D7.1
Harmonic training dataset

Authors	Maximos Kaliakatsos-Papakostas, Andreas Katsiavalos, Costas Tsougras, Emilios Cambouropoulos
Reviewers	Allan Smaill, Ewen Maclean, Kai-Uwe Kuhnberger

Grant agreement no.	611553
Project acronym	COINVENT - Concept Invention Theory
Date	October 1, 2014
Distribution	PU

Disclaimer

The project COINVENT acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open Grant number 611553.

Abstract

The COINVENT melodic harmonizer is based on blending harmonic concepts extracted either through statistical learning or from a data pool with "real example/world" representations of historical traditions of music creation. To this end, rich multi-level structural descriptions of harmony of different idioms are necessary, so that meaningful mappings may be made, giving rise to "coherent" harmonic blends. Therefore, a diverse collection of musical pieces drawn from different historic eras and from drastically different harmonic styles has been assembled. Each idiom/style is internally as coherent as possible such that regularities of the specific harmonic space can be extracted; the idioms among them are as different as possible. Additionally, the musical pieces are manually annotated such that structural harmonic features may be extracted at various hierarchic levels. A novel music encoding scheme along with a framework based on musicXML have been developed, allowing this dataset to act as a knowledge repository, where music primitives together with manually annotated analytical descriptions are encoded and extracted according to the requirements of the user/researcher. At a harmonic level above the plain note level, a new idiom-independent representation of chord types that is appropriate for encoding tone simultaneities in any harmonic context (such as tonal, modal, jazz, octatonic, atonal) is proposed and utilized, namely the General Chord Type (GCT).
Keyword list: **Dataset**, **Harmonic training**, **Harmony annotation**, **Harmony encoding**

Contents

1 Introduction

The development of the COINVENT melodic harmonizer incorporates statistical learning and the extraction of harmonic concepts from a data pool with "real example/world" representations of historical traditions of music creation. The COINVENT system should be able to learn different harmonies, and then allow the invention of new harmonic spaces via mechanisms of conceptual blending. Conceptual blending enables structural mapping between diverse harmonic spaces and combination of different harmonic concepts into a novel coherent harmonic system. A melodic harmonisation assistant that facilitates conceptual blending should allow a highly structured representation of harmonic concepts in an explicit manner at various hierarchic levels and parametric viewpoints. Rich multi-level structural descriptions of harmony of different idioms are necessary, so that meaningful mappings may be made, giving rise to "coherent" harmonic blends.

To achieve this goal, a diverse collection of musical pieces drawn from different historic eras and from drastically different harmonic styles has been assembled. Each idiom/style is internally as coherent as possible such that regularities of the specific harmonic space can be extracted; the idioms among them are as different as possible. Additionally, the musical pieces are manually annotated such that structural harmonic features may be extracted at various hierarchic levels, assembling the harmonic spinal chord of the generic space that pairs of diverse idioms (input spaces) share. More specifically, the following structural aspects are manually annotated: a) harmonic reduction(s) of each musical work/excerpt so that structural harmonic/non-harmonic notes are explicitly marked, b) local scale/key changes are determined so that harmonic concepts relating to modulations can be learnt, and c) grouping structure is given so that cadential patterns at various hierarchic levels can be inferred. Although this dataset was created with respect to the formation of a training dataset mainly for harmonic concepts, other important musical aspects (e.g. rhythmic organization, voice movement, etc.) are also taken into account. By selecting an appropriate music representation scheme and developing a framework to extract information, this dataset acts as a knowledge repository, where music primitives together with manually annotated analytical descriptions are encoded and extracted according to the requirements of the user/researcher.

At the lowest level of the musical surface (i.e. the actual notes of a musical piece) a custom text-based encoding is used; annotated pieces are encoded using the musicXML format. This format has been chosen for the following reasons: a) it is a widely used music encoding scheme for music scores and most commercial music notation packages have adopted it, b) it is user-friendly in the sense that musicians/musicologists can readily create annotated files for music pieces and save them as musicXML files using the music notation software of their preference, c) it can be adjusted to incorporate any required structural harmonic feature, d) computational environments (such as music21) are available for handling musicXML files and extracting appropriate harmonic structural features, e) annotations can easily be inspected by a user as they are score files that can be read by any standard music notation package.

At a harmonic level above the plain note level, a new idiom-independent representation of chord types that is appropriate for encoding tone simultaneities in any harmonic context (such as tonal, modal, jazz, octatonic, atonal) is proposed. The General Chord Type (GCT) [1] representation, allows the re-arrangement of the notes of a harmonic simultaneity such that abstract idiom-specific types of chords may be derived; this encoding is inspired by the standard roman numeral chord type labeling, but is more general and flexible. The proposed representation is ideal for hierarchic harmonic systems such as the tonal system and its many variations, but adjusts to any other harmonic system such as post-tonal, atonal music, or traditional polyphonic systems. This novel chord-type representation is considered of utmost importance for developing a common harmonic generic space for chords that will facilitate blending of seemingly incompatible harmonic idioms (e.g. tonal and atonal).

In this report we present a custom information retrieval scheme for musical data and annotations, and a computational framework that manages the representation of the above information for the objectives of the COINVENT's deliverable "D7.1 Harmonic training dataset". Initially, we selected music pieces with diverse harmonic content and organized them into groups depending on various analytical criteria. Secondly, we created an open representation for musical data and annotations to encode them. Thirdly, we developed an extensible computational shell to manage various aspects of the dataset and perform operations for exporting the desired information. Finally, we created a novel chord-type representation scheme that allows a common encoding of chords in any harmonic idiom facilitating thus further learning and conceptual blending.

The remaining of the report is organized as follows: Section 2 describes the harmonic content of the dataset from a musicological point of view indicating various ways it can be grouped to obtain meaningful information (historically, evolutionary, composer-styles etc.). In Section 3, we present an extensible and versatile music encoding scheme for annotated music. Although many symbolic representation templates have been proposed for music, we justify some of the reasons and benefits that led us to the customization of a selected scheme. Additionally, this section describes the computational framework that manages the above representation. In general, this is an extensible data interface that can operate various datasets and extract primitive and structured information. This system could be used as a data-layer service providing parsing and database operations interfacing datasets and models. In Section 4 we explain the General Chord Type Representation that forms the basis for representing note simultaneities in any idiom. Future perspectives about the data, the data management tools and the overall orientation of the COINVENT melodic harmonizer research are presented in Section 5.

2 Dataset description

The dataset consists of over 430 manually annotated musicXML documents categorized in 7 sets and various subsets. The separation of pieces in sets primarily focuses on genre categorization, while subsets are created within genres that present notable differences in their harmonic structure. For instance, the Chorales of Bach belong to different set than Jazz music, while modal Jazz pieces belong to a different subset from jazz standards that incorporate tonality transpositions. To this end, the purpose of this dataset is not only to constitute a rich knowledge background of examples that facilitates conceptual blending, but also to provide valid and accurate harmonic content in an accessible, user-friendly encoding for various model applications in computational musicology.

There are many ways to set up and manage a music database. The most common approach is to use file documents as the lowest dataset entry (RS200 dataset, 9GDB dataset, Isophonic annotations and ISPG dataset, McGill Billboard dataset). Although there are many custom file formats that are usually accompanied with scripts and frameworks to operate them (e.g. humdrum), there are efforts to standardize both music data documents structure (KERN, musicXML) and even more specific music encodings (e.g. Harte, text-Harmony). Other approaches aggregate data encodings in single text files (KP dataset[1] , DFT dataset[2] , JazzCorpus) and there are some interesting implementations that use database systems like MySQL (Jazzomat monophonic dataset). (More details about music representations in Section 2)

The symbolic music file format "musicXML" is becoming a standard in score notation[3] and its formal document structure provides a convenient data container for computational access. In recent years there is active development of computational tools that operate with musicXML files (e.g. music21) and public datasets with musicXML are emerging (e.g. GTTM dataset[4]). Advances in music computational frameworks potentially enable each musicXML score to become the basis for general-purpose music datasets.

Since the dataset is oriented for harmonic training, the criteria for music selection were focused on the harmonic content of musical pieces and excerpts. Depending on a variety of harmonic features, these dataset entries were grouped into subsets to form various training cases. The primary criteria for selecting the dataset's pieces were *diversity* and *consistency* of harmonic features between subsets. Diversity in harmonic features allows the inclusion of a wider spectrum of options– a richer knowledge background – for potential blends in harmonic concepts, enabling the COINVENT melodic harmonizer to produce harmonizations with strong references to diverse idioms. The term "consistency" on the other hand, indicates each subset of pieces encompasses a "pattern" in harmonic features that is characteristic to this subset, in a sense that these features are often encountered in several pieces within this idiom. Thereby, the produced harmonic blends will include diverse harmonic features that constitute strong references to the incorporated idioms.

2.1 Background

The selection of pieces or excerpts for the creation of the categories and subcategories of the dataset was mainly determined by a contextual and evolutional conceptualization of harmony and of the art of melodic harmonization. In this context, two different melodic harmonization principles are considered as processional frameworks:

1. the use of sonorities stemming from the melodic pitch space (closed system), or
2. the use of sonorities beyond the pitch space of the melody (open system).

In this context, the pitch class material of the melody and of the harmony can be regarded as two separate pc sets with variable number of common elements (their intersection ranging from the null set to full equality).

Between these extremes various mixed/hybrid systems can occur and a multitude of harmonization concepts can be employed. Commencing from the hypothesis that the creation of hybrid systems by the composers of each particular era/place/style actually generated the historical and stylistical evolution and diversity of the art of harmonization, we arrived at a hierarchy of categories of musical pieces/excerpts suitable for the training dataset.

[1] http://theory.esm.rochester.edu/temperley/kp-stats/

[2] http://marl.smusic.nyu.edu/wordpress/projects/feature-learning-deep-architectures/deep-learning-python-tutorial/#data

[3] http://www.musicxml.com/software/

[4] http://music.iit.tsukuba.ac.jp/hamanaka/gttm.htm#

The creation of a simple and stylistically coherent harmonization system involves some basic harmonization principles that can be considered universal and diachronic:

1. the identification of the melody's structural pitches, which will be part of the chords/sonorities that will be used in the context of the chosen harmonic system and
2. the application of rules for voice-leading (linear movement, manipulation of dissonance) and rules for functional progression of chords (if such exist) and
3. the creation of melodic/harmonic closure formulas (cadence patterns).

However, following the concept described above, the evolution of new complex and diverse harmonic styles and idioms can be seen as a combination of closed or open harmonic systems and/or of the harmonic concepts that they incorporate, i.e. by conceptual blending. So, since our goal is a system that can invent interesting novel harmonizations through blending, a need for the study of simple harmonic systems emerges, from which blending can occur at later stages.

2.2 Criteria and categories

The main criteria for the inclusion of a category of pieces or excerpts in the dataset was stylistic integrity, the capability of breaking down the harmonization style to a relatively small number of simple interconnecting concepts, the degree of evolutionary overlapping between neighboring categories and the broad stylistic diversity of the entire corpus. In this context, the following historical or stylistical broad categories – with a brief description of their properties/concepts – were considered:

1. Modal Harmonization in the Middle Ages (11th – 14th centuries).
    - Closed system: the pitch material is confined to the diatonic space of the eight diatonic modes (1st to 8th – later called Dorian, Phrygian, Lydian, Mixolydian and their plagal forms). External/open element: the use of B♭ for the avoidance of the tritone.
    - Almost exclusive use of parallel sonorities made of perfect 8ths and 5ths (parallel organum technique).
    - Optional use of parallel sonorities made of minor or major 3rds and 6ths (fauxbourdon technique), in combination with perfect interval at cadences points
    - Gradual use of contrary and oblique movement, as well as of other vertical intervals.
2. Modal Harmonization in the Renaissance (15th – 16th centuries).
    - Closed System: both melody and harmony come from the heptatonic diatonic pitch space (system of 8 or 12 modes, with two possible pitch collections) with one main pitch center (initial/final mode) and other temporary pitch centers (other modes). External/Open Elements: Pitch classes B♭ and E♭ (mobile pitches) and the artificial leading tones (C♯, F♯, G♯). Overall, a non-enharmonic chromatic scale is produced.
    - Use of tertian chords (chords built from stacked major and minor 3rds): major chords, minor chords, diminished chords in 1st inversion.

- Non-functional chord progression, except from the cadence points, where the descending perfect 5th relation prevails.
- Voice leading: Avoidance of parallel perfect intervals, dissonance in weak beats as passing or auxiliary notes, dissonance in strong beats with preparation and stepwise downward resolution, use of chromatic leading tones in cadences.
- During the transitional homophonic style (ca 1600 a.D.) a gradual liberation of chromaticism occurred, especially in madrigals.

3. Tonal Harmonization (17th-19th centuries).

    - Closed System: heptatonic diatonic scales of two types (major, minor) in 12 transpositions. Permanent external element: the chromatic leading tone in minor scales. The pitches of pitch collections come from seven steps in the cycle of perfect 5ths. There are 12 discrete diatonic pitch collections (key signatures).
    - Main use of tertian chords belonging to the basic tonality.
    - Possible transposition of the harmonized melody to other tonalities, that are up to six steps away in the circle of perfect 5ths.
    - Open elements: use of certain types of external/chromatic tertian chords that do not belong to the main tonality, but are borrowed from other tonalities (e.g. borrowed dominant chords, neapolitan chord, etc.) or transformation of the diatonic chords (e.g. chromatic or altered chords, mode mixture). So, chromatic tonal harmony is a partially open/hybrid system.
    - Functional chord progressions, stemming mainly from 5th and 3rd chord relationships. Part of the functional system is the systematic use of the "perfect cadence" V $\rightarrow$ I (major dominant to major or minor tonic) at the conclusion of every major formal segment.
    - Rules of voice leading: 1) avoidance of parallel perfect intervals, 2) dissonances (mainly in 7th chords) are resolved with downward stepwise motion.

4. Harmonization in National Schools (19th – 20th centuries).

    - Return of diatonic modality, but with more liberal, chromatic harmonization (mixture of closed and open systems).
    - Liberation from functional chord progressions and from the obligatory use of the perfect cadence.
    - Mixture of modal, tonal and free chromatic harmony (blending)
    - Gradual use of post-tonal harmonic structures: whole-tone scale, octatonic scale, synthetic scales.

5. Harmonization in the 20th century (extensive blending).

    - Closed systems with diatonic or non-diatonic (symmetrical scales, chromatic/synthetic scales) pitch structures.
    - Closed systems with non-triadic sonorities (verticalities created from 4ths, 5ths, 2nds, mixed intervals, extended/altered tertian chords, polychords, free verticalization of scales/modes).

- Open harmonization systems of every possible type and hybridization level, up to the complete absence of common pitches between melody and harmony.
- Abolishment of traditional voice leading: emancipation of parallel sonorities (diatonic or real) and of dissonances.
- Abolishment of functionality stemming from the circle of 5ths, emphasis on the “color” and “character” of sonorities. Free cadence patterns, absence of explicit pitch center, pandiatonicism.
- Multiple layers of musical texture and harmonization.

6. Harmonization in folk traditions.

    - Mainly closed systems, where melody and harmony stem from the same pitch class set/mode.
    - Most systems are strictly monophonic and the harmonization is considered an external distorting element (consequence of blending).
    - External elements (blending types) create widely-ranging chromaticism (e.g. the use of major chords with minor 7th for the harmonization of minor pentatonic melodies in the blues idiom).

7. Harmonization in 20th century popular music and jazz.

    - Closed modal or tonal harmonic systems greatly differing in style and individual features.
    - Open systems with fusion between styles (extensive blending).

Within the aforementioned categories, we preferred (with some interesting exceptions) mainly homophonic styles/sub-categories that allow for explicit separation of the melodic line and its harmonization (e.g. chorale harmonizations, folk song harmonizations within national styles, songs for voice and piano, choir music, etc.) and for identification of cadence/closure patterns (which provide syntactical organization). Also, within the 20th century category, we preferred the neotonal styles from the atonal ones, as the former yield more clearly identifiable harmonizational features through the concept of pitch center and the use of specific pitch spaces. Moreover, an emphasis was given to the harmonization of Greek folk songs (by three different composers: Labelet Constantinidis and Skalkottas, each with distinctively different harmonic style), as well on Greek folk music (Epirus polyphonic songs and Rebetika songs).

The dataset consists mainly of complete relatively short pieces, that provide syntactically complete harmonic structures. However, the use of short excerpts over complete pieces was preferred in cases where the amount of harmonic concepts of a category was relatively large, and a concise way of training the system to this multitude of concepts was required. Two such cases must be highlighted:

1. the classical/romantic tonal harmonization category, that was described by the public dataset of Kostka-Payne corpus (KP dataset) as presented from David Temperley (chord-list file) and Bryan Pardo (midi files with chord’s quality)

2. the 20th century harmonization techniques category, that was described by selected excerpts from the textbooks of Stefan Kostka, Kent Williams, Stefan Kostka and Dorothy Payne, Connie Mayfield, Walter Piston and various other sources (see details in Table 1, point 5.1.).

2.3 The Dataset – categories and sub-categories

Summarizing, the categorization in Table 1 was made with specific training targets in mind:

- Historical evolution of harmony through closed or open systems.
- Harmonic content (types of sonorities employed and their diversity).
- Harmonic syntax (chord functions, cadence types).
- Amount of blending between idioms or harmonization concepts.
- Styles of special interest (composer or idiom specific).

Moreover, since research goals are forming the grouping criteria, we are creating a versatile filter mechanism so that groups can be created from meta-data queries (e.g. time period composed) or even by their harmonic properties (e.g. major phrases)(see Section 4).

Conclusively, the full list of categories and subcategories that we considered appropriate for this approach is the following:

Table 1: Categorization of the dataset files. For an extensive listing of the files attributes, the interested reader is referred to the webpage. The numbers in parenthesis beside the dataset subset name indicate the number of pieces that comprise the collection.

1. Modal Harmonization in the Middle Ages (11th–14th centuries).
   (a) Medieval Sets (12)
      - Organum
      - Fauxbourdon
      - Franconian motet
2. Modal Harmonization in the Renaissance (15th-16th centuries) (44)
   (a) Modal 16-17th cent. music (10)
      - Motet
      - Madrigal
      - Frottola
      - Stabat Mater
   (b) Modal Chorales (34)
      - ionian
      - dorian
      - phrygian
      - lydian
      - mixolydian
      - aeolian

3. Tonal Harmonization (17th–19th centuries) (100)

   (a) J.S. Bach Chorales (35)
       - major
       - minor
   (b) Kostka-Payne corpus (KP dataset) (46)
   (c) Tonal Harmonization Sets (18th–19th century) (19)

4. Harmonization in National Schools (19th–20th centuries) – folk song harmonizations (55)

   (a) Norway: Edvard Grieg (10)
   (b) Hungary-Romania: Béla Bartók (17)
   (c) Greece: George Labelet 5/ (simple diatonic modal style) (5)
   (d) Greece: Yannis Constantinidis 20/ (impressionistic style) (20)
   (e) Greece: Nikos Skalkottas 6/ (chromatic style) (3)

5. Harmonization in the 20th century (109)

   (a) Excerpts (ca 80) for training on specific harmonization techniques and concepts (89):
       - Category 1: simple concepts
         – Scales/harmonic spaces: diatonic modes, pentatonic modes, acoustic scale, octatonic scale, whole-tone scale, hexatonic scale, altered scale, chromatic scale/atonal space
         – Harmony and Sonorities: tertian harmony, quartal harmony, secundal harmony
         – Texture and Voice-leading: parallel harmony - diatonic, parallel harmony - chromatic, pandiatonicism, pedal notes/drones
       - Category 2: conceptual blending
         – Scales/harmonic spaces: modal interchange, bitonality/bimodality
         – Harmony and Sonorities: free sonorities, poly-chords, chords with added notes, chords with split members
         – Texture and Voice-leading: multi-layer texture
   (b) Choral music a capella (neoclassical-neotonal styles) (15):
       - Claude Debussy: Trois Chansons
       - Paul Hindemith: Six Chansons
       - Eric Whitacre: A boy and a girl, Cloudburst (b. 1-57), Sleep, Water Night, With a lilly in your hand
       - Igor Stravinsky: Pater Noster, Ave Maria (2 versions)
       - Fidel Calalang: Pater Noster
       - Alfred Schnittke: 3 choral pieces
   (c) Vocal music (neotonal style) (5)

6. Harmonization in folk traditions (75)

   (a) Tango (classical and nuevo styles) (24)
   (b) Epirus polyphonic singing (based on minor pentatonic pitch space) - Selection of 22 three or four-voice pieces (transcriptions by Kostas Lolis) (29):
       - Three-voice: nos. 28, 29, 30, 32, 35, 36, 39, 42, 44, 45, 49, 51, 56, 57, 59, 61
       - Four-voice: nos. 68, 70, 72, 73, 76, 77, 78, 79, 82, 84, 87, 90, 91, 92
   (c) Rebetika songs (based on the triadic harmonization of oriental modes) (22)

7. Harmonization in 20th century popular music and jazz (65)

   (a) Mainstream Jazz (30)
   (b) Bill Evans (25)
   (c) Beatles songs (10)

3 Encoding of the annotated data

The creation and operation of a harmonic dataset requires the demarcation of the specifics in harmonic information that are required for the task at hand, as well as the level of detail. Additionally, since the project's goal is based in machine learning and artificial intelligence algorithms, the data encoding process that is followed has been developed to satisfy two aspects: human annotations (concerning facilitation with musical score) and richness and consistency of harmonic information. Harmony and music in general potentially incorporate many levels of information, beginning from low-level audio, to high-level descriptive information of composite concepts, e.g. chords, tonality, genre, etc. The types of information that are required for a musical piece, in the context of COINVENT, are only symbolic (no audio information is required), roughly described by the following point: a) primitive musical data that are described by musical surfaces, b) expert annotations that are provided as manually entered analytical information about the contents and c) meta-data for database management.

The *musical surface* is the lowest level of representation that has musical significance ([18], p.219). The current proposal provides voicing, temporal and pitch attributes to music elements, essentially notes. Voicing information concerns the horizontal organization of the notes, while temporal information positions each note in the rhythmic and metric lattice. Pitch has many ways to be represented depending on the pitch context we refer to. Concerning the dataset collection and the algorithmic context that is utilized for the present project, each note has an exact symbolic notation based on the equal tempered scale.

Expert annotations in a music piece describe musicological aspects that refer to specific analytical concepts (e.g. the use of harmonic symbols to describe note simultaneities, modulations, phrasing etc.). In general, annotations may incorporate any kind of musical information that may be required, for any element or group of elements from the musical surface. Expert annotations are of utmost importance towards the COINVENT melodic harmonizer development, since these annotations isolate and describe some harmonic concepts that will comprise parts of the blending process. Expert annotations are more thoroughly described in Section 3.3.

Meta-data are helpful for the organization of large collections of pieces and therefore their inclusion is essential. Meta-data include information about the composer, year of composition and additional information concerning some compositional techniques utilized in each piece – as a means of indexing harmonic concepts that may be included.

3.1 Previous work on text based music encoding

There is a plethora of text-based encoding schemata for music [31]. While it is beyond the purposes of this report to provide a complete reference for all these encoding methodologies, some of them are outlined, focussing on their drawbacks concerning the COINVENT melodic harmonizer perspective, consequently highlighting the reasons that lead us to develop a novel encoding and annotation scheme.

Among the most prominent – yet legacy – musical encodings for polyphonic encodings are the Humdrum and Kern [17] and the MuseData [15]. Humdrum consists of two distinct components, namely the Humdrum Syntax and the Humdrum Toolkit. The first is a grammar for representing information using a variety of encoding schemes. The **kern format is a score representation scheme. The Humdrum Toolkit is a collection of UNIX scripts that were designed to work in conjunction with each other, as well as in combination with UNIX commands in order to manipulate text data (ASCII) that conform to the humdrum syntax. Humdrum syntax is like a spreadsheet where columns can represent various types of data and rows successive moments in time. The

**kern encoding represents the basic or core musical score-related information, emphasising on the underlying functional information between musical elements rather than the visual information of a score.

The principle that supports the development of these encodings and their accompanying tools, is the conversion of symbolic music data to ASCII-based, machine-readable formats. These encodings and the accompanying tools are suitable for representing musical primitives, as well as extracting some harmonic abstractions, like chords and tonality. However, these encoding systems are "closed", in a sense that they do not support in an obvious way expandability to custom annotations. For instance, it would be difficult to annotate phrase grouping, the beginning of a cadence or a cadence extension.

The need for transforming information about music to machine readable formats has led to the development of the Music Encoding Initiative (MEI[5]). MEI is am attempt to create a unified framework that encompasses heterogeneous information about music, covering a wide range in levels of abstraction, e.g. information about historical data, audio, technological aspects of recording, or music-theoretical considerations. All this diverse information is gathered and encoded into the MEI schema, which is a core set of rules for recording physical and intellectual characteristics of music notation documents. Ultimate goal of MEI is to eliminate the need for creating specialized and potentially incompatible notation encoding standards. However, the MEI framework has not yet expanded to the harmonic encoding according to the standards required for the COINVENT melodic harmonizer research – even though the encoding developed for COINVENT could potentially be a part of the MEI framework in the future [12].

3.2 Selecting a music representation for the harmonic training dataset

Regarding the harmonic representation, a goal was set to achieve a balance between human-friendly interpretation of data and computationally accessible encoding. Specifically, the desired characteristics that shaped the representation format were:

1. Low level – maximum information from musical surface, which allows the extraction of musical information from the original musical text, according to potential requirements of future research.
2. Access to primitive data, which again potentially enables the extraction of information at any level.
3. Visual (score) edits, a fact that enables musicologists and musicians who are familiar with music notation software to actively participate in the current and potential future research, by providing manual annotations to a "written language" they understand better.
4. Manual and algorithmic annotations, thus enabling not only human, but also computer-driven interpretation of data, making the dataset easily manageable for computationally-based research.
5. Extensibility for future additions, allowing the inclusion of annotated fields that might emerge from future research on the dataset under discussion.

[5]http://www.music-encoding.org

To accomplish these qualifications we selected common music notation as the basic representation for both music surfaces and annotations, i.e. both notes and harmonic structural markers are inserted and displayed on musical staves. Encoding is thereby realized in a structured musicXML file with specific staff types and custom music-notation encoding schemes. Since the level of descriptiveness and information is meant to be maximal, standard music notation provides enough detail to map the musical surface and also a convenient representation to reference complex temporal and data formations.

We selected the musicXML format as the representation scheme for the encoding of the harmonic training dataset. MusicXMLis an XML instance intended as an interchange format sufficient for notation, analysis, retrieval, and performance applications, and, in fact, it is the most widely implemented in music notation programs [14]. The content represented by the format is score-oriented, i.e., notes are represented as symbolic and graphical objects. MusicXML addresses the integration of the performance and notational aspects of music, but it does not address the integration of other layers such as audio or structural. As demonstrated by MusicXML [8, 9], representation of music information with XML can be a successful approach to data standardization and interchange. In this case, the adoption of musicXML by commercial software has taken place very quickly. This has led it to be the first XML music format widely used in the music notation software market.

Additionally, the musicXML format offers capabilities to graphically represent and process music through the music21 toolkit [3], which has a well-developed and expandable framework for importing scores and other data from the most common symbolic music software (e.g. Finale, Sibelius, MuseScore, among others). Thereby, music21 and musicXML offer the capability to have both notational efficiency for music experts (through advanced score editing facilities) and flexibility in extracting symbolic or numerical music features – for applying machine learning and artificial intelligence algorithms. In fact, the music21 toolkit has been successfully integrated in a system that allows the extraction of musical features from musical score, allowing various music categorization tasks to be successfully materialized [4].

The graphical edit abilities of this representation enable the creation of a dataset entry by any user accustomed with music notation software. Moreover, we selected and extended the music21 toolbox since this is a highly active and adequate programming interface for this and other formats.

3.3 A musicXML template for the Harmonic training dataset

Since musicXML represents a score, score components such as staves and stave groups act as the basic data containers for all operations. We group the staves of the document in two basic categories: a) musical surfaces and b) annotations. Music surface (ms) staves encode the music data while annotation staves encode any of the annotator's descriptions in custom dictionaries of music notation. Examples of such annotated descriptions are "tonality" and "grouping" structure, which are elaborated later. Additionally, all parts of the score (ms and annotation staves) share temporal organization (time-signatures), whereas annotation staves have independent and neutral key. Annotation staves are required to comply with the temporal organization of the piece, since their indications concern the time instance where an event happens (e.g. when tonality changes). On the other hand, annotation staves use pitch-specific identifiers with an encoding scheme that does not require the indication of a tonality key.

A music surface can be encoded in single or paired staves. The original music content is encoded in staves tagged as ms_0. Any reduction levels (time-span reductions) of the original content are encoded in another set of staves named ms_N where N is the level of the time-span reduction. The amount of musical surface reductions (ms_N) that a document may reach is unlimited. A feature that is examined in each surface part is voicing information. There are three cases concerning the voicing information of the music data: a) voice organization exists and each voice is monophonic (Vm), b) voice organization exists but voices are not monophonic (streams) (Vs), and c) no voicing information is available (Vf). Knowing the status of voicing information is crucial since various operations depend on that.

Annotation staves are parts with specific music notation. Each annotation staff interprets the indicators it contains in music notation. Using music notation for annotations, we can annotate using common music notation software and edit custom indicators visually. Although the annotation parts were added to the template to manually annotate valid information about aspects of the music surface, various algorithms can be used to add information computationally. For instance, the tonality annotation can be automatically computed by a key detection algorithm [21]. Annotation staves are unlimited, but each part must follow a specific formalism, as defined by a *dictionary*. A *dictionary* file includes translations of the annotation descriptions, as defined by the annotator; since all the annotations are manually entered, the annotator is responsible for the translation encoding that an annotation staff offers. For the tonality and grouping annotations that are presented later, we have developed a custom dictionary that is analyzed in detail.

Apart from the music surface and the annotations, for each piece in the harmonic training dataset we encoded the meta-data in a text file and the music data together with the annotations in a musicXML file, using at least the information levels exhibited in Table 2. Specifically, the musicXML pieces in the dataset consist of (at least) six staves: two or more staves for the original song/content preserving voicing information (ms0, Vm), one for tonality annotations where the scale is written as a note cluster, one for grouping boundaries where the number of notes indicates grouping level and two parts that contain an annotated time-span reduction of the original content (ms1) (see Figures 2 and 3). Whenever voicing information is not available (e.g. in homophonic pieces), at least the voice of the piece's melody is annotated in Vm.

Table 2: Least requirements for information information levels for musicXML files annotation.

music surface	ms0, Vm/Vs (monophonic voices when available)
music surface	ms1, Vm/Vs (monophonic voices when available)
annotation	Tonality - basic scales dictionary
annotation	Grouping - phrases, monophonic content, cadence extension

The songs were encoded at two reductional levels, corresponding to two adjacent levels of the metrical/time-span hierarchy: level ms0 closely describes the musical surface by including embellishing figures, neighbor notes, etc. and corresponds to the metrical level of sixteenth or eighth notes (depending on the metrical tactus and beat level) of the transcription, while level ms1 describes a deeper structure by omitting most elaborations[6] and corresponds to the eighth or fourth notes of the transcription – see example in Figures 2 and 3. The reduction was deemed analytically

[6]Elaborations include passing tones, neighboring tones, appoggiatura, escape tone, suspension, retardation, anticipation

Figure 1: Annotated xml file containing original song transcription (ms0), time-span reduction of the original content (ms1), as well as tonality and grouping information.

necessary in order to disclose the idiom's harmonic functions and cadence patterns. There is no limit to the number of reductions that can be added to the xml. However, through reduction, while we gain clear chordal content, we have loss of voice leading information.

Tonality staff annotations This part contains information about the tonality changes at the lowest level musical surface. Tonality indicators contain all the scale tones in the form of note-cluster chords and are placed whenever there is key change. The system reads the enharmonic pitch of the base of the cluster and by analyzing the interval components it matches a tonality description from the tonality annotations dictionary. A tonality dictionary includes vectors with scale degrees for standardized scales, e.g. major, minor-harmonic, pentatonic, octatonic, whole-tone, acoustic and modal modes among others (see tonality.dic). Examples of how various tonalities are presented on score are given in Figure 2, where a part of an annotated xml piece is illustrated. Therein, the tonality staff indicated two tonality changes. The indicators of the tonality – and the tonality changes – include accidentals in chordal form, while the time instance of a tonality activation/change is defined by the indication's onset. Additionally, it has to be noted that at least one tonality indicator at the beginning of the piece is required otherwise the tonality annotations of the piece are considered absent, while repetitions of the same indicator are ignored.

Figure 3 demonstrates examples of various tonalities. Using this encoding to identify the tonality annotations there is a conflict, for example, between Major scales and Ionian modal modes, since these scales have the same intervalic content. A possible solution to this and other conflicts that may appear is to make the identifiers more complex, showing the structural/dominant tones of the scale. Another possible improvement might be the extension of the syntax to be able to map modulation regions where the tonality is not clear and more than one tonality indicators are required. At the moment, using pieces from a variety of idioms, the current representation is

Figure 2: Tonality change annotations for Bach Chorale B.230, BWV 273. mm. 1-2

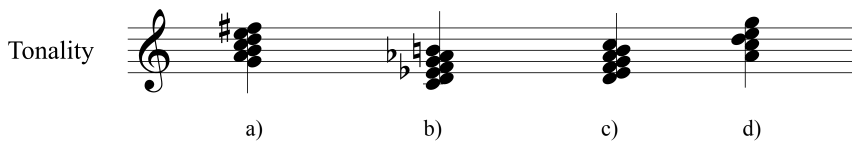

Figure 3: Examples of tonality indicators: a) G Major, b) C Minor, c) D Dorian, d) A minor pentatonic.

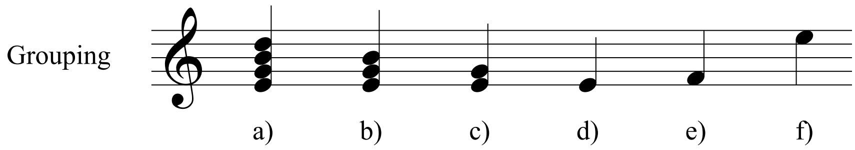

Figure 4: Phrasing indicators. (a-d) phrase level from higher to lower. The cardinality of the indicator chord indicates the level. e) cadence extension f) monophonic content.

adequate.

Grouping staff annotations This part contains annotations about melodically coherent temporal regions of the music surface. The current grouping dictionary contains three different classes of grouping indicators: a) phrasing levels of the musical surface b) monophonic phrases and c) cadence extensions. At the beginning of each phrase, a group identifier is placed indicating the level of the phrase hierarchy. One note on the lowest line indicates the lowest level groupings (e.g. phrase); two notes on the lowest two lines indicate an immediately higher-level of grouping (e.g. period); three notes indicate even higher level of grouping and so on. We added an indicator for monophonic passages so that we can exclude them from harmonic content extraction. The cadence extension indicator was used to isolate parts that succeed a cadence, but are not part of it.

Table 3: Analysis of the grouping indicators, dictionary and syntax.

Indicators	phrase levels {E=lowest level, EG, EGB, EGBD= highest level} monophonic content {F=77} cadence extension {F=65}	
Dictionary	the cardinality of the chord indicates the level of the phrase	
	MIDI note	descriptor
	`64`	`tsr1`
	`64 67`	`tsr2`
	`64 67 71`	`tsr3`
	`64 67 71 74`	`tsr4`
	`76`	`mono`
	`65`	`cadExt`
Syntax	all possible orders of group indicators are valid the indicators cannot be combined at least one indicator must be present	

In general, this part can be used to label temporal regions based on any aspect of their content. While the examination of the order of the phrases can reveal morphological aspects, the grouping dictionary can be easily extended to include morphological description (e.g. verses) and any other information that needs to be taken into consideration. Even more, the development of a more complex syntax could enable the combination of overlapping temporal descriptions of different classes and perform more accurate temporal references.

The meta-data of the documents In the current version of the dataset the meta-data of the pieces are stored in separate text files and are processed separately. The fields stored within the musicXML file are the following:

1. file, the filename of the piece follows a naming convention that also indicates the piece information level e.g. "BC_030_036300B_a1.xml"
2. genre(s)
3. piece name
4. catalogue(s)
5. segment
6. measures
7. composer
8. date of composition
9. original resource
10. annotator

3.4 Computational usage of the harmonic dataset

A purpose-oriented computational interface has been developed to access the aforementioned representation of the musicXML dataset. The xml format of the dataset files can be utilized using any xml parser, however we chose to interface with the musicXML documents through the music21 toolkit, for the reasons elaborated in Section 3.2, as well as for the fact this toolkit provides the essential music ontology to operate score documents computationally in a flexible way using Python programming language. On top of music21, we created a simple framework to manage the musicXML template of the dataset, perform musicological queries and export structured information.

The main usage of this tool is to extract structured descriptions of harmonic content from score selections following a specific workflow. This workflow contains the following steps (see the example it Table 4): a) the conversion of the musicXML files within a folder into document objects (`<musicXMLfilesfolderPath>`), b) the selection and grouping of musical surface elements (`<selectionFunction>`) and c) the extraction of their content in various formats (`<formatFunction>`). To support the above operations we created a simple set of classes and modules.

Table 4: Pseudocode for selecting specific chords from musicXML files. Lines of code that represent comments begin with the symbol "#", variables are represented in inequality signs (e.g. "`<variable>`"), while the function `len([list])` returns the length of a list.

Example 1: Selection of the chords from `<ms1>` that appear in `<strong>` metric positions.

```
output = []
doc = htd_document(<musicXML file>)
# getChords returns a stream with music21.Chord objects
# use of music21.Stream.chordify()
stream = doc.getChords(<ms>)
FOR chord IN stream
   # use of Music21Object.beatStrength
   IF chord.beatStrength == <strong>
      output.append(chord)
RETURN output
```

Example 2. Selection of all the chords from `<ms1>` that appear in phrases with specific `<group level>`.

```
Input : a musicXML file,
Output : a list with music21.Chord objects

output = []
doc = htd_document(<musicXML file>)
# get all the chords in a stream. use of music21.Stream.chordify()
stream = doc.getChords(<ms1>)
FOR chord IN stream
   # getGroupLevel() returns the grouping context of the chord
   IF chord.getGroupLevel() == <group level>
      output.append(chord)
RETURN output
```

Example 3. Selection of the last 3 chords of each phrase with no tonality modulation.

```
Input : a musicXML file
Output : a list with music21.Stream objects

output = []
doc = htd_document(<musicXML file>)
phrases = doc.getPhrases(<ms1>) # returns a list with streams
FOR phrase IN phrases
   # the number of tonalities
   IF len(phrase.getTonalities)== 1
      phraseChords = phrase.getChords()
      # last 3 elements of the list
      output.append(phraseChords[:-3])
RETURN output
```

Example 4. Selection of all the chords and that have extensions (including 7th and above).

```
Input : a musicXML file
Output : a list with music21.Chord objects

output = []
doc = htd_document(<musicXML file>)
# returns a music21.Stream with music21.Chords
chords = doc.getChords(<ms1>)
FOR chord IN chords
   # use of music21.Chord.isTriad()
   IF not (chord.isTriad())
      output.append(chord)
RETURN output
```

Example 5. Get tonal deviation (only relations V7 / <>).

```
Input : a musicXML file
Output : a list with music21.Stream objects. Each stream contains the
two chords that partain to the V7 / <> relation.

output = []
doc = htd_document(<musicXML file>)
chords = doc.getChords(<ms1>)
FOR index IN range(len(chords)-1)
   deviationChord = chords[index]
   baseChord = chords[index+1]
   # use of music21.Chord.isDominantSeventh()
   IF deviationChord.isDominantSeventh()
      # use of music21.Chord.findRoot ()
      baseRoot = baseChord.findRoot()
      deviationRoot = deviationChord.findRoot()
      if (((deviationRoot.midi+12) - (baseRoot.midi)) % 12) == 7
      stream = []
      stream.append(deviationChord)
      stream.append(baseChord)
      output.append(stream)
RETURN output
```

4 An Idiom-independent Representation of Chords for Computational Music Analysis and Generation

In this section we focus on issues of harmonic representation and computational analysis. A new idiom-independent representation is proposed of chord types that is appropriate for encoding tone simultaneities in any harmonic context (such as tonal, modal, jazz, octatonic, atonal). The General Chord Type (GCT) representation, allows the re-arrangement of the notes of a harmonic simultaneity such that abstract idiom-specific types of chords may be derived; this encoding is inspired by the standard roman numeral chord type labeling, but is more general and flexible. Given a consonance-dissonance classification of intervals (that reflects culturally-dependent notions of consonance/dissonance), and a scale, the GCT algorithm finds the maximal subset of notes of a given note simultaneity that contains only consonant intervals; this maximal subset forms the base upon which the chord type is built. The proposed representation is ideal for hierarchic harmonic systems such as the tonal system and its many variations, but adjusts to any other harmonic system such as post-tonal, atonal music, or traditional polyphonic systems. The GCT representation is applied to a small set of examples from diverse musical idioms, and its output is illustrated and analysed showing its potential, especially, for computational music analysis and music information retrieval.

There exist different typologies for encoding note simultaneities that embody different levels of harmonic information/abstraction and cover different harmonic idioms. For instance, for tonal musics, chord notations such as the following are commonly used: figured bass (pitch classes denoted above a bass note – no concept of “chord”), popular music guitar style notation or jazz notation (absolute chord), roman numeral encoding (relative to a key) [22]. For atonal and other non-tonal systems, pc-set theoretic encodings [7] may be employed.

A question arises: is it possible to devise a "universal" chord representation that adapts to different harmonic idioms? Is it possible to determine a mechanism that, given some fundamental idiom features, such as pitch hierarchy and consonance/dissonance classification, can automatically encode pitch simultaneities in a pertinent manner for the idiom at hand? Before attempting to answer the above question one could ask: What might such a "universal" encoding system be useful for? Apart from music–theoretic interest and cognitive considerations/implications, a general chord encoding representation may allow developing generic harmonic systems that may be adaptable to diverse harmonic idioms, rather than designing ad hoc systems for individual harmonic spaces. This was the primary aim for devising the General Chord Type (GCT) representation. In the case of the project COINVENT [30], a creative melodic harmonisation system is required that relies on conceptual blending between diverse harmonic spaces in order to generate novel harmonic constructions; mapping between such different spaces is facilitated when the shared generic space is defined with clarity, its generic concepts are expressed in a general and idiom-independent manner, and a common general representation is available.

In recent years, many melodic harmonisation systems have been developed, some rule-based ([6, 24]) or evolutionary approaches that utilize rule based fitness evaluation ([27, 5]), others relying on machine learning techniques like probabilistic approaches ([25, 32]) and neural networks ([16]) , grammars ([11]) or hybrid systems (e.g. [2]). Almost all of these systems model aspects of tonal harmony: from "standard" Bach–like chorale harmonisation ([6, 16] among many others) to tonal systems such as "classic" jazz or pop ([32, 11] among others). Aim of these systems is to produce harmonizations of melodies that reflect the style of the discussed idiom, which is pursued by utilising chords and chord annotations that are characteristic of the idiom. For instance, the chord representation for studies in the Bach chorales include standard Roman numeral symbols while jazz approaches encompass additional information about extensions in the guitar style encoding.

For tonal computational models, Harte's representation [13] provides a systematic, context-independent syntax for representing chord symbols which can easily be written and understood by musicians, and, at the same time, is simple and unambiguous to parse with computer programs. This chord representation is very useful for annotating manually tonal music – mostly genres such as pop, rock, jazz that use guitar-style notation. However, it cannot be automatically extracted from chord reductions and is not designed to be used in non-tonal musics.

In this section, firstly, we present the main concepts behind the General Chord Type representation and give an overall description, then, we describe the GCT algorithm that automatically computes chord types for each chord, then, we present examples form diverse music idioms that show the potential of the representation and give some examples of applying statistical learning on such a representation, and, finally, we will discuss problems and future improvements.

4.1 Representing Chords

Harmonic analysis focuses on describing the harmonic content of pitch collections/patterns within a given music context in terms of harmonic labels, classes, functions and so on. Harmonic analysis is a rather complex musical task that involves not only finding roots and labeling chords within a key, but also segmentation (points of harmonic change), identification of non-chord notes, metric information and more generally musical context [14]. In this section, we focus on the core problem of labeling chords within a given pitch hierarchy (e.g. key); thus we assume that a full harmonic reduction is available as input to the model (manually constructed harmonic reductions).

Our intention is to create an analytic system that may label any pitch collection, based on a set of user-defined criteria rather than on standard tonal music theoretic models or fixed psychoacoustic properties of harmonic tones. We intend our representation to be able to cope with chords not only in the tonal system, but any harmonic system (e.g. octatonic, whole-tone, atonal, traditional harmonic systems, etc.).

Root–finding is a core harmonic problem addressed primarily following two approaches: the standard stack-of-thirds approach and the virtual pitch approach. The first attempts to re-order chord notes such that they are separated by (major or minor) third intervals preserving the most compact ordering of the chord; these stacks of thirds can then be used to identify the possible root of a chord (see, for instance, recent advanced proposal by [29]). The second approach, is based on Terhard's virtual pitch theory [33] and Parncutt's psychoacoustic model of harmony [26]; it maintains that the root of a chord is the pitch most strongly implied by the combined harmonics of all its constituent notes (intervals derived from the first members of the harmonic series are considered as "root supporting intervals").

Both of these approaches rely on a fixed theory of consonance and a fixed set of intervals that are considered as building blocks of chords. In the culture-sensitive stack-of-thirds approach, the smallest consonant intervals in tonal music, i.e. the major and minor thirds, are the basis of the system. In the second "universal" psychoacoustic approach, the following intervals, in decreasing order of importance, are employed: unison, perfect fifth, major third, minor seventh, and major second. Both of these approaches are geared towards tonal harmony, each with its strengths and weaknesses (for instance, the second approach has an inherent difficulty with minor harmonies). Neither of them can be readily extended to other idiosyncratic harmonic systems.

Harmonic consonance/dissonance has two major components: Sensory-based dissonance (psychoacoustic component) and music-idiom-based dissonance (cultural component) [23]. Due to the music-idiom dependency component, it is not possible to have a fixed universal model of harmonic consonance/dissonance. A classification of intervals into categories across the dissonance-consonance continuum can be made only for a specific idiom. The most elementary classification is into two basic categories: consonant and dissonant. For instance, in the common-practice tonal system, unisons, octaves, perfect fifths/fourths (perfect consonances) and thirds and sixths (imperfect consonances) are considered to be consonances, whereas the rest of the intervals (seconds, sevenths, tritone) are considered to be dissonances; in polyphonic singing from Epirus, major seconds and minor sevenths may additionally be considered "consonant" as they appear in metrically strong positions and require no resolution; in atonal music, all intervals may be considered equally "consonant".

Let's examine the case of tonal and atonal harmony; these are probably as different as two harmonic spaces may be. In the case of tonal and atonal harmony, some basic concepts are shared; however, actual systematic descriptions of chord-types and categories are drastically different (if not incompatible), rendering any attempt to "align" two input spaces challenging and possibly misleading (Figure 5). On one hand, tonal harmony uses a limited set of basic chord types (major, minor, diminished, augmented) with extensions (7ths, 9ths etc.) that have roots positioned in relation to scale degrees and the tonic, reflecting the hierarchic nature of tonal harmony; on the other hand, atonal harmony employs a flat mathematical formalism that encodes pitches as pitch-

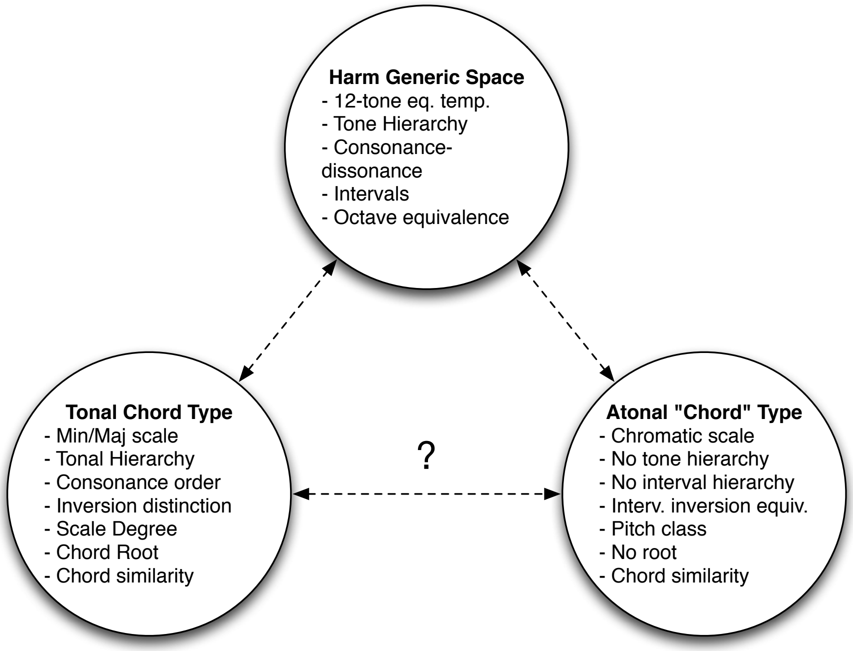

Figure 5: Is mapping between “opposing” harmonic spaces possible?

class sets leaving aside any notion of pitch hierarchy, tone centres or more abstract chord categories and functions. It seems as if it is two worlds apart having as the only meeting point the fact that tones sound together (physically sound-ing together or sounding close to one another allowing implied harmony to emerge).

Pc-set theory of course, being a general mathematical formalism, can be applied to tonal music, but, then its descriptive potential is mutilated and most interesting tonal harmonic relations and functions are lost. For instance, the distinction between major and minor chords is lost if Forte’s prime form is used (037 for both – these two chord have identical interval content), or a dominant seventh chord is confused with half-diminished seventh (prime form 0258); even, if normal order is used, that is less general, for the dominant seventh (0368), the root of the chord is not the 0 on the left of this ordering (pc 8 is the root). Pitch-class set theory is not adequate for tonal music. At the same time, the roman-numeral formalism is inadequate for atonal music as major/minor chords and tonal hierarchies are hardly relevant for atonal music.

In trying to tackle issues of tonal hierarchy, we have devised a novel chord type representation, namely the *General Chord Type* (GCT) representation, that takes as its starting point the common-practice tonal chord representation (for a tonal context, it is equivalent to the standard roman-numeral harmonic encoding), but is more general as it can be applied to other non-standard tonal systems such as modal harmony and, even, atonal harmony. This representation draws on knowledge from the domain of psychoacoustics and music cognition, and, at the same time, “adjusts” to any context of scales, tonal hierarchies and categories of consonance/dissonance.

At the heart of the GCT representation is the idea that the “base” of a note simultaneity should be consonant. The GCT algorithm tries to find a maximal subset that is consonant; the rest of the notes that create dissonant intervals to one or notes of the chord "base" form the chord "extension". The GCT representation has common characteristics with the stack-of-thirds and the virtual pitch root finding methods for tonal music, but has differences as well (see section 4.3). Moreover, the user can define which intervals are considered "consonant" giving thus rise to different encodings. As will be shown in the next sections, the GCT representation encapsulates naturally the structure of tonal chords and at the same time is very flexible and can readily be adapted to different harmonic systems.

4.2 The General Chord Type representation

4.2.1 Description of the GCT algorithm

Given a classification of intervals into consonant/dissonant (binary values) and an appropriate scale background (i.e. scale with tonic), the GCT algorithm computes, for a given multi-tone simultaneity, the "optimal" ordering of pitches such that a maximal subset of consonant intervals appears at the "base" of the ordering (left-hand side) in the most compact form. Since a tonal centre (key) is given, the position within the given scale is automatically calculated.

Input to the algorithm is the following:

- Consonance vector: The user defines which intervals are consonant/dissonant through a 12-point Boolean vector of consonant (1) or dissonant (0) intervals. For instance, the vector $[1,0,0,1,1,1,0,1,1,1,0,0]$ means that the unison, minor and major third, perfect fourth and fifth, minor and major sixth intervals are consonant – dissonant intervals are the seconds, sevenths and the tritone; this specific vector is referred to in this text as the common-practice consonance vector.

- Pitch Scale Hierarchy: The pitch hierarchy (if any) is given in the form of scale tones and a tonic (e.g. a D maj scale is given as: $2,[0,2,4,5,7,9,11]$, or an A minor pentatonic scale as: $9,[0,3,5,7,10]$).

- Input chord: list of MIDI pitch numbers (converted to pc-set).

Algorithm 1 GCT algorithm (core) – computational pseudocode

Input: (**i**) the pitch scale (tonality), (**ii**) a vector of the intervals considered consonant, (**iii**) the pitch class set (pc–set) of a note simultaneity

Output: The roots and types of the possible chords describing the simultaneity

find all maximal subsets of pairwise consonant tones
for all selected maximal subsets **do**
    order the pitch classes of each maximal subset in the most compact form (chord "base")
    add the remaining pitch classes (chord extensions) above the highest of the chosen maximal subset's (if necessary, add octave – pitches may exceed the octave range)
    the lowest tone of the chord is the "root"
    transpose the tones of the chord so that the lowest becomes 0
    the lowest tone of the chord is the "root"
    transpose the tones of the chord so that the lowest becomes 0
    find position of the "root" in regards to the given tonal centre (pitch scale)
end for

The GCT algorithm encodes most chord types "correctly" in the standard tonal system. In example 1, Table 5 the note simultaneity $[C,D,F\sharp,A]$ or $[0,2,6,9]$ in a G major key is interpreted as $[7,[0,4,7,10]]$, i.e. as a dominant seventh chord (see similar example in Section 4.2.3).

However, the algorithm is undecided in some cases, and even makes "mistakes" in other cases. In most instances of multiple encodings, it is suggested that these ideally should be resolved by taking into account other harmonic factors (e.g., bass line, harmonic functions, tonal context, etc.). For instance, the algorithm gives two possible encodings for a $[0,2,5,9]$ pc-set, namely minor seventh chord or major chord with sixth (see Table 5, example 2); such ambiguity may be resolved if tonal context is taken into account. For the $[0,3,4,7]$ pc-set with root 0, the algorithm produces two answers, namely, a major chord with extension $[0,[0,4,7,15]]$ and a minor chord with extension $[0,[0,3,7,16]]$; this ambiguity may be resolved if key context is taken into account: for in-stance, $[0,4,7,15]$ would be selected in a C major or G major context and $[0,3,7,16]$ in a C minor or F minor context. Symmetric chords, such as the augmented chord or the diminished seventh chord, are inherently ambiguous; the algorithm suggests multiple encodings which can be resolved only by taking into account the broader harmonic context (see Table 5, example 3). Since the aim of this algorithm is not to perform sophisticated harmonic analysis, but rather to find a practical and efficient encoding for tone simultaneities (to be used, for instance, in statistical learning and automatic harmonic generation – see end of Section 4), we decided to extend the algorithm so as to reach in every case a single chord type for each simultaneity (no ambiguity).

Table 5: Examples of applying the GCT algorithm.

		example 1	example 2	example 3
	tonality – key	G: [7, [0, 2, 4, 5, 7, 9, 11]]	C: [0, [0, 2, 4, 5, 7, 9, 11]]	C: [0, [0, 2, 4, 5, 7, 9, 11]]
	consonance vector	[1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0]	[1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0]	[1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0]
	input	[60, 62, 66, 69, 74]	[50, 60, 62, 65, 69]	[62, 68, 77, 71]
	pc–set	[0, 2, 6, 9]	[0, 2, 5, 9]	[2, 5, 8, 11]
	maximal subsets	[2, 6, 9]	[2, 5, 9] and [0, 5, 9]	[2, 5], [5, 8], [8, 11], [2, 11]
	add extensions	[2, 6, 9, **12**]	[2, 5, 9, **12**] and [5, 9, 0, **14**]	all rotations of [2, 5, 8, 11]
	lowest is root	2 (note D)	2 and 5	2,5,8, 11 (resp. for each rotation)
	chord in root position	[2, [0, 4, 7, 10]]	[2, [0, 3, 7, 10]] and [5, [0, 4, 7, 9]]	[X, [0, 3, 6, 9]], X $\in$ {2, 5, 8, 11}
	relative to key	[7, [0, 4, 7, 10]]	[0, [0, 3, 7, 10]] and [3, [0, 4, 7, 9]]	X $\in$ {2, 5, 8, 11}
extra	Subset overap		[2, [0, 3, 7, 10]]	
steps:	Base in scale			[11, [0, 3, 6, 9]]

Algorithm 2 GCT algorithm (additional steps) – for unique encoding

Input: Multiple maximal subsets-encodings

Output: Unique chord encoding

if more than one maximal subsets exist: **then**

    Overlapping of maximal subsets: create a sequence of maximal subsets by ordering them so as to have maximal overlapping between them and keep the maximal subset that appears first in the sequence (chord's base)

    Chord base notes are scale notes: prefer maximal subset that contains only pcs that appear in the given scale (tonal context) – i.e. avoid non-scale notes in the chord base (this rule is rather arbitrary and is under consideration)

    if neither of the above give a unique solution, chose one encoding at random

end if

Additional adjustment: for dyads, in a tonal context, prefer perfect fifth over perfect fourth, and prefer seventh to second intervals

The additional steps select chord type $[2,[0,3,7,10]]$ in example 2, Table 5 (maximal overlapping between two maximal subsets), and $[11,[0,3,6,9]]$ in example 3, Table 5 (last pitch-class is A♭ that is a non-scale degree in C major).

4.2.2 Formal description of the Core GCT Algorithm

The proposed algorithm for computing the GCT, receives a simultaneity of pitches that are transformed into pitch classes and produces chord elements, namely the *root*, the *type* and the *extension*, which specify qualitative information about the chord that more precisely describes this simultaneity. A detailed description of the algorithm follows, based on an example input simultaneity. Suppose that the input set of notes results in the pc-set $[0,2,6,9]$, which could be described as a D major chord with flat seventh regarding the tonal music consonance environment – described by the $\vec{v} = (1,0,0,1,1,1,0,1,1,1,0,0)$ consonance vector. Therefore, the algorithm should produce and output in the form: $[\mathrm{R},[\mathrm{T}],[\mathrm{E}]] = [2,[0,4,7],[10]]$. For the rest of the paper, the i–th element of a vector $\vec{v}$ will be denoted as $\vec{v}_i$.

By utilising the input pc-set and given a consonance vector that represents a selected music idiom (in this example the consonance vector is $\vec{v} = (1,0,0,1,1,1,0,1,1,1,0,0)$), a binary matrix is constructed that is denoted as B. Each row and column of B represents a pitch class of the input chord, while a matrix entry is 1 or 0, signifying whether the pair of row and column pcs are consonant or dissonant respectively – according to the current consonance vector. Strictly, if the consonance vector is denoted as $\vec{v}$ and the input pcset as $\vec{p}$, then $\forall i,j \in \{1,2,\ldots,\mathrm{length}(\vec{p})\}$

$$B_{ij} = c_{|p_i - p_j|}, \tag{1}$$

where the function length$(\vec{x})$ return the length of vector $\vec{x}$. The B matrix in the discussed example, where $\vec{p} = [0,2,6,9]$, is the following:

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{2}$$

Afterwards, a tree is constructed for each of the rows of B. The root node of these trees is the pitch class that corresponds to the respective row, while their branches from leaves to nodes include pitch classes that are pairwise consonant (according to $\vec{v}$). The construction of the tree that corresponds to the i–th element of $\vec{p}$, is implemented by recursively traversing B in a depth–first–search (DFS) fashion, beginning from the i–th row and following the paths "circumscribed" by the occurrences of units. Such a traversal is exhibited in Table 6 for the second row of the current example's B matrix. This step's outcome is a collection of trees, each of which corresponds to a row of B. The trees of the current example are shown in Table 7.

Table 6: The steps of the algorithm when scanning the path of the second row.

line 2 to column 3	evaluating to top	line 3 to column 4	evaluating to top
$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & \mathbf{1} & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & \boxed{1} & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & \mathbf{1} \\ 1 & 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & \boxed{1} \\ 0 & 1 & 1 & \boxed{1} \\ 1 & 1 & 1 & 1 \end{pmatrix}$
$\begin{array}{c} 2 \\ 6 \end{array}$	$\begin{array}{c} 2 \\ 6 \end{array}$	$\begin{array}{c} 2 \\ 6 \\ 9 \end{array}$	$\begin{array}{c} 2 \\ 6 \\ 9 \end{array}$

$$\vdots$$

$$\begin{array}{cc} \multicolumn{2}{c}{2} \\ 6 & 9 \\ 9 & \end{array}$$

Table 7: All the trees for the current example. The maximal path is highlighted with boldface typesetting.

line 1	line 2	line 3	line 4
$\begin{array}{c} 0 \\ 9 \end{array}$	$\begin{array}{cc} \multicolumn{2}{c}{\mathbf{2}} \\ \mathbf{6} & 9 \\ \mathbf{9} & \end{array}$	$\begin{array}{c} 6 \\ 9 \end{array}$	9

After the application of the above procedure, the paths from root to leaves with maximal length are kept either as the output chord candidates, or for further processing in the steps described in the remaining of this section. In the current example there is a single maximal path ($[2,6,9]$), which is highlighted with boldface typesetting. After the longest path has been extracted, the pitch classes that constitute it are recombined in their most compact form, which in the current example is $[2,6,9]$ (unaltered). The pitch class 0 of the initial $[2,6,9]$ pc-set is considered as an *extension*. Thereby, the simultaneity $[0,2,6,9]$ is circularly shifted to $[2,6,9,12]$, disregarding the fact that pitch classes can take integer values between 0 and 11. In turn, $[2,6,9,12]$ is transformed to the following $[\mathtt{r},[\mathtt{t}],[\mathtt{e}]]$ denotation: $[2,[0,4,7],[10]]$. This denotation clarifies that the simul-

J.S.Bach - Chorale 54 (Lobt Gott, ihr Christen, allzugleich) in G major - 2nd phrase

Roman Numeral Analysis:							
	D major	I^6	vii_o^6	I	ii^6	V^7	I
GCT Analysis (tonal major profile)							
	2,[0,2,4,5,7,9,11]	0,[0,4,7]	11,[0,3,6]	0,[0,4,7]	2,[0,3,7]	7,[0,4,7,10]	0,[0,4,7]
Pc-Set Analysis (chromatic scale):							
	normal orders	[0,4,7]	[0,3,6]	[0,4,7]	[0,3,7]	[0,2,6,9]	[0,4,7]
	prime forms	[0,3,7]	[0,3,6]	[0,3,7]	[0,3,7]	[0,3,6,8]	[0,3,7]
GCT Analysis (atonal profile)							
	[0,1,2,3,4,5,6,7,8,9,10,11]	2,[0,4,7]	1,[0,3,6]	0,[0,4,7]	4,[0,3,7]	7,[0,2,6,9]	2,[0,4,7]

Figure 6: Chord analysis of a Bach Chorale phrase by means of traditional roman numeral analysis, pc-sets and two versions of the GCT algorithm.

taneity $[0,2,6,9]$ is actually a major chord (type $[0,4,7]$) with a flat seventh (extension $[10]$) and fundamental pitch class 2, (i.e. D7). As the tonal context is given as input, for instance G major key, the absolute chord type $[2,[0,4,7,10]]$ (i.e. D7 chord) is converted to relative chord type, i.e., $[7,[0,4,7,10]]$ which means dominant seventh in G major. This is equivalent to the roman numeral analytic types.

4.2.3 An example analysis with GCT

An example harmonic analysis of a Bach Chorale phrase illustrates the proposed GCT chord representation (Figure 6). For a tonal context, chord types are optimised such that pcs at the left hand side of chords contain only consonant intervals (i.e. 3rds and 6ths, and Perfect 4ths and 5ths). For instance, the major 7th chord is written as $[0,4,7,10]$ since set $[0,4,7]$ contains only consonant intervals whereas 10 that introduces dissonances is placed on the right-hand side – this way the relationship between major chords and major seventh chords remains rather transparent and is easily detectable. Within the given D major key context it is simple to determine the position of a chord type in respect to the tonic – e.g. $[7,[0,4,7,10]]$ means a major seventh chord whose root is 7 semitones above the tonic, amounting to a dominant seventh. This way we have an encoding that is analogous to the standard roman numeral encoding (Figure 6, top row). If the tonal context is changed, and we have a chromatic scale context (arbitrary "tonic" is 0, i.e. note C) and we consider all intervals equally "consonant", we get the second GCT analysis in Figure 6 which amounts to normal orders (not prime forms) in a standard pc-set analysis – for tonal music this pc-set analysis is weak as it misses out important tonal hierarchical relationships (notice that the relation of the dominant seventh chord type to the plain dominant chord is obscured). Note that relative "roots" to the "tonic" 0 are preserved as they can be used in harmonic generation tasks.

For practical reasons of space in the musical illustrations, the form $[r,[b],[e]]$ is not preserved: the base and extension is concatenated and brackets are omitted. For instance: $[7,[0,4,7],[10]]$

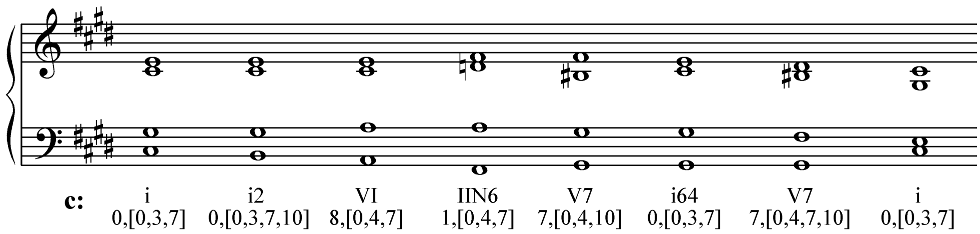

Figure 7: Beethoven, Sonata 14, op.27-2 (reduction of first five measures). Top row: roman numeral harmonic analysis; bottom row: GCT analysis. GCT successfully encodes all chords, including the Neapolitan sixth chord (fourth chord).

may be depicted as $7,[0,4,7,10]$ or even as 7.04710.

4.3 Harmonic encoding and analysis with the GCT

The GCT algorithm has been applied to tonal extracts from standard tonal pieces, such as Bach Chorales, but additionally it has been tested out on harmonic structures from diverse harmonic idioms. Some examples are presented below to give an idea of the potential of the GCT representation. Strong points of the encoding are given along with weaknesses. Some aspects of the analysis are difficult to judge in some idioms and further study in required.

4.3.1 GCT Encoding Examples

In common-practice tonal music, GCT works very well. Mistakes are sometimes made in case of symmetric chords such as the diminished seventh chord or the augmented triad. In the case of the half diminished seventh chord GCT "prefers" to label it as a minor chord with added sixth instead of a diminished chord with minor seventh. Chords that include chromatic notes such as the German sixth, Italian sixth, Neapolitan sixth are encoded consistently even though not necessarily coinciding with analytic interpretations by theorists (the French sixth is more tricky as it is a symmetric chord and GCT finds two equally prominent "roots").

Below, a number of examples are presented that illustrate the application of the GCT algorithm on diverse harmonic textures. The first example (Figure 7) is taken from the first measures of Beethoven's Moonlight Sonata. In this example, GCT encodes classical harmony in a straightforward manner. All instances of the tonic chord inverted or not (i.e. C♯minor) are tagged as $[0,[0,3,7]]$ and $[10]$ is added when the 7^{th} is present; the dominant seventh is $[7,[0,4,7,10]]$ and once it appears once without the fifth $[7]$; the fourth chord is a Neapolitan sixth and it is encoded as $[1,[0,4,7]]$ which means major chord on lowered second degree (D major chord in the C♯ minor key).

In the example of Figure 8 a tonal chord progression by G. Gershwin is presented. More chromaticism is apparent in this passage. The GCT "agrees" with the roman numeral analysis of the excerpt including the Italian sixth chord that is labelled as $[8,[0,4,10]]$, and it even labels the chord that was left without a roman numeral tag by the analyst (see question mark) encoding it as a minor chord with sixth on the flattened sixth degree (G♭-B♭♭-D♭-E♭) (Note: actually it could be

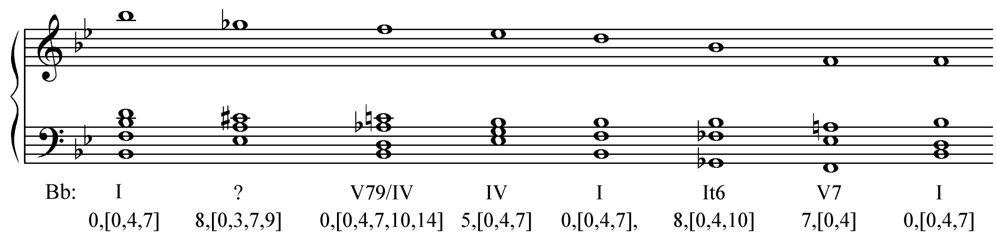

Figure 8: G. Gershwin, Rhapsody in Blue (reduction of first five measures). Top row: roman numeral harmonic analysis; bottom row: GCT analysis. GCT successfully identifies all chords, including the third secondary dominant and the second-to-last augmented sixth (Italian) chord. Additionally it labels the second chord as a minor chord with added sixth on the flat VI degree (G♭-B♭♭-D♭-E♭).

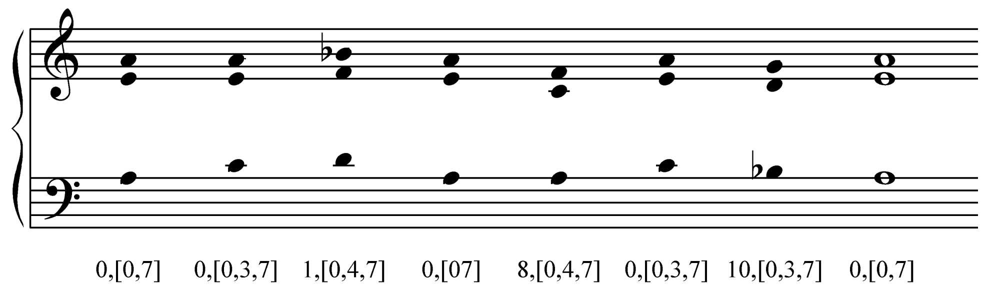

Figure 9: G. Dufay's *Kyrie* (reduction) – first phrase in A phrygian mode that exemplifies parallel motion in fauxbourdon and a phrygian cadence (early Renaissance) . GCT correctly identifies and labels the open fifths as well as the triadic chords.

even encoded as a half-diminished 7$^{\text{th}}$ on the fourth degree E♭-G♭-B♭♭-D♭).

Figure 9 illustrates an Early Renaissance example of fauxbourdon by G. Dufay. Parallel motion of voices is typical in this idiom. The GCT labels correctly all dyads and triads, taking into account musica ficta that produces rather unusual chord progressions in regards to standard tonal harmony.

In Figure 10 an example from the polyphonic singing tradition of Epirus is presented. This very old 2-voice to 4-voice polyphonic singing tradition is based on the anhemitonic pentatonic pitch collection and more specifically the pentatonic minor scale that functions as source for both the melodic and harmonic content of the music. A unique harmonic aspect of these songs is the unresolved dissonances (major second and minor seventh intervals) at structurally stable positions of the pieces (e.g. cadences). In the example two GCT version are presented: the first (top row) depicts the encoding for the standard consonance vector and the second (bottom row) presents the GCT labelling that considers additionally major seconds and minor sevenths as "consonant" (it is the same as for the "atonal" consonance vector as no minor seconds and major sevenths exist in the idiom). It is interesting to note that for the standard consonance vector almost all chords have the drone tone as their root. On the other hand, in the second encoding different relations between chords become apparent (e.g. $[10,[0,2,5]]$ and $[10,[0,2,5,7]]$) and also an oscillation of the chord "root" between the tonic and a note a tone lower is highlighted. Polyphonic songs from Epirus are

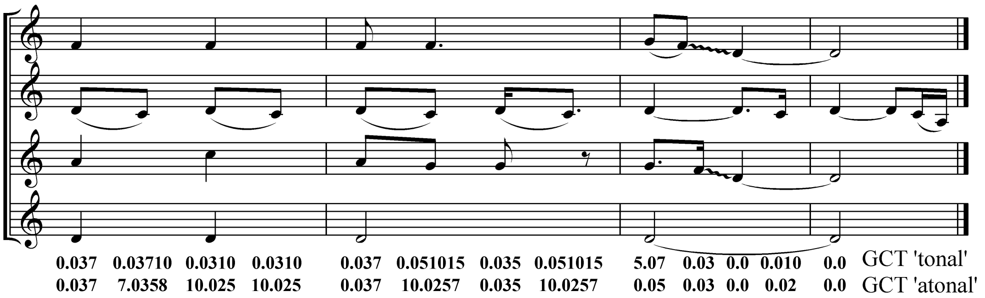

Figure 10: Excerpt from a traditional polyphonic song from Epirus. Top row: GCT encoding for standard common-practice consonance vector; bottom row: GCT encoding for atonal harmony – all intervals “consonant” (this amounts to pc-set “normal orders”).

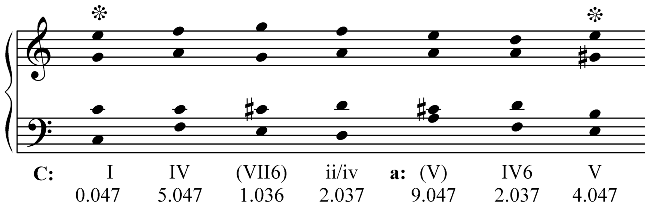

Figure 11: Automatically generated GCTs for a Bach Chorale melody employing a HMM for fixed boundaries (first and last chords are given). Voice leading has been arranged manually.

studied more extensively in a different study (REFERENCE for FMA2014 – forthcoming).

4.3.2 Learning and generation with GCT

In a current study, the GCT representation has been utilised in automatically analysing and encoding scores (actually, harmonic reductions of scores) from diverse idioms, and then employing this extracted information for melodic harmonisation. In [19] the authors discuss the utilization of a well-studied probabilistic methodology, namely, the hidden Markov model (HMM) methodology, in combination with constraints that incorporate fixed beginning, ending and/or intermediate anchor chords. To this end, a constrained HMM (CHMM) is developed, which allows the manual or deterministic insertion of intermediate chords, providing alternative harmonisations that comply with specific constraints.

The reported results indicate that the CHMM method, harnessed with the novel General Chord Type (GCT) algorithm, functions effectively towards convincing melodic harmonisations in diverse idioms. In Figures 7 and 8, two examples of melodic harmonisation are illustrated for a Bach chorale melody and for a traditional melody from Epirus. In both cases, the system has been trained on a corpus of harmonic reductions of pieces in the idiom, and, then, used to generate new melodic harmonisations. The results are very good: the Bach chorale harmonisation is typical of the style and at the same time not trivial (uses secondary dominants that enrich the harmonisation); the Epirus melody harmonisation is close to the style of polyphonic singing (if additional melodic and rhythmic elements were added the phrase would become rather typical of the idiom).

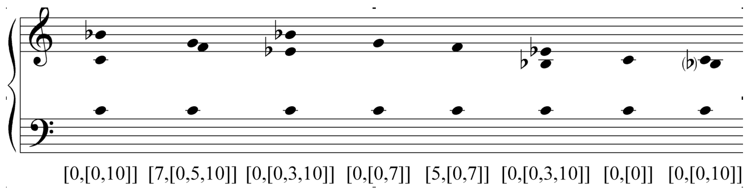

Figure 12: Automatically generated GCTs for an Epirus melody (reduced version) employing a HMM for fixed boundaries. Voice leading has been arranged manually.

4.3.3 Discussion and future development

The current version of GCT encodes only the chord type and the relative position of its “root” to the local tonic of a given scale. However, it can readily be extended to in-corporate explicit information on chord inversions (i.e. bass note position), on scale degrees (chromatic notes that do not belong to the current scale can be tagged so that indirectly scale degrees are indicated), and, even, on voice-leading (for instance, motion of bass, or even for note extensions that may require resolution by down-wards step-wise motion). A rich chord representation should embody such information.

The organisation of tones by GCT for the “standard” consonance vector gives results quite close to those produced by the stack-of-thirds technique, as implicit in the latter is consonance of thirds and fifths (as two thirds sum up to a fifth). Some difference are:

- the stack-of-thirds approach usually requires traditional note names (that allow enharmonic spellings) whereas the GCT is based on pitch classes (no direct explicit link to a scale). For instance, GCT considers the chord CEG♯ or CEA♭ ($[0,4,8]$) as consonant since its intervals are pairwise consonant[7], i.e. two 4 semitone intervals (major thirds) and one 8 semitone interval (minor sixth or augmented fifth) with root any one of the three tones; stack-of-thirds determines C as the root in the first case and A♭ in the second case. The GCT algorithm misses out on sophisticated tonal scale information but is still informative at the same time being simpler, and easier to implement.
- in the standard consonance vector version of GCT, diminished fifths are not allowed whereas in the stack-of-thirds approach all fifths are allowed. For instance, the root of the half-diminished chord BDFA is B according to the stack-of-thirds whereas GCT considers D as the root and B as a sixth above the root (DFAB), i.e. diminished triads are not consonant chords according to CGT. Of course, the consonance vector in GCT may be altered so that the tritone is also consonant in which case the two approaches are closer.
- the stack-of-thirds method allows empty third positions in the lower part of the stack whereas GCT always prefers to have a compact consonant set of pitches at the bottom. For instance, a chord comprising of notes: CEFG ($[0,4,5,7]$) will be arranged as FCEG by the stack-of-thirds technique and CEGF ($[0,4,7,17]$) by GCT.

In relation to the virtual pitch root finding method, the proposed approach differs in that minor thirds are equally consonant to major thirds allowing equal treatment of major and minor chord

[7]Question: why is the augmented triad considered dissonant when all its tones are pairwise consonant?

(as opposed to the virtual pitch approach that is biased towards major thirds due to the structure of the harmonic series).

It is also possible to redesign the GCT algorithm altogether so as to make use of non-binary consonance/dissonance values allowing thus a more refined consonance vector. Instead of filling in the consonance vector with 0s and 1s, it can be filled with fractional values that reflect degrees of consonance derived from perceptual experiments (e.g., [21]) or values that reflect culturally-specific preferences. Such may improve the algorithm's performance and resolve ambiguities in certain cases (future work).

4.4 Concluding remarks for the GCT

In this paper a new representation of chord types has been presented that adapts to diverse harmonic idioms allowing the analysis and labeling of tone simultaneities in any harmonic context. The General Chord Type (GCT) representation, allows the rearrangement of the notes of a harmonic simultaneity such that idiom-specific types of chords may be derived. Given a consonance/dissonance classification of intervals (that reflects culturally-dependent notions of consonance/dissonance), and a (set of) scales, the GCT algorithm finds the maximal subset of notes of a given note simultaneity that contains only consonant intervals; this maximal subset forms the basis up-on which the chord type is built. The proposed representation is ideal for hierarchic harmonic systems such as the tonal system and its many variations, but adjusts to any other harmonic system such as post-tonal, atonal music, or traditional polyphonic systems.

The GCT representation was applied to a small set of examples from diverse musical idioms, and its output was presented and analysed showing its potential use, especially, for computational music analysis and music in-formation retrieval tasks. The encoding provided by GCT is not always correct according to the interpretation given by music theorists, but, at least, it is consistent (i.e. a certain chord will always be encoded the same way) rendering it adequate for machine learning and generation (e.g. melodic harmonisation) where music theoretical correctness is not so important. Sometimes GCT "uncovers" chordal relations that are obscured by notation and en-harmonic spellings, and may assist a musician in harmonic analysis. Overall, the proposed encoding seems to be promising and potentially useful in computational music applications.

5 Future perspectives

This section discusses the future perspectives on two different domains: a) the utilization of the dataset's extracted components towards achieving conceptual blending on various levels of harmony and b) the tools to operate the harmonic training dataset – given the requirements for extracting concepts towards blending. Inevitably, the discussion reaches the borders of algorithmic design for conceptual blending in harmony, deviating from the main subject that is the training dataset and its operation. However, the target of the following paragraphs is to further justify the selection of the dataset material and the decision to develop a custom-made music encoding scheme.

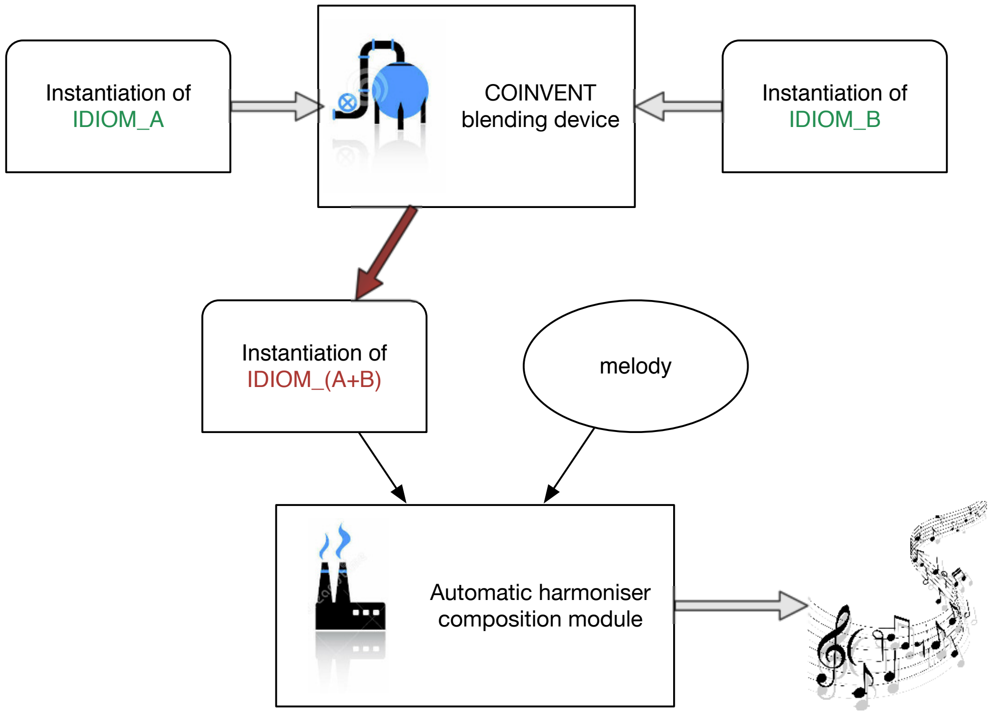

Figure 13: Overview illustration of the COINVENT melodic harmonizer. Conceptual blending takes part in the ontologies that describe harmony, creating blended harmonizations that constitute the descriptive guidelines for the composition module.

5.1 Dataset components and conceptual blending

The development of the melodic harmonizer is designed to encompass two main parts: the *blending* and the *harmonization* modules. This scheme is roughly illustrated in Figure 13. The blending module incorporates all the background knowledge in the form of harmonic ontologies that represent different idioms, as extracted form the pieces comprising the harmonic training dataset. Harmonic blending concerns the information included in this module, while the richness of background information is provided by the diversity of idioms in the dataset, as well as by the extracted components of the dataset pieces. The harmonization module receives information from the blending module, thereby building a harmonization framework of deterministically produced chord constraints. Given the chord constraints, the cHMM [19] methodology is employed to fill-in the remaining chords, while the harmonization is completed with the determination the voice leading however, composition and blending specifics are beyond the scope of this report.

Manual annotations concern the extraction of key-elements in harmony, from a harmonically reduced version of the examined pieces' phrases. These elements include the required information to describe harmonic concepts on many levels, with a parallel goal to achieve a balance between human-friendly interpretation of data and computationally accessible encoding. Specifically, the harmonic levels that are currently isolated and represented are the following (the first two of which are also described in Section 3.3):

1. *Tonality* and tonality changes, which are important for tracking the tonality constitution of phrases within idioms (e.g. modal or tonal harmony), while also determining the harmonic changes that occur and how often they occur.
2. *Grouping* of phrases for tracing the sub-phrases that include harmonically and melodically coherent parts.
3. Determination of *harmonic devices*, for demarcating areas of specific harmonic mechanisms or concepts.

More specifically, the *harmonic devices* are unique or distinctive harmonic areas within pieces or idioms, that can be described by themselves as musical concepts. For instance, the *perfect cadence* can be considered as a harmonic device, which is a characteristic harmonic element of tonal music. Other examples of harmonic devices are parts where chords in specific tonal environments are used (e.g. in the whole-tone or other symmetric scales), or parts with specific chord movement (e.g. parallel diatonic or chromatic movement) among others.

Before analyzing the utilization of harmonic devices in conceptual blending, a rough description of the levels that blending takes place is required:

1. *Meta-level* of harmony: This level discusses blending on some aspects of harmony that are described either as general properties of harmony (e.g. the mood) or even high-level harmonic features as harmonic parts that adhere to specific conceptual characteristics (e.g. chromatic voice leading of a particular voice in a chord sequence).
2. *Phrase structure*: Phrase structure refers to the harmonic blocks that succeed one another, including cadences or the ordering of parts that belong to harmonic devices (or harmonic device combinations).
3. *Chord level*: Blending in the chord level yields on the one hand novel chords that preserve some crucial harmonic elements, regarding the relations between notes of successive chords. On the other hand, chord blending allows the invention of intermediate chords for consecutive harmonic parts that belong to different conceptual spaces, providing common-ground solutions for "incompatible" consecutive parts between different cHMMs.

Additionally, there is a key algorithmic process that has been developed in the context of COINVENT that allows the idiom–dependent probabilistic harmonization of different parts, preserving the harmonic characteristics of a selected idiom. This technique is based on the well-known hidden Markov models (HMMs), with the vital difference that they allow the deterministic insertion of intermediate chords, thus allowing the functionality of the hitherto described blending framework.

In the level of phrase structure, probabilistic grammars provide intermediate chords that signify important structural part, like phrase endings (cadences) on different levels of phrase grouping. Previous work on grammars for harmony indicate that harmony in an idiom is a highly specialized characteristic and, therefore, the development of an idiom's grammar requires a deep musicological analysis of this idiom. For instance, a highly specialized grammars for harmony was presented in [10], which discussed the employment of combinatory categorial grammars to describe the harmony in jazz standards. This grammar allowed to either automatically parse, or generate harmonic parts of the jazz idiom. A large number of rules was included in this grammar, describing highly specialized harmonic devices of the jazz idiom, e.g. special types of cadences like tritone substitution and backdoor cadence. Additionally, the highly specialized nature of harmony is further amplified by the of the vast complexity of the rules that are employed in works that examine the development of grammars for the probably deepest studied musical idiom: the tonal music(e.g. [28, 20]).

Grammars constitute an effective methodological approach to represent information that is layered in a hierarchical manner. The structure of a grammar employes rules that describe symbol substitutions from upper to lower hierarchical levels, reaching the lowest level of terminal symbols that represent the grammar's output string. However, the development of harmonic grammars that are accurate enough to capture the basic characteristics of a musical idiom, is a task that requires deep knowledge and ready-made musicological analysis of the harmonic elements and their relation in this idiom. Therefore, the development of musical grammars incorporates on the one hand a process of identifying the main components that will be represented (e.g. phrase body, phrase endings and tonality modulations among others) and on the other hand the construction of a probabilistic set of rewrite rules that describe the successions of these components. To this end, the annotation of grouping structure and tonality are vitally important for the construction of the phrase grammars, allowing conceptual blending to foster novel combinations of phrasing parts from different idioms, on different levels of phrase grouping.

5.2 Dataset operation framework perspectives

The previous discussion (in Section 5.1) exposed some additional requirements that the data operation processes (annotation and extraction) must fulfill. Among the first additional requirements is that the annotation process of novel concepts – as isolated parts on score – should become expandable: a musician/musicologist should be allowed to easily annotate the positions where a special concept (or harmonic device) begins and ends. Afterwards, it should also be easy to retrieve the parts related to some additional annotations, with a query-search process.

For instance, in Section 3.3, the process of interpreting annotations to their harmonic meaning is described, where the entries of a *dictionary* translate musical parts into harmonic entities. In the aforementioned section, examples are given for the interpretation of tonality (and tonality changes) and grouping structure through the addition of extra staves in the score of the examined musical phrase. In the current development stage of the dataset management software, the annotation of tonality and grouping are *hard-coded*, in a sense that the incorporated dictionaries (for tonality and grouping) are unmodifiable. To this end, the first-in-priority goal for the future development of the dataset operation framework, is to allow the creation of custom made dictionaries but in an efficient and user-friendly manner.

To picture the potential utilization of the dictionaries framework under development, some examples are given that refer to specific harmonic devices, delineating the employment of dictionaries for retrieving harmonic information:

- *Special chord passages*: employment of chords in specific tonal environments (e.g. whole-tone, octatonic). An extra staff could be included to the piece's staves, where different notes in this staff would indicate a predefined – from the dictionary – special tonal environment (e.g. C could indicate the whole-tone scale, while G the octatonic).

- *Special voice leading device*: employment of chords that adhere to specific voice leading constraints. For instance, the extra staff that describes voice leading would be followed by a dictionary where the notes, e.g., C could indicate the existence of a bass-note drone, D the existence of a middle voice drone, E chromatic ascending movement on the bass, etc.

The above mentioned examples are just indicative of the potentials that the proposed framework offers. It should be noticed that the utilization of more advanced dictionaries, with "polyphonic" indications (instead of single-note translations) should offer a greater variety of annotation possibilities, allowing the description of harmonic elements in any considerable detail. Thereby, searching for phrase parts with "composite" harmonic characteristics is enabled, allowing the isolation and examination of harmonic parts that satisfy more detailed constraints. Examples of queries that would return such parts are the following:

- all chord in strong metric positions
- get all chords from surface reduction in phrases with specific tonality
- find the extension position (from 7th and above)
- find all the descending harmonic voice movements that consist of at least 3 chords in the metric span of a measure
- get all tonal deviations
- get all the cadences from phrases with specific tonality
- find all the pieces where there is a modulation between relative scales (I-vi, i-III)

6 Conclusions

The collection of a diverse and extensive harmonic training dataset, along with the implementation of effective and extendable computational tools for deriving information from it, are of vital importance towards the development of the COINVENT melodic harmonization assistant. Thereby, a large availability of harmonic concepts is achieved from many musical idioms, while the blending of these concepts yields hybrid harmonic concepts with diverse characteristics, leading to a "computationally creative" – capable of generating novel concepts – melodic harmonizer. This report describes the collection of dataset and the development of an accompanying data management framework that are oriented towards the aforementioned directions, which lead to the construction of a rich harmonic knowledge background, where music primitives together with manually annotated analytical descriptions are encoded and extracted to compose concept descriptions. These concepts are subsequently utilized for statistical learning and rule induction, building the artificial intelligence context that will foster conceptual blending.

The annotations of harmonic concepts are encoded on musical staves, with the utilization of a dictionary that is constructed by the annotator, thus allowing the custom demarcation of areas on-score that incorporate harmonic concepts. The resulting score, with the music surface and the annotation staves, is stored in a musicXML file, a fact that facilitates musical annotations through well-known computational environments (like music21) or music software (like Finale), allowing musicians/musicologists to annotate musical files. Additionally, the musicXML file format is already a data management "standard" for several programming languages, facilitating the extraction of the annotations for computational use.

An additional concern that is discussed in this report is the representation of harmony. The dataset includes diverse idioms with chords (note simultaneities) that can hardly be represented within a common symbolization context, e.g. while alphabetic chord symbols are adequate to represent chords in the chorales of Bach, they cannot describe the pitch simultaneities that appear in atonal music. To this end, the General Chord type has been developed, which is an encoding that is inspired by the standard roman numeral chord type labeling, but is more general and flexible, allowing abstract idiom-specific types of chords may be derived. The development of this chord representation systems is crucial for the construction of the generic spaces that will facilitate conceptual blending between diverse harmonic spaces.

The long-term goals for developing the COINVENT melodic harmonization methodology are shortly presented, justifying the choices made for the compilation of sets and subsets that comprise the dataset, as well as the computation framework for data annotations. Additionally, future insights are given for the enhancement and the inclusion of more features for the data management framework, transforming the data annotation tool to a query-based harmonic concept retrieval engine. Thereby, the extraction of musical parts that include specific harmonic concepts will be available, allowing the system to perform statistical learning and rule induction.

References

[1] CAMBOUROPOULOS, E., KALIAKATSOS-PAPAKOSTAS, M., AND TSOUGRAS, C. An idiom-independent representation of chords for computational music analysis and generation. In *Proceeding of the joint 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC)* (2014), ICMC–SMC 2014, p. (to appear).

[2] CHUAN, C.-H., AND CHEW, E. A hybrid system for automatic generation of style-specific accompaniment. In *Proceedings of the 4th International Joint Workshop on Computational Creativity* (2007), Goldsmiths, University of London.

[3] CUTHBERT, M. S., AND ARIZA, C. Music21: A toolkit for computer-aided musicology and symbolic music data. In *ISMIR* (2010), J. S. Downie and R. C. Veltkamp, Eds., International Society for Music Information Retrieval, pp. 637–642.

[4] CUTHBERT, M. S., ARIZA, C., AND FRIEDLAND, L. Feature extraction and machine learning on symbolic music using the music21 toolkit. In *ISMIR* (2011), A. Klapuri and C. Leider, Eds., University of Miami, pp. 387–392.

[5] DONNELLY, P., AND SHEPPARD, J. Evolving four-part harmony using genetic algorithms. In *Proceedings of the 2011 International Conference on Applications of Evolutionary Computation - Volume Part II* (Berlin, Heidelberg, 2011), EvoApplications'11, Springer-Verlag, pp. 273–282.

[6] Ebcioglu, K. An expert system for harmonizing four-part chorales. *Computer Music Journal 12*, 3 (1988), 43–51.

[7] Forte, A. *The structure of atonal music*. Yale University Press, New Haven, 1973.

[8] Good, M. Musicxml in practice: Issues in translation and analysis. In *First International Conference MAX 2002: Musical Application Using XML* (2002), pp. 47–54.

[9] Good, M. MusicXML for notation and analysis. *The Virtual Score: Representation, Retrieval, Restoration 29*, 1 (Feb. 2005), 70–85.

[10] Granroth-Wilding, M., and Steedman, M. A robust parser-interpreter for jazz chord sequences. *Journal of New Music Research 0*, 0 (2014), 1–20.

[11] Granroth-Wilding, M. T. *Harmonic analysis of music using combinatory categorial grammar*. PhD thesis, Institute for Language, Cognition and Computation School of Informatics University of Edinburgh, Edinburgh, Scotland, Nov. 2013.

[12] Hankinson, A., Roland, P., and Fujinaga, I. The music encoding initiative as a document-encoding framework. In *ISMIR* (2011), A. Klapuri and C. Leider, Eds., University of Miami, pp. 293–298.

[13] Harte, C., Sandler, M., Abdallah, S. A., and Gómez, E. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)* (London, UK, 2005), pp. 66–71.

[14] Haus, G., and Longari, M. A Multi-Layered, TimeBased Music Description Approach Based on XML. *Computer Music Journal 29*, 1 (Feb. 2005), 70–85.

[15] Hewlett, W. B. Beyond midi. MIT Press, Cambridge, MA, USA, 1997, ch. MuseData: Multipurpose Representation, pp. 402–447.

[16] Hild, H., Feulner, J., and Menzel, W. HARMONET: A neural net for harmonizing chorales in the style of j. s. bach. J. E. Moody, S. J. Hanson, and R. Lippmann, Eds., Morgan Kaufmann, pp. 267–274.

[17] Huron, D. Beyond midi. MIT Press, Cambridge, MA, USA, 1997, ch. Humdrum and Kern: Selective Feature Encoding, pp. 375–401.

[18] Jackendoff, R. *Consciousness and the computational mind*, vol. xvi of *Explorations in cognitive science, No. 3.* The MIT Press, Cambridge, MA, US, 1987.

[19] Kaliakatsos-Papakostas, M., and Cambouropoulos, E. Probabilistic harmonisation with fixed intermediate chord constraints. In *Proceeding of the joint 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC)* (2014), ICMC–SMC 2014, p. (to appear).

[20] Koops, H. V., Magalhães, J. P., and de Haas, W. B. A functional approach to automatic melody harmonisation. In *Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design* (New York, NY, USA, 2013), FARM '13, ACM, pp. 47–58.

[21] KRUMHANSL, C. L., AND KESSLER, E. J. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review 89*, 4 (1982), 334–368.

[22] LAITZ, S. G. *The complete musician: an integrated approach to tonal theory, analysis, and listening*. Oxford University Press, New York, 2012.

[23] OXENHAM, A. *The Perception of Musical Tones*. 2013.

[24] PACHET, F., AND ROY, P. Musical harmonization with constraints: A survey. *Constraints 6*, 1 (Jan. 2001), 7–19.

[25] PAIEMENT, J.-F., ECK, D., AND BENGIO, S. Probabilistic melodic harmonization. In *Proceedings of the 19th International Conference on Advances in Artificial Intelligence: Canadian Society for Computational Studies of Intelligence* (Berlin, Heidelberg, 2006), AI'06, Springer-Verlag, pp. 218–229.

[26] PARNCUTT, R. *Harmony: A Psychoacoustical Approach*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1989.

[27] PHON-AMNUAISUK, S., AND WIGGINS, G. A. The four-part harmonisation problem: A comparison between genetic algorithms and a rule–based system. In *In proceedings of the AISB'99 symposium on musical creativity* (1999), AISB, pp. 28–34.

[28] ROHRMEIER, M. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music 5*, 1 (Mar. 2011), 35–53.

[29] SAPP, C. S. Computational chord-root identification in symbolic musical data: Rationale, methods, and applications. *Computing in Musicology 15* (2008), 99–119.

[30] SCHORLEMMER, M., SMAILL, A., KÃ¼HNBERGER, K.-U., KUTZ, O., COLTON, S., CAMBOUROPOULOS, E., AND PEASE, A. Coinvent: Towards a computational concept invention theory. In *5th International Conference on Computational Creativity (ICCC) 2014* (June 2014).

[31] SELFRIDGE-FIELD, E., Ed. *Beyond MIDI: The Handbook of Musical Codes*. MIT Press, Cambridge, MA, USA, 1997.

[32] SIMON, I., MORRIS, D., AND BASU, S. Mysong: Automatic accompaniment generation for vocal melodies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), CHI '08, ACM, pp. 725–734.

[33] TERHARDT, E. Pitch, consonance, and harmony. *The Journal of the Acoustical Society of America 55*, 5 (Aug. 2005), 1061–1069.