Musical Harmony Analysis with Description Logics

Spyridon Kantarelis[0000−0001−8034−8047], Edmund Dervakos[0000−0001−7838−3919], Natalia Kotsani[0000−0002−6369−7402], and Giorgos Stamou[0000−0003−1210−9874]

National Technical University of Athens

Abstract. Symbolic representations of music are emerging as an important data domain both for the music industry, and for computer science research, aiding in the organization of large collections of music and facilitating the development of creative and interactive AI. An aspect of symbolic representations of music, which differentiates it from audio representations, is their suitability to be linked with notions from music theory which have been developed over the centuries. One core such notion is that of functional harmony, which involves analyzing progressions of chords. This paper proposes a description of the theory of functional harmony within the OWL 2 RL profile and experimentally demonstrates its practical use.

Keywords: description logics · ontology engineering · SPARQL · music information retrieval · functional musical harmony · music theory.

1 Introduction

Music in general is complex and can be analyzed in many different ways, across different levels of abstraction, and regarding different aspects of it. Algorithmic analysis of music is increasingly determining how music is consumed –for instance through recommender systems of various streaming platforms, and how music is produced –for instance using plugins which implement various music analysis algorithms within digital audio workstations (DAWs). Music information retrieval (MIR) and automatic music analysis on a large scale, mostly rely on signal processing and machine learning (ML) –in the audio domain– and do not make use of symbolic representations. However, there exists an abundance of symbolic music which is openly available on the web and could be utilized, in conjunction with knowledge representation technologies, for analyzing music.

A particular aspect of music which may be analyzed based on symbolic representations is that of harmony. In music, harmony relates to the way humans perceive superpositions of sounds –for instance chords– and how sequences of such superpositions –such as chord progressions– may be used within a composition in order to serve the underlying music and the emotions it conveys. There are multiple different theories of harmony, across different cultures, which however share some common features. Many approaches to harmonic analysis for instance focus on the ideas of building tension and resolution by the appropriate use of chords or pitches within a piece of music. One of the most widely used such frameworks for analyzing harmony, especially in western culture, is that of tonal harmony [12], where given a chord progression each chord is assigned one of three general functions –depending on if the chord builds tension or resolution in a given context.

Algorithms for automatic functional harmonic analysis have been proposed in the literature and have successfully been applied in a practical setting. An important example of such work is HarmTrace [5,6] which is used by platforms such as Chordify [16]. HarmTrace defines a language through a context free grammar (CFG), based on the formalism proposed by Rohmeier [27]. It is implemented in the Haskell programming language, and is able to produce functional harmonic analyses which are compliant with western tonal harmony, when given a symbolic (text) representation of a chord progression, in addition to a key signature. We perform an extensive comparison of HarmTrace and our semantic web based approach in the evaluation section of this paper.

In the proposed approach, we represent the theory of tonal harmony, modal harmony and functional harmonic analysis using Description Logic (DL) languages. DL languages enrich knowledge representation systems, in terms of the language expressivity for describing the knowledge base, the reasoning techniques and the effectiveness of the interaction between the user and the system [20]. DLs allow for implementation-independent precise specification of both the provided functionality and the inferences. The "TBox" and "ABox" of an ontology accomplish a clear distinction between the intensional knowledge, or general knowledge about the problem domain, and the extensional knowledge.

By representing music theory as a TBox we are able to perform analysis of chord progressions via reasoning. We have identified many benefits of this approach when compared to others, namely ease of use –through the utilization of standardized technologies such as the web ontology language OWL 2, extendibility of the framework by either adding axioms to the TBox or by describing different aspects of music theory with description logics, and their ability to be interlinked with other sources of data –since these technologies are intended for use on the semantic web.

The rest of the paper is structured as follows: Section 2 presents a comprehensive list of ontologies and data models which are aimed at representing information related to music. Section 3 describes in detail the translation of tonal and modal harmony into a fragment of description logics which is compatible with the OWL 2 RL [13] profile. In Section 4 we perform an extensive evaluation of the proposed framework and in Section 5 we conclude the paper.

2 Related Ontologies and Data Models

2.1 Symbolic Music Representation

Symbolic representations of music vary in form, information content and application domain. One of the most widely used such representations is the Musical Instrument Digital Interface (MIDI) [28][19][2]. MIDI allows for light-weight communication of musical information between devices and since its inception it has been the *de facto* protocol for representing music symbolically. MIDI may also be losslessly converted to Linked Data [17] which may be used in a variety of ways –such as generating mashups via SPARQL [18]. Another widely used framework for symbolic music representation is MusicXML [10] which is a format proposed for representing sheet music and digital scores. MusicXML's information content is similar to that of MIDI, however it is aimed at rendering human-readable musical scores with information which is relevant to a performer, when compared to MIDI which is aimed at controlling digital instruments. Among the many symbolic representations of music are also text representations, such as ASCII tablatures for fretted instruments, and chord progressions with or without lyrics, which are widely available on the web [1].

2.2 Related Ontologies

Alvaro et al. [1] developed a novel dynamic representation system for musical knowledge adapted to computer-aided composition. Raimond et al. [24] introduce the Music Ontology, including editorial, cultural and acoustic information using the Resource Description Framework, the Timeline[2] and the Event Ontology,[3] and incorporating music production specific concepts. They also divide the ontology in several levels of expressiveness (editorial information, events, event decomposition), in order to allow a large range of granularity. Song et al. [29] present a Context-based Music Recommendation (COMUS) ontology to reason desired user emotion state from context capturing low-level musical features. Fields et al. [9] propose a Segment Ontology for MIR signal processing tasks, modeling musicological properties whilst maintaining an accurate and complete description of their relationships. Fazekas et al. [8] introduce the Studio Ontology Framework specialized in music production, providing an explicit, application and situation independent conceptualisation of the studio environment. Kolozali et al. [15] propose an instrument ontology based on the Hornbostel and Sach's classification scheme, considering terminology and conceptualisation used by ethnomusicologists. Their ontology, based on the Musicbrainz instrument tree and the SKOS instrument taxonomy, use the Ontology Web Language (OWL) and SPARQL queries, which demonstrate real-world use cases for instrument

[1] http://www.ultimate-guitar.com, http://www.guitartabs.cc, http://www.e-chords.com
[2] http://motools.sf.net/timeline/timeline.html
[3] http://motools.sf.net/event/event.html

knowledge representation. Jones et al. [14] developed the MusicOWL Ontology[4] for encoding music scores of Western music. Using MusicOWL Ontology they managed to convert existing music scores into RDF. S. Cherfi et al. [4] introduce the MUSICNOTE Ontology[5], an ontology of music notation that relies on the structural aspects of a score.

Sabbir et al. [25] introduce the Music Theory Ontology, including classes for musical notation, duration, chords, progression and degrees. These are fundamental concepts used for the analysis of music. They limit their focus to Western music concepts. Music Theory Ontology references several namespaces prefixes for relevant Ontologies, such as the Music Ontology and the Chord Ontology[6].

3 Harmonic Analysis with Description Logics

In this section we describe the terminology used from music theory, and we present axioms which may be used to infer the function of a chord in the context of a chord progression [26]. Where applicable we denote corresponding concepts from the music theory ontology with the prefix *mto*, while for concepts from our ontology we use the prefix *fho*.

3.1 Preliminaries

Let $\mathcal{V} = \langle \mathsf{CN}, \mathsf{RN}, \mathsf{IN} \rangle$ be a *vocabulary*, with CN, RN, IN mutually disjoint finite sets of *concept*, *role* and *individual* names, respectively. Let also $\mathcal{T}$ and $\mathcal{A}$ be a terminology (TBox) and an assertional database (ABox), respectively, over $\mathcal{V}$ using a Description Logics (DL) dialect $\mathcal{L}$, i.e. a set of axioms and assertions that use elements of $\mathcal{V}$ and constructors of $\mathcal{L}$. The pair $\langle \mathcal{V}, \mathcal{L} \rangle$ is a *DL-language*, and $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is a *knowledge base* (KB) over this language.

The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. OWL classes correspond to description logic (DL) concepts, OWL properties to DL roles, while individuals are called the same way in both the OWL and the DL terminology.

3.2 Chords and Chord Progressions

In Western culture, music typically uses twelve pitch classes across any number of octaves, the musical notes (*mto:Note*). Harmony concerns itself with sets of the twelve pitch classes in a given context. The context is ambiguous –it relates to both local and global characteristics of a piece of music, however in every case it involves a specific pitch class which is called the tonic note. Humans tend to perceive superpositions of sounds by identifying a fundamental frequency, which in most cases we assume to be the tonic note, and all other frequencies are perceived in relation to the fundamental frequency [7].

[4] http://linkeddata.uni-muenster.de/ontology/musicscore/
[5] http://cedric.cnam.fr/isid/ontologies/files/MusicNote.html
[6] http://purl.org/ontology/chord/

Scales and Modes In music theory, a scale (*mto:Key*) is any set of musical notes ordered by pitch. The root of the scale is its tonic note. In Western music, scales are typically formed by seven different pitches (heptatonic scales). A difference in pitch between two notes is called an interval (*mto:Interval*). The smallest of these intervals is a semitone (*mto:Semitone*). A tone (*mto:Tone*) is an interval composed by two semitones. A diatonic scale is an heptatonic scale (*mto:HeptatonicScale*) structured using five tone (T) intervals and two semitone (S) intervals. The two semitone intervals should be separated by two and three tones. The most commonly used diatonic scale is the major scale (*mto:MajorKey*). The sequence of its intervals is (T-T-S-T-T-T-S). In Western music, there are seven different diatonic scales, which are built by taking a different position of the major scale as their root. They are most commonly known as the Modes of major scale (Ionian/Major, Dorian, Phrygian, Lydian, Mixolydian, Aeolian/Natural Minor, and Locrian). The most commonly used non-diatonic scales are the harmonic minor (*mto:HarmonicMinorScale*) and the melodic minor scale (*mto:MelodicMinorScale*). Modes that are structured using the same seven pitches are called relative modes. Modes that are structured using the same root are called parallel modes. In heptatonic scales, the position each different note holds is called scale degree (*mto:Degree*), usually starting with first for tonic.

Tertian Chords In music theory, a chord (*mto:Chord*) is any set of three or more musical notes that are heard as if sounding simultaneously. In Western music, chords are typically based on thirds known as Tertian Chords. The most frequently used tertian chords are triads (*mto:Triad*). Triads consist of three distinct notes: the root, an interval of a third above the root and an interval of a fifth above the root. Tertian Chords with four or more notes are called extended chords. Extended chords include seventh chords (*mto:SeventhTetrad* –a triad with an interval of a seventh above the root), ninth chords (*mto:NinthPentad* –a seventh chord with an interval of a ninth above the root), eleventh chords (a ninth chord with an interval of an eleventh above the root) and a thirteenth chord (an eleventh chord with an interval of a thirteenth chord above the root).

Other Chords The most commonly used non Tertian chords are the suspended chords (*mto:SuspendedChord*) and the added ninth chords. A suspended chord is a tertian chord whose third interval above the root has been replaced by a second or a fourth interval above the root. An added ninth chord is a triad with an interval of a ninth above the root.

Chord Progressions In musical composition, a series of chords is called a chord progression (*mto:HarmonicProgression*). In Western music, chord progressions are the foundation of harmony. All popular western music genres use chord progressions as the base on which melody and rhythm are built.

3.3 Chord Functions

Tonal Harmony As we mentioned, the tonic note is the root of any mode. It is also its tonic center, meaning that every other pitch is perceived in relation to it. In tonal harmony, a chord can serve three different functions: a tonic function, a dominant function or a subdominant function. A tonic chord gives us the acoustic impression that we are in the tonic center. A dominant chord creates an acoustic instability that most frequently leads to a tonic chord (resolution). A subdominant chord has an abstract function. Whilst it is not the tonic center, it is not as unstable as a dominant chord. It can lead to a tonic chord or prepare a dominant chord. The rules of tonal harmony apply to the major scale (Ionian Mode) and the minor scales (Harmonic, Melodic and Natural Minor Mode). In the major scale, the tonic chord is the chord of first degree (*fho:IonianTonic*) and the relative tonic is the chord of sixth degree (*fho:IonianRelativeTonic*), dominant chords are the chords of fifth and seventh degree (*fho:IonianDominant*,*fho:IonianSeventhDominant*) and subdominant chords are the chords of fourth and second degree (*fho:IonianSubdominant*, *fho:IonianRelativeSubdominant*). Chords of third degree can serve as a relative tonic or a relative dominant (*fho:IonianRelativeDominant*). In the minor scales, the tonic chord is the chord of first degree (*fho:AeolianTonic*) and the relative tonic is the chord of third degree (*fho:AeolianRelativeTonic*), dominant chords are the chords of fifth and seventh degree (*fho:AeolianDominant*, *fho:AeolianRelativeDominant*) and subdominant chords are the chords of fourth and sixth degree (*fho:AeolianSubdominant*, *fho:AeolianRelativeSubdominant*). We extend these rules to all modes depending on the type of their first degree chord.

Modal Harmony Modal Harmony [23] uses all Modes of the major scale. Chords align in function with their tonal counterparts, although the additional modal inflections create greater overlap between these functions. For example, a chord of fifth degree can serve as a dominant chord to any tonic chord of a parallel mode. We can perceive tonal harmony as part of modal harmony.

3.4 Axioms for Functional Analysis

Chords At the top level of our hierarchy is the $\mathsf{fho{:}Chord} \equiv \mathsf{mto{:}Chord}$ class . There are direct subclasses for different types of tertian and suspended chords: $\mathsf{AugChord} \sqcup \mathsf{DimChord} \sqcup \mathsf{MajChord} \sqcup \mathsf{MinChord} \sqcup \mathsf{SusChord} \sqsubseteq \mathsf{Chord}$ and for different note cardinalities of chords: $\mathsf{Triad} \sqcup \mathsf{Tetrad} \sqcup \ldots \sqsubseteq \mathsf{Chord}$. There is also a direct subclass for chords which belong to a mode $\mathsf{ModalChord} \sqsubseteq \mathsf{Chord}$, where the class $\mathsf{ModalChord}$ has as direct subclasses, all major and minor scales:

$$\mathsf{CMajorScaleChord} \sqcup \mathsf{CsMajorScaleChord} \sqcup \ldots \sqsubseteq \mathsf{ModalChord}$$

and each major scale chord class has as direct subclasses each of the seven corresponding modes:

$$\mathsf{CIonianChord} \sqcup \mathsf{DDorianChord} \sqcup \ldots \sqsubseteq \mathsf{CMajorScaleChord}$$

In addition, each of these classes which correspond to the chords of a specific mode has as direct subclasses the scale degrees of the mode:

$$\mathsf{CIonianFirst} \sqcup \mathsf{CIonianSecond} \sqcup \ldots \sqsubseteq \mathsf{CIonianChord}$$

Finally, each scale degree has as direct subclasses, all tertian and suspended chords which belong to the specific scale degree of the specific mode:

$$\mathsf{C} \sqcup \mathsf{Csus4} \sqcup \mathsf{Cmaj7} \sqcup \ldots \sqsubseteq \mathsf{CIonianFirst}$$

The type of each chord (Major, Minor, Diminished, Augmented, Suspended), in addition to the number of notes it consists of are also asserted in the ontology:

$$\mathsf{Cmaj9} \sqsubseteq \mathsf{Pentad} \sqcap \mathsf{MajorChord}$$

Tonalities At the top level of our hierarchy is also the Tonality class. Its subclasses are all the different modes:

$$\mathsf{CIonianTonality} \sqcup \mathsf{CsIonianTonality} \sqcup \mathsf{DDorianTonality} \sqcup \ldots \sqsubseteq \mathsf{Tonality}$$

Each Tonality has its Tonic chord. To describe this concept, we define the object property $\mathsf{hasTonality}$ with a domain of Chord and range of $\mathsf{Tonality}$:

$$\mathsf{CIonianFirst} \sqcap \exists \mathsf{hasTonality}.\mathsf{CIonianTonality} \sqsubseteq \mathsf{CIonianTonic}$$

All Tonic chords are subclasses of the class Tonic.

Chord Functions At the top level of our hierarchy is also the FunctionChord class. Among its subclasses are the three harmonic functions:

$$\mathsf{TonicChord} \sqcup \mathsf{DominantChord} \sqcup \mathsf{SubdominantChord} \sqsubseteq \mathsf{FunctionChord}$$

Each of these classes has nine subclasses, one for each mode:

$$\mathsf{IonianDominant} \sqcup \mathsf{DorianDominant} \sqcup \mathsf{LocrianDominant} \sqcup \ldots \sqsubseteq \mathsf{DominantChord}$$

and each of these classes has twelve subclasses, one for each pitch class:

$$\mathsf{AIonianDominant} \sqcup \mathsf{AsIonianDominant} \sqcup \mathsf{BIonianDominant} \sqcup \ldots \sqsubseteq \mathsf{IonianDominant}$$

Chord Progressions and Axioms For representing chord progressions we define the object properties $\mathsf{hasNext} \equiv \mathsf{hasPrevious}^-$ with a domain and range of Chord. Now we can add axioms to each $\mathsf{FunctionChord}$, based on the functions of the scale degree of each mode as we have described, for example:

$$\mathsf{CIonianFifth} \sqcap \exists \mathsf{hasNext}.\mathsf{CIonianFirst} \sqsubseteq \mathsf{CIonianDominant}$$

Parallel Functions In order to use concepts of the Modal Harmony, we first introduce the ParallelTonic class fho:ParallelTonic. Its direct subclasses are the Tonic chords of every different pitch:

$$\mathsf{CParallelTonic} \sqcup \mathsf{CsParallelTonic} \sqcup \mathsf{DParallelTonic} \sqcup \ldots \sqsubseteq \mathsf{ParallelTonic}$$

Each of these subclasses has its direct subclasses:

$$\mathsf{CIonianTonic} \sqcup \mathsf{CDorianTonic} \sqcup \mathsf{CPhrygianTonic} \sqcup \ldots \sqsubseteq \mathsf{CParallelTonic}$$

Now we can extend the chord functions by adding axioms based on the Modal harmony, for example:

$$\mathsf{CIonianFifth} \sqcap \exists \mathsf{hasNext.CParallelTonic} \sqsubseteq \mathsf{CParallelDominant}$$

Local Functions In a music piece, the tonic center can change. Any chord can serve as a tonic center depending on its relationship with its adjacent chords. For this reason, we introduce the classes LocalTonic, LocalDominant and LocalSubdominant, where:

$$\mathsf{Tonic} \sqcap \exists \mathsf{hasPrevious.ParallelDominant} \sqsubseteq \mathsf{LocalTonic}$$

$$\mathsf{DominantChord} \sqcap \exists \mathsf{hasNext.LocalTonic} \sqsubseteq \mathsf{LocalDominant}$$

$$\mathsf{DominantChord} \sqcap \exists \mathsf{hasNext.LocalDominant} \sqsubseteq \mathsf{LocalSubdominant}$$

4 Evaluation

4.1 Experiment Setting

For setting up our experiments we first gathered a set of commonly used datasets of music annotated with chords, and represented them them in an RDF knowledge graph. We fed chord progressions for which the key was known through HarmTrace and compared the resulting analyses with those resulting from reasoning on our ontology. We setup a semantic repository where we loaded the ontology (described in section 3) and data (described in section 4.1.2), and then get the answers for a set of SPARQL queries which are described in detail in section 4.2. All experiments were performed on the same Intel Core i7-7500U machine running at 2.90 GHz with 8 GB of installed RAM and a stable internet connection of 4.84 Mbps average upload speed.

Datasets For the evaluation procedure we gathered four datasets from different sources. These are:

– **Isophonics Beatles Reference Annotations** [11]. This dataset contains annotations concerning the whole discography of The Beatles. Among the available information are also chord annotations, given in multiple formats - including RDF (turtle).

- **McGill Billboard Annotations** [3]. This dataset contains multiple annotations (including chords) for 740 distinct songs which have appeared in the Billboard charts from 1958 to 1991. The annotations are available in multiple different formats. We used the tsv style LAB format, and converted each chord progression into RDF.
- **The Weimar Jazz Database**[7] [21][22]. This is an SQL database containing transcriptions and meta-data for a set of 456 Jazz improvisations. From this we only only utilize the *beats* table which contains information relating to chords, and we represent this information in our knowledge graph.
- **Data scraped from hooktheory**.[8] This is MusicXML data used to render the web application, which contains information about the diatonic mode of the progression and the scale degree of each chord. We gathered a set of 743 chord progressions and added them to our knowledge graph, after setting the tonic of the scale to C and converting scale degrees to chords.

Knowledge Graph Representation We represented each chord progression in our dataset as a RDF knowledge graph. Consecutive chords are connected by fho:hasNext edges, and for each chord in the progression we assert its chord type by adding triples of the form ⟨chrd_1⟩ ⟨rdf:Type⟩ ⟨fho:Cmaj9⟩, where we acquire the type of each chord by parsing the strings in each dataset. For our ontology reasoning and SPARQL query answering, we created a repository on GraphDB,[9] where we uploaded our ontology and the RDF files. The ontology was imported successfully in 8 minutes. The time required to import a RDF file varies depending on its size with an estimated average around 10 seconds.

Ontology Expressivity By loading our ontology into Protégé, we can observe several metrics. Our ontology counts 4347 classes and 4 object properties. There are 21131 logical axioms and 21122 class axioms. The Description Logic (DL) expressivity of the ontology is $\mathcal{ALEHIF}$.

4.2 Results

Comparison with HarmTrace HarmTrace performs an automatic functional harmonic analysis based on the rules of Tonal Harmony. Taking as input the chord progression and the key of a musical piece, it generates a parse tree to visualize the function of each chord. It performs deletions and insertions of chords, so every chord depicted on the tree has a function based on the rules of Tonal Harmony. HarmTrace uses only tonic and dominant function to characterize chords or groups of chords. We used HarmTrace to analyze the hooktheory Dataset, which is the only Dataset with known song keys. The analysis took

[7] https://jazzomat.hfm-weimar.de/
[8] https://www.hooktheory.com/
[9] https://graphdb.ontotext.com/

1253 seconds (1.68s per chord progression). HarmTrace performed 3832 deletions (5.25 per chord progression) and 4768 insertions (6.41 per chord progression) in a total of 16883 chords. In this given dataset, HarmTrace found 3264 chords or groups of chords which serve a tonic function and 7604 chords or groups of chords which serve a dominant function. In order to compare our system with HarmTrace, we enriched our ontology with the concepts of SongTonic, SongRelativeTonic, SongDominant and SongRelativeDominant using the object property hasSongTonality, considering that the given dataset provides the tonality of every chord progression. Our system found 2767 chords that serve a tonic function and 7529 chords that serve a dominant function. In Figure 1 we show an example chord progression, along with the analysis generated by HarmTrace and in Table 1 we show the analysis from reasoning on our ontology for the same progression.

Even though our ontology cannot be directly compared with HarmTrace, mainly due to them being based on different theoretical frameworks (tonal vs modal functional harmony) and due to the fact that HarmTrace inserts and deletes chords, the resulting analyses are in most cases identical. A benefit of HarmTrace is the hierarchical representation of analyses which could provide a musician with more insight when compared to our approach. On the other hand, our approach is theoretically more thorough, since tonal harmony is part of modal harmony, in addition to being empirically more usable. We base this claim firstly on the fact that insertions and deletions of chords performed by HarmTrace can be confusing for a user who has not extensively studied the grammar which HarmTrace is based on, and secondly we argue that using description logics along with standardized technologies such as OWL 2, provides more usability and extendability when compared to Haskell.

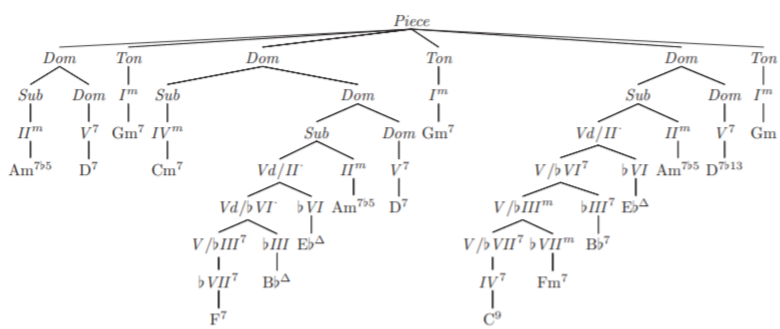

Fig. 1. Parse tree generated by HarmTrace. The key of the piece is G minor. The △ denotes an added major seventh.[5]

Dataset Comparison For this set of experiments we performed SPARQL Queries in order to find differences based on the harmonic structure between

Table 1. Harmonic analysis resulting from reasoning on the proposed ontology. We show two of the inferred types for each chord in the progression.

Chord	Inferred Types	
Am7♭5	LocalSubdominant	D_PhrygianParallelDominant
D7	LocalDominant	G_IonianParallelDominant
Gm7	G_AeolianSongTonic	C_AeolianParallelDominant
Cm7	LocalSubdominant	F_MixolydianParallelDominant
F7	LocalDominant	B_ParallelDominant
B♭△	G_AeolianRelativeTonic	E_LydianParallelDominant
E♭△	LocalTonic	A_LocrianParallelDominant
Am7♭5	LocalSubdominant	D_ParallelDominant
D7	LocalDominant	G_ParallelDominant
Gm7	G_AeolianSongTonic	C_MixolydianDominant
C9	LocalDominant	F_ParallelDominant
Fm7	LocalTonic	B♭_AeolianDominant
B7	LocalSubdominant	E♭_ParallelDominant
E♭7	LocalDominant	A_LocrianDominant
Am7♭5	LocalTonic	-
D7♭13	LocalDominant	G_ParallelDominant
Gm	G_AeolianSongTonic	-

the Jazz Dataset, and the Billboard and Beatles Dataset, which can both be labeled as pop-rock genre. On these datasets, the tonality of each song is not known, so we are not able to define the tonic center. We will focus on the dominant and subdominant functions instead. As we mentioned, a chord with a dominant function is used to build tension, whereas a chord with a subdominant function has a much more peaceful sound when it resolves to its tonic. We can also assume that a chord with a local tonic function, which does not serve a dominant function, offers a brief acoustic stability. The results of our queries are shown in Table 2.

As we can see, in the Jazz genre, over the half of the chords are used to build tension. In contrast, in the pop-rock genre, the dominant and subdominant functions are more equally distributed. This demonstrates that even with simple SPARQL Queries, we are able to find distinct differences between dissimilar music genres based only on the harmonic structure.

Table 2. SPARQL Queries results

	The Weimar Jazz Database	Billboard and Beatles
Total Chords	29122	133411
Dominant Function Chords	15990 (55%)	44939 (34%)
Subdominant Function Chords	3401 (12%)	33850 (25%)
Local Tonic Function Chords	5772 (20%)	18210 (14%)

5 Conclusions and Future Work

Expressing notions from music theory with description logics provides a practical way for semantically enriching datasets of symbolically represented music. This process could significantly aid in the organization of large collections of music, in addition to potentially enhancing performance for various music information retrieval tasks such as genre or mood classification . We plan to explore the impact of utilizing these additional features for machine learning pipelines in future work.

Furthermore, formally representing music theory with description logics could be useful for music education. Learning abstract concepts such as harmony can be challenging for music students. By using knowledge representation we are able to provide countless real world examples of chord patterns, in addition to providing explanations for entailments. One of our future research directions involves working with music educators with the goal of developing a tool, by utilizing music theory knowledge representation, to be used by music students in an educational setting.

Additionally, RDF triples provide a versatile way to represent music information. A direction of future research involves converting MIDI and MusicXML files into RDF using concepts introduced by our ontology in order to perform functional harmonic analysis. Considering that MIDI and MusicXML are the most common tools used for symbolic music representation, this will provide accessibility to a much larger variety of music information as well as make the procedure of functional harmonic analysis easier to the average user.

Finally, as mentioned in the introduction, there are multiple music-theoretical frameworks for describing harmony, especially when taking under consideration different cultures. In addition there are other aspects of music besides harmony, such as rhythm, melody and structure, which have strong theoretical foundations. Thus another direction of our future research involves representing additional notions from music theory with Description Logics.

6 Acknowledgements

This research is carried out / funded in the context of the project "Music Synthesis based on Knowledge Representation, Automated Reasoning and Machine Learning" (MIS 5049188) under the call for proposals "Researchers' support with an emphasis on young researchers-2nd Cycle". The project is co-financed by Greece and the European Union (European Social Fund- ESF) by the Operational Programme Human Resources Development, Education and Lifelong Learning 2014-2020."

References

1. Alvaro, J.L., Miranda, E.R., Barros, B.: Ev: Multilevel music knowlegde representation and programming. Proceedings of SBCM (2005)

2. Association, M.M., et al.: The complete midi 1.0 detailed specification. Los Angeles, CA, The MIDI Manufacturers Association (1996)
3. Burgoyne, J.A., Wild, J., Fujinaga, I.: An expert ground truth set for audio chord recognition and music analysis. In: ISMIR. vol. 11, pp. 633–638 (2011)
4. Cherfi, S.S.s., Guillotel, C., Hamdi, F., Rigaux, P., Travers, N.: Ontology-based annotation of music scores. In: Proceedings of the Knowledge Capture Conference. K-CAP 2017, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3148011.3148038, https://doi.org/10.1145/3148011.3148038
5. De Haas, W.B., Magalhaes, J.P., Wiering, F., Veltkamp, R.C.: Harmtrace: Automatic functional harmonic analysis. Tech. rep., Technical Report UU-CS-2011-023, Department of Information and Computing ... (2011)
6. De Haas, W.B., Rodrigues Magalhães, J., Veltkamp, R.C., Wiering, F.: Harmtrace: Improving harmonic similarity estimation using functional harmony analysis. In: Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR) (2011)
7. Deutsch, D.: Grouping mechanisms in music. In: The psychology of music, pp. 299–348. Elsevier (1999)
8. Fazekas, G., Sandler, M.: The studio ontology framework. pp. 471–476 (01 2011)
9. Fields, B., Page, K., De Roure, D., Crawford, T.: The segment ontology: Bridging music-generic and domain-specific. pp. 1–6 (07 2011). https://doi.org/10.1109/ICME.2011.6012204
10. Good, M., et al.: Musicxml: An internet-friendly format for sheet music. In: Xml conference and expo. pp. 03–04. Citeseer (2001)
11. Harte, C.: Towards automatic extraction of harmony information from music signals. Ph.D. thesis (2010)
12. Hindemith, P.: A Concentrated Course in Traditional Harmony, volume 1. New York: Associated Music Publishers (1943)
13. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S., et al.: Owl 2 web ontology language primer. W3C recommendation **27**(1), 123 (2009)
14. Jones, J., de Siqueira Braga, D., Tertuliano, K., Kauppinen, T.: Musicowl: The music score ontology. In: Proceedings of the International Conference on Web Intelligence. p. 1222–1229. WI '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3106426.3110325, https://doi.org/10.1145/3106426.3110325
15. Kolozali, S., Barthet, M., Fazekas, G., Sandler, M.B.: Knowledge representation issues in musical instrument ontology design. In: ISMIR. pp. 465–470 (2011)
16. Magalhaes, J.P.: Chordify: Three years after the launch (2015)
17. Meroño-Peñuela, A., Hoekstra, R.: The song remains the same: lossless conversion and streaming of midi to rdf and back. In: European Semantic Web Conference. pp. 194–199. Springer (2016)
18. Meroño-Peñuela, A., Meerwaldt, R., Schlobach, S.: Sparql-dj: The midi linked data mashup mixer for your next semantic party. In: International Semantic Web Conference (Posters, Demos & Industry Tracks) (2017)
19. Moog, R.A.: Midi: Musical instrument digital interface. Journal of the Audio Engineering Society **34**(5), 394–404 (1986)
20. Nardi, D., Brachman, R.J., et al.: An introduction to description logics. Description logic handbook **1**, 40 (2003)
21. Pfleiderer, M., Frieler, K.: The jazzomat project. issues and methods for the automatic analysis of jazz improvisations. Concepts, experiments, and fieldwork: Studies in systematic musicology and ethnomusicology pp. 279–295 (2010)

22. Pfleiderer, M., Frieler, K., Abeßer, J., Zaddach, W.G., Burkhart, B.: Inside the jazzomat. New Perspectives for Jazz Research (2017)
23. Piston, W., Devoto, M.: Harmony. W. W. Norton Company (1978)
24. Raimond, Y., Abdallah, S.A., Sandler, M.B., Giasson, F.: The music ontology. In: ISMIR. vol. 2007, p. 8th. Citeseer (2007)
25. Rashid, S.M., De Roure, D., McGuinness, D.L.: A music theory ontology. In: Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music. p. 6–14. SAAM '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3243907.3243913, `https://doi.org/10.1145/3243907.3243913`
26. Riemann, H.: Vereinfachte Harmonielehre oder die Lehre von den tonalen Funktionen der Akkorde. London: Augener Co. (1893)
27. Rohrmeier, M.: Towards a generative syntax of tonal harmony. Journal of Mathematics and Music **5**(1), 35–53 (2011)
28. Smith, D., Wood, C.: The'usi', or universal synthesizer interface. In: Audio Engineering Society Convention 70. Audio Engineering Society (1981)
29. Song, S., Kim, M., Rho, S., Hwang, E.: Music ontology for mood and situation reasoning to support music retrieval and recommendation. In: 2009 Third International Conference on Digital Society. pp. 304–309 (2009)