B E Bm*
- Key of F♯: *Bm F♯m C♯ F♯m F♯7 Bm F♯m*

The E sequence is too small to be analyzed. It is an harmonic ambiguous passage that acts as a bridge between the A and F♯ sequences rather than a modulation. When we analyze the other two sequences we obtain a syntax error. This happens because the modulation is in the middle of a musical phrase, so the harmony is "incomplete". But the ambiguous passage does not have any theoretical borders, so, exploiting the ambiguity of music, we regroup the chords removing some of them from the A and F♯ sequences and adding them to the E one:

- Key of A: *F♯m E A C♯7 D E A F♯m Bm E A A E*
- Key of E: *E B E Bm F♯m*
- Key of F♯: *F♯m C♯ F♯m F♯7 Bm F♯m*

Now we can analyze the A and F♯ sequences, obtaining the parse trees of Figure 5.5. We can see in the A sequence parse tree in Figure 5.5a that the chords are grouped in tonic regions. There is also an example of a secondary dominant irregular resolution (V^{VI} IV). In the F♯ sequence parse tree in Figure 5.5b we can see an example of the progression I IV I with a secondary dominant. It is important to note the difference between the chord IV of the first sequence and the chord IV of the second one. In the key A sequence, the chord IV has the function of subdominant (it is under a subdominant region in Figure

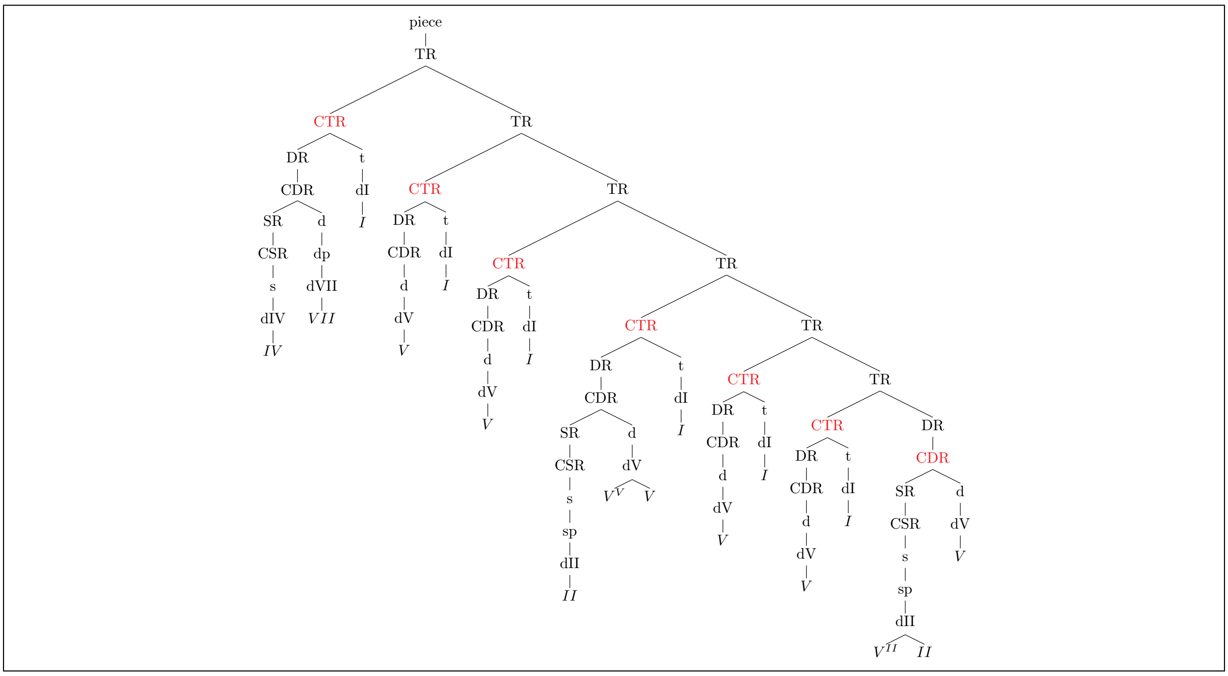
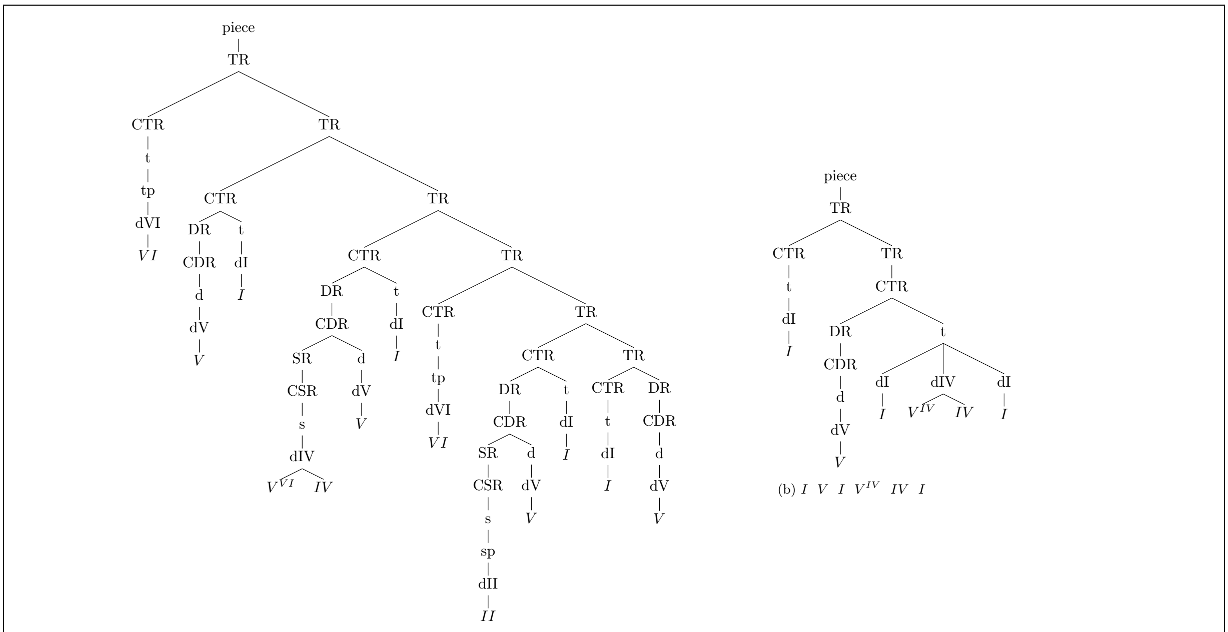


Figure 5.4: D sequence of Eine kleine Nachtmusik. Parse tree.

5.5a), whereas in the key F♯ one, it has a tonic function (it is in the tonic sequence $I \quad IV \quad I$ under the tonic region in Figure 5.5b). So the syntax analysis of chord sequences using the grammar built above allows us to know the functions of each chord of the sequence and the relationships between them.



(a) $VI \;\; V \;\; I \;\; V^{VI} \;\; IV \;\; V \;\; I \;\; VI \;\; II \;\; V \;\; I \;\; I \;\; V$

(b) $I \;\; V \;\; I \;\; V^{IV} \;\; IV \;\; I$

Figure 5.5: J.S. Bach, Chorale 320 "Gott sei uns gnädig". Parse trees.

6

Conclusions and future work

6.1 Conclusions

This work has focused on the automated harmonic analysis of musical fragments. More specifically, we have analyzed pieces of music from the classical tonal music of the XVII and XVIII century. The main reason for restricting the musical scope is that, in order to deeply analyze musical fragments, we need to take into consideration the harmonic structures of the corresponding music style, which differ from one style to another. A generic analysis of any musical fragment is less interesting as we will obtain less information about its harmonic structures. Our analytical approach has been created in the context of tonal music, so it might not work as well with other styles. For example, the modulation algorithm does not work with modal music, as modal music is not organized around tonalities but around modes (a type of musical scale with a set of characteristic melodic behaviors) [5].

In this project, we added new productions to the grammar from [18] in order to expand the set of harmonic sequences that it can analyze. Our objective was to try to make the grammar as complete as possible, in other words, that the grammar could analyze as many harmonic sequences from the target music style as possible. But sometimes, when we tried to add productions that capture a new relation between two or more chords, the grammar became ambiguous. Other times, we could not analyze a particular concept using context-free grammars, especially if the concept depended on the context. That was the case of modulation, so we separated it from the grammar and we implemented a numeric cost-based method to detect the modulations of chord sequences. Once the modulations were detected, we analyzed each non-modulating part of the sequence using the grammar.

In essence, the depth of the analysis depends on the limitations of the technology that is been used: in this case, the context-free grammars. Due to these limitations, it was not possible to expand the grammar to formalize the entire classical music. But, as future work, we can use the grammar for the harmonic characteristics that can be analyzed using it and other approaches (as we did with the modulation) for the concepts that the grammar cannot resolve.

6.2 Future work

We have already talked about some future work in Section 3.6 (Extensions and Limitations of the grammar). Some unresolved conflicts related with the grammar approach were the uncommon subdominant progressions, the generalization of the secondary dominant resolutions, the identification and analysis of "paso chords" and the detection and analysis of harmonic and descending fifths sequences. The grammar can be further expanded and improved in order to consider more specific scenarios, taking always into account the limitations of such extensions. However, most of the future work stated before came from the limitations of the formal grammars themselves, meaning that some of these conflicts cannot be resolved just using the properties of formal grammars. So, this project opens an avenue of research into different approaches to resolve the limitations, working together with the grammar for a more complete analysis. For example, machine learning and deep learning are very promising approaches to formalize music. As our objective was to determine the structure of harmonies, we selected formal grammars as the main technology for this work, as they lead naturally to modeling hierarchies. But we think that machine learning can contribute a lot in this area working along formal grammars.

It is possible that the "mixed" approach that we have developed in this work may be extended to the analysis of descending fifths sequences (Section 3.2.3, p. 13) and harmonic sequences (Section 3.6, p. 21). Just like modulation, descending fifths and harmonic sequences are not a precise concept that can be recognized unambiguously by a grammar - if nothing else, these sequences must be of a minimum length, which entails either a grammar with counting capabilities (taking us outside of the realm of context-free grammars) or a grammar with such a complex structure that it would cloud, rather than revealing, the underlying structure of the harmony. Instead, we can recognize a descending fifths or harmonic sequence following the same idea of modulation: by building up evidence for the various competing hypotheses and adopting the solution best supported by the evidence. Once the sequence has been divided into unambiguous parts, grammars can be used to analyze them. We regard this as a promising future development for this work.

With regard to the modulation detection algorithm, in this work we have chosen for each key a set of the most representative chords. As future work, a further study of all the possible chords used within a key and their level of importance can be done, in order to complete these sets and have a more accurate algorithm. Pivot chords of modulations are another cloudy concept of musical harmony. In Section 4.3.2 we presented a simple, yet satisfactorily-accurate, method for determining them. As pivot chords depend heavily on the harmonic context, cost-based approaches can be investigated in order to develop a more accurate method for approximating them.

In this work, the combination of two different approaches for the purpose of formalizing harmony: formal grammars and numeric cost-based methods, has given very good results. In fact, the combination of different methods and technologies appears to be a very bright perspective for analyzing musical harmonies. And not only for harmonic analysis, in general, for analysis and synthesis of music, the restriction to a method can limit the results whereas the combination of various methods could help to approach and resolve the several problems encountered more successfully.

Glossary

- **change of mode**: Change of the key from major to minor or vice versa.
- **chord**: Set of notes that are heard simultaneously.
- **dominant**: The fifth (V) chord of a key. Usually gives a feeling of suspension.
- **GLR parser**: Extension of LR parser that handles ambiguous and nondeterministic grammars.
- **harmony**: One of the main elements of music. The study of chords and how to arrange them.
- **irregular resolution**: Less common resolutions, usually to a chord other than the tonic.
- **LR parser**: A class of bottom-up parser to read deterministic context-free languages.
- **modal music**: Music organized around modes (a type of musical scale).
- **mode**: Whether a key is major or minor.
- **mode mixture**: Use of chords belonging to any mode of a key indistinctly.
- **modulation**: Process of changing the key within a musical fragment.
- **pivot chord**: Chord common to two keys used to change from one key to another.
- **regular resolution**: A common resolution, usually to the tonic chord.
- **resolution**: Movement of a note or a chord from an unstable sound to another note or chord with a more final or stable sound.
- **secondary dominant**: Borrowed chords from other keys that are the dominants of a scale degree of the actual key and usually resolve in them.
- **tonal center**: Key that the closer chords from a certain point are related to. Its variation determines modulations, tonicizations and change of modes.
- **tonal music**: Music organized around tonalities.
- **tonality**: Arrangement of a determined set of notes and chords with different functions and relations.
- **tonicization**: Treatment of a key as a temporary key in a musical fragment.
- **key**: Tonality.

Bibliography

[1] Alfred V Aho, Ravi Sethi, and Jeffrey D Ullman. *Compilers: Principles, Tools, and Techniques.* Addison-Wesley, 1986.

[2] Charles Ames. Automated composition in retrospect: 1956-1986. *Leonardo*, 20(2):169–185, 1987.

[3] Charles Ames. The markov process as a compositional model: A survey and tutorial. *Leonardo*, 22(2):175–187, 1989.

[4] Allan G Bromley. Charles babbage's analytical engine, 1838. *Annals of the History of Computing*, 4(3):196–217, 1982.

[5] Stephen M. Cormier. *Modal Music Composition.* Inman & Artz Publishers, third edition, 2010.

[6] Alfonso Ortega de la Puente, Rafael Sánchez Alfonso, and Manuel Alfonseca Moreno. Automatic composition of music by means of grammatical evolution. In *Proceedings of the 2002 conference on APL: array processing languages: lore, problems, and applications*, volume 32, pages 148–155. ACM, 2002.

[7] Jon Gillick, Kevin Tang, and Robert M Keller. Machine learning of jazz grammars. *Computer Music Journal*, 34(3):56–66, 2010.

[8] SR Holtzman. Using generative grammars for music composition. *Computer Music Journal*, 5(1):51–64, 1981.

[9] Rowan Hooper. Ada lovelace: My brain is more than merely mortal. *New Scientist*, 216(2886):29, 2012.

[10] Robert M Keller and David R Morrison. A grammatical approach to automatic improvisation. In *Proceedings of the 4th Sound and Music Computing Conference*, Lefkada, Greece, July 2007.

[11] John Levine. *Flex & Bison: Text Processing Tools.* O'Reilly Media, Inc., 2009.

[12] Jon McCormack. Grammar based music composition. *Complex systems*, 96:321–336, 1996.

[13] Luigi Federico Menabrea and Ada Lovelace. Sketch of the analytical engine invented by charles babbage, 1842.

[14] George Papadopoulos and Geraint Wiggins. AI methods for algorithmic composition: A survey, a critical view and future prospects. In *AISB Symposium on Musical Creativity*, volume 124, pages 110–117. Edinburgh, UK, 1999.

[15] Salim Perchy and Gerardo Sarria. Musical composition with stochastic context-free grammars. In *8th Mexican International Conference on Artificial Intelligence (MICAI 2009)*, 2009.

[16] Walter Piston. *Harmony.* W W Norton & co., fifth edition, March 1987. Trad. Esp. Walter Piston. Armonía. Idea Books, 2001.

[17] Martin Rohrmeier. A generative grammar approach to diatonic harmonic structure. In *Proceedings of the 4th sound and music computing conference*, pages 97–100, Lefkada, Greece, July 2007.

[18] Martin Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53, 2011.

[19] Mark J Steedman. A generative grammar for jazz chord sequences. *Music Perception: An Interdisciplinary Journal*, 2(1):52–77, 1984.

A

A few notes on musical theory

In this Annex we explain briefly some foundational musical concepts that are necessary to fully understand the project. This is but a brief overview of some of the numerous notions of music, and will be limited to the point of view of the tonal music of the XVII and XVIII centuries. We recommend Walter Piston's book *Harmony* [16] for a more extended explanation of these concepts.

Music is the art of organizing in a coherent way a combination of sounds and silences under the fundamental principles of melody, harmony and rhythm. A *musical note* is a sound determined by a vibration whose fundamental frequency is constant. That fundamental frequency determines the pitch of the note and its harmonics determine the timbre.

There are notes that sound similar: they have fundamental frequencies in a ratio equal to any integer power of two, and we group them under the same pitch class. Using the temperament tuning, we differentiate between twelve pitch classes that name the different notes: C, $C\sharp$ or $D\flat$, D, $D\sharp$ or $E\flat$, E, F, $F\sharp$ or $G\flat$, G, $G\sharp$ or $A\flat$, A, $A\sharp$ or $B\flat$ and B. In figure A.1 the twelve notes are represented using seven positions in the staff and accidentals: *sharps* ($\sharp$) and *flats* ($\flat$), which modify the pitch of the notes.

A note can have different names: for example the note $C\sharp$ and $D\flat$ are identical in pitch. However, they have different musical meaning. This is called *enharmonic equivalence*. In Figure A.1, each note of the first staff and the corresponding one on the second staff are enharmonically the same.

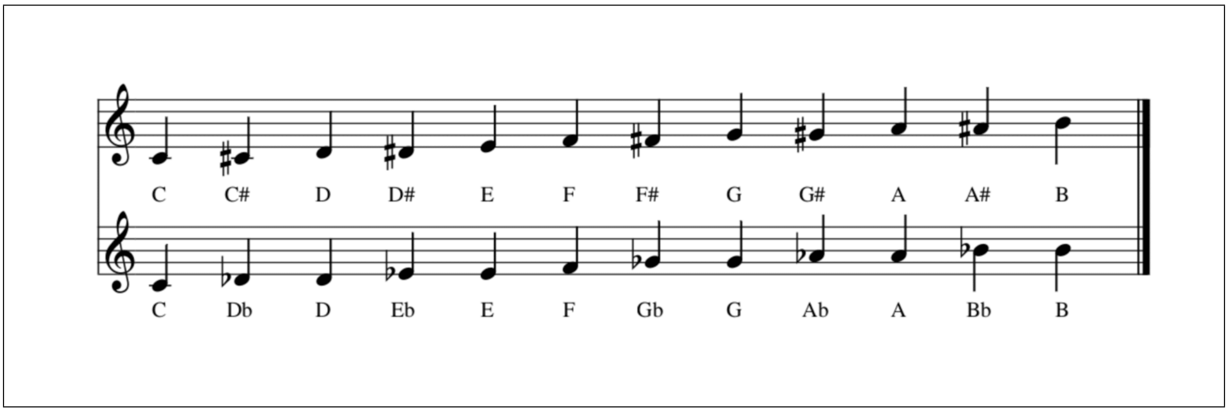


Figure A.1: Notes representing the twelve pitch classes. The notes in the same column are the same or enharmonically equivalent.

The unit of harmony is the *interval* [16], which is defined as the distance between the pitches of two notes. In figure A.1 the distance between one note and the adjacent ones is a *semitone* and two semitones form a *tone*. For example, the distance between C and $C\sharp$ is a semitone, and the distance between E and G is three semitones or one tone and one semitone.

Scales are ordered sets of musical notes. There are many types of scales used in different music styles. The tonal music of the XVII and XVIII centuries is based in what is called the *diatonic scales*, with seven notes each. The seven notes of a diatonic scale are called the scale degrees and it is customary to refer to them by Roman numerals. They have the following names:

- I: Tonic
- II: Supertonic
- III: Mediant
- IV: Subdominant
- V: Dominant
- VI: Submediant
- VII: Leading-tone

For each note, there are five diatonic scales based on that note: one major and four minor, which are represented in Figure A.2 for the note C. The scales determine the *tonalities* or *keys*: the scales of Figure A.2 start with the note C, so they are the scales of the tonality of C major and C minor. To obtain the scales of the rest of the tonalities we only have to translate all the notes of the C scales. For example, if we translate all the notes two semitones, we obtain the scales of D major and D minor. As there are twelve notes, we can obtain twelve major tonalities and twelve minor tonalities.

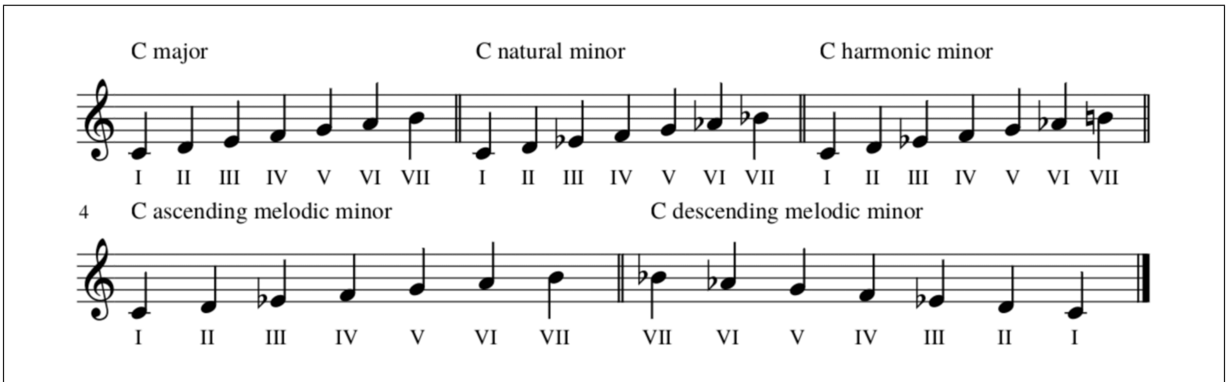


Figure A.2: Diatonic scales.

A *chord* is a set of notes (usually three or four) that are heard simultaneously. A chord is formed using one note (which is called the *fundamental*) and adding other notes at a determined tonal distance. There are different types of chords depending on the intervals between their notes. We are going to take into consideration major, minor, augmented and diminished chords. The English notation of chords is the name of the fundamental pitch of the chord followed by the type of the chord. For example, the different chords with C as the fundamental note are:

- C major chord: C
- C minor chord: Cm
- C augmented chord: $C+$
- C diminished chord: C^o

These notations assume that there are three different notes in each chord. If there are a fourth one we add additional symbols to the notation. For example, a C chord with a major seventh (a note at a distance of five tones and one semitone) is represented as: $C7$.

Each note of the scales of Figure A.2 can be used as the fundamental note of a chord. In this way, we can obtain the chords of a tonality. In this case, using the diatonic scales of C, we obtain the chords of the tonality of C, represented in Figure A.3. We can see that there are two different ways of naming the chords of Figure A.2: using the English nomenclature explained below or using the Roman numerals. For example, the chord F and the IV scale degree of C major are the same chord. It is important to note that, given a chord in the English nomenclature and a tonality, we can determine the scale degree of that chord (if the chord belongs to that tonality).

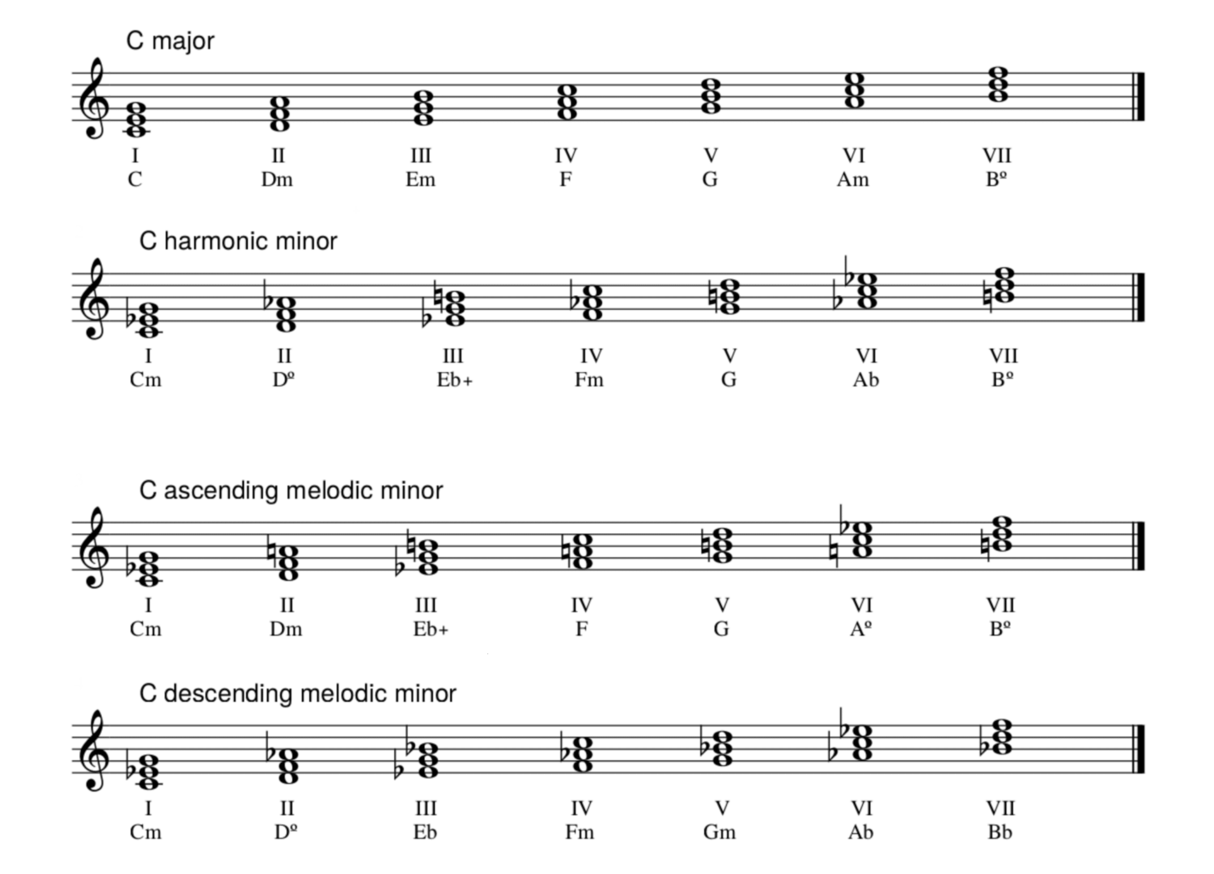


Figure A.3: C chords formed from the diatonic scales.

These are the chords more directly linked to the tonality of C. Likewise, we can obtain the chords of the rest of the tonalities using their corresponding diatonic scales. When we are writing a musical piece in the tonality of C we usually use the chords of C, represented in Figure A.3. If we are using C chords we say that the *tonal center* of the fragment is C and all chords are dependent on the scale degree I of C (the chord C). However, the tonal center is not a stationary property and it can slightly vary without leaving the tonality of C. This happens when one of the scales degrees (different from I) is the new tonal center for a brief period of time. In that period of time we use the chords from the diatonic scales in which that scale degree of C is the I in the new tonal center. For example, if the tonal center is C, we can have the scale degree V of C (G) as the new tonal center for a brief period of time using, for example, its dominant (D). The most common chords used when the tonal center varies are the dominants of the chords of the main tonality. These are called *secondary dominants* and are the following:

- V^{II}: the dominant of the scale degree II
- V^{III}: the dominant of the scale degree III
- V^{IV}: the dominant of the scale degree IV
- V^{V}: the dominant of the scale degree V
- V^{VI}: the dominant of the scale degree VI

- V^{VII}: the dominant of the scale degree VII

It is important to note that, when the scale degree is an augmented or a diminished chord, it cannot determine a new tonal center, so the corresponding secondary dominant cannot be defined.

Aside from the chords from the diatonic scales, there are other chords that also belong to a tonality. These chords are formed using the diatonic scale chords. In this project we consider the Neapolitan sixth chord and the dominant seventh chords, although there are more of them. The Neapolitan sixth is a special chord formed from the second degree chord and it is represented as $\flat II$. The dominant seventh chords are formed adding an additional note at a distance of five tones and one semitone to the dominant chord.

In Annex B, we have created for each tonality a table with the most common chords. These tables link the English nomenclature with the Roman numerals one. The chords that are included are: the chords formed from the diatonic scales, the secondary dominants, the dominant seventh chords and the Neapolitan sixth chord.

B

Tables for the modulation algorithm

In this Annex we present for each tonality the different chords that are closer to it and their associated weight.

The first row represents the scale degrees (I, II, III, IV, V, VI and VII) and the Neapolitan sixth chord ($\flat II$). The first two columns are the tonality (distinguishing between the major and the minor mode) and the secondary dominants and diatonic scales (major, ascending melodic, harmonic minor and descending melodic). In the rows of the diatonic scales their corresponding chords are shown. In the secondary dominant rows, the secondary dominant of the chord immediately above is displayed. The Neapolitan sixth chord in independent from the scales and the secondary dominants. All chords are represented using the English nomenclature. A symbol in brackets means that both the chord with and without the symbol are considered. For example, *G(7)* is either *G* or *G7*.

As two notes can be enharmonically equivalent, two chords can be too. When two chords are enharmonic, we display the most common representation of it. When both representation are relevant, we present both of them indicating that they are the same with a "/ ".

For our numerical cost-based method we have assigned a weight for each chord. In the tables, chords with a green background have a weight of 5.0; with an orange background, a weight of 3.0; with a yellow one, a weight of 2.0 and without background, a weight of 1.0. If a chord does not appear in a table, we consider that it has a weight of 0.0 in that tonality.

		I	**II**	**III**	**IV**	**V**	**VI**	**VII**	**♭II**
C (B♯)	**Major**	C	Dm	Em	F	G(7)	Am	B°	D♭
	Secondary dominants	-	A(7)	B(7)	C7	D(7)	E(7)	-	
Cm	**Ascending Melodic**	-	Dm	-	F	-	-	-	
	Harmonic minor	Cm	D°	-	Fm	G(7)	A♭	B°	
	Descending Melodic	-	-	E♭	-	Gm	-	B♭	
	Secondary dominants	-	-	B♭7	C7	D(7)	E♭7	F7	

Table B.1: Key of C

		I	II	III	IV	V	VI	VII	♭II
G (A♭♭)	**Major**	G	Am	Bm	C	D(7)	Em	F♯°	A♭
	Secondary dominants	-	E(7)	F♯(7)	G7	A(7)	B(7)	-	
Gm	**Ascending Melodic**	-	Am	-	C	-	-	-	
	Harmonic minor	Gm	A°	-	Cm	D(7)	E♭	F♯°	
	Descending Melodic	-	-	B♭	-	Dm	-	F	
	Secondary dominants	-	-	F7	G7	A(7)	B♭7	C7	

Table B.2: Key of G

		I	II	III	IV	V	VI	VII	♭II
D (E♭♭)	**Major**	D	Em	F♯m	G	A(7)	Bm	C♯°	E♭
	Secondary dominants	-	B(7)	C♯(7)	D7	E(7)	F♯(7)	-	
Dm	**Ascending Melodic**	-	Em	-	G	-	-	-	
	Harmonic minor	Dm	E°	-	Gm	A(7)	B♭	C♯°	
	Descending Melodic	-	-	F	-	Am	-	C	
	Secondary dominants	-	-	C7	D7	E(7)	F7	G7	

Table B.3: Key of D

		I	II	III	IV	V	VI	VII	♭II
A (B♭♭)	**Major**	A	Bm	C♯m	D	E(7)	F♯m	G♯°	B♭
	Secondary dominants	-	F♯(7)	G♯(7)	A7	B(7)	C♯(7)	-	
Am	**Ascending Melodic**	-	Bm	-	D	-	-	-	
	Harmonic minor	Am	B°	-	Dm	E(7)	F	G♯°	
	Descending Melodic	-	-	C	-	Em	-	G	
	Secondary dominants	-	-	G7	A7	B(7)	C7	D7	

Table B.4: Key of A

		I	II	III	IV	V	VI	VII	♭II
E (F♭)	**Major**	E	F♯m	G♯m	A	B(7)	C♯m	D♯°	F
	Secondary dominants	-	C♯(7)	D♯(7)	E7	F♯(7)	G♯(7)	-	
Em	**Ascending Melodic**	-	F♯m	-	A	-	-	-	
	Harmonic minor	Em	F♯°	-	Am	B(7)	C	D♯°	
	Descending Melodic	-	-	G	-	Bm	-	D	
	Secondary dominants	-	-	D7	E7	F♯(7)	G7	A7	

Table B.5: Key of E

		I	II	III	IV	V	VI	VII	♭II
B (C♭)	**Major**	B	C♯m	D♯m	E	F♯(7)	G♯m	A♯°	C
	Secondary dominants	-	G♯(7)	A♯(7)	B7	C♯(7)	D♯(7)	-	
Bm	**Ascending Melodic**	-	C♯m	-	E	-	-	-	
	Harmonic minor	Bm	C♯°	-	Em	F♯(7)	G	A♯°	
	Descending Melodic	-	-	D	-	F♯m	-	A	
	Secondary dominants	-	-	A7	B7	C♯(7)	D7	E7	

Table B.6: Key of B

		I	II	III	IV	V	VI	VII	♭II
F♯ / G♭	**Major**	F♯ / G♭	G♯m / A♭m	A♯m / B♭m	B / C♭	C♯(7) / D♭(7)	D♯m / E♭m	E♯° / F°	G
	Secondary dominants	-	D♯(7) / E♭(7)	E♯(7) / F(7)	F♯7 / G♭7	G♯(7) / A♭(7)	A♯(7) / B♭(7)	-	
F♯m	**Ascending Melodic**	-	G♯m	-	B	-	-	-	
	Harmonic minor	F♯m	G♯°	-	Bm	C♯(7)	D	E♯°	
	Descending Melodic	-	-	A	-	C♯m	-	E	
	Secondary dominants	-	-	E7	F♯7	G♯(7)	A7	B7	

Table B.7: Key of F♯ or G♭

		I	II	III	IV	V	VI	VII	♭II
D♭ (C♯)	**Major**	D♭	E♭m	Fm	G♭	A♭(7)	B♭m	C°	E♭♭ / D
	Secondary dominants	-	B♭(7)	C(7)	D♭7	E♭(7)	F(7)	-	
C♯m	**Ascending Melodic**	-	D♯m	-	F♯	-	-	-	
	Harmonic minor	C♯m	D♯°	-	F♯m	G♯(7)	A	B♯°	
	Descending Melodic	-	-	E	-	G♯m	-	B	
	Secondary dominants	-	-	B7	C♯7	D♯(7)	E7	F♯7	

Table B.8: Key of D♭ or C♯

		I	II	III	IV	V	VI	VII	♭II
A♭ (G♯)	**Major**	A♭	B♭m	Cm	D♭	E♭(7)	Fm	G°	B♭♭ / A
	Secondary dominants	-	F(7)	G(7)	A♭7	B♭(7)	C(7)	-	
G♯m (A♭m)	**Ascending Melodic**	-	A♯m	-	C♯	-	-	-	
	Harmonic minor	G♯m	A♯°	-	C♯m	D♯(7)	E	Fx° / G°	
	Descending Melodic	-	-	B	-	D♯m	-	F♯	
	Secondary dominants	-	-	F♯7	G♯7	A♯(7)	B7	C♯7	

Table B.9: Key of A♭ or G♯

		I	II	III	IV	V	VI	VII	♭II
E♭ / (D♯)	**Major**	E♭	Fm	Gm	A♭	B♭(7)	Cm	D°	F♭ / E
	Secondary dominants	-	C(7)	D(7)	E♭7	F(7)	G(7)	-	
D♯m / E♭m	**Ascending Melodic**	-	E♯m / Fm	-	G♯ / A♭	-	-	-	
	Harmonic minor	D♯m / E♭m	E♯° / F°	-	G♯m / A♭m	A♯(7) / B♭(7)	B / C♭	Cx° / D°	
	Descending Melodic	-	-	F♯ / G♭	-	A♯m / B♭m	-	C♯ / D♭	
	Secondary dominants	-	-	C♯7 / D♭7	D♯7 / E♭7	E♯(7) / F(7)	F♯7 / G♭7	G♯7 / A♭7	

Table B.10: Key of E♭ or D♯

		I	II	III	IV	V	VI	VII	♭II
B♭ (A♯)	**Major**	B♭	Cm	Dm	E♭	F(7)	Gm	A°	C♭ / B
	Secondary dominants	-	G(7)	A(7)	B♭7	C(7)	D(7)	-	
B♭m (A♯m)	**Ascending Melodic**	-	Cm	-	E♭	-	-	-	
	Harmonic minor	B♭m	C°	-	E♭m	F(7)	G♭	A°	
	Descending Melodic	-	-	D♭	-	Fm	-	A♭	
	Secondary dominants	-	-	A♭7	B♭7	C(7)	D♭7	E♭7	

Table B.11: Key of B♭

		I	II	III	IV	V	VI	VII	♭II
F (E♯)	**Major**	F	Gm	Am	B♭	C(7)	Dm	E°	G♭
	Secondary dominants	-	D(7)	E(7)	F7	G(7)	A(7)	-	
Fm	**Ascending Melodic**	-	Gm	-	B♭	-	-	-	
	Harmonic minor	Fm	G°	-	B♭m	C(7)	D♭	E°	
	Descending Melodic	-	-	A♭	-	Cm	-	E♭	
	Secondary dominants	-	-	E♭7	F7	G(7)	A♭7	B♭7	

Table B.12: Key of F