Towards a Unified Model of Chords in Western Harmony

Johannes Hentschel, Fabian C. Moss, Andrew McLeod, Markus Neuwirth, Martin Rohrmeier

Digital and Cognitive Musicology Lab, École Polytechnique Fédérale de Lausanne

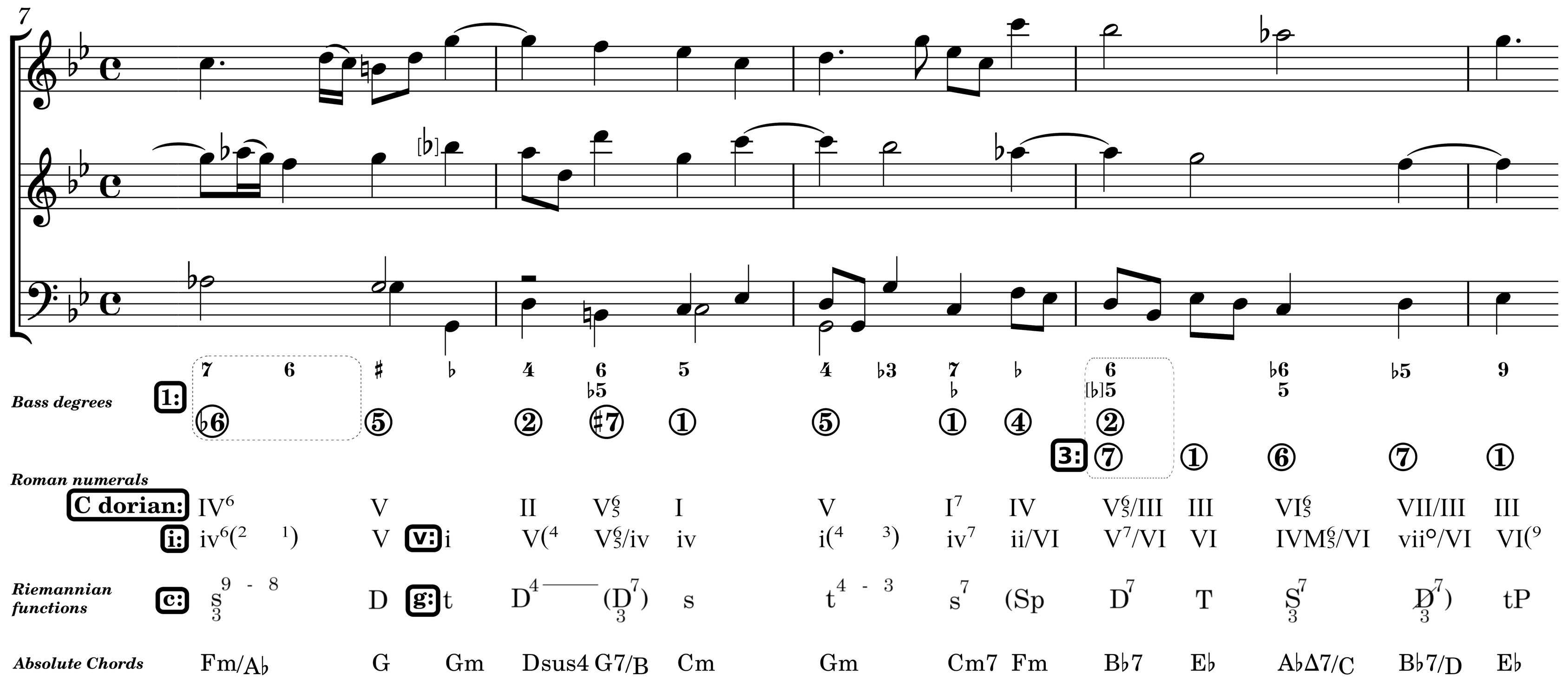


Fig. 1: A. Corelli, *Sonata a tre*, op. 1, no. 8, III. Largo, mm. 7–11, with several harmonic annotation systems underneath, specifically figured bass (taken from the 1681 princeps edition) with bass degrees [1], two types of Roman numerals, Riemannian function symbols [2], and absolute chords.

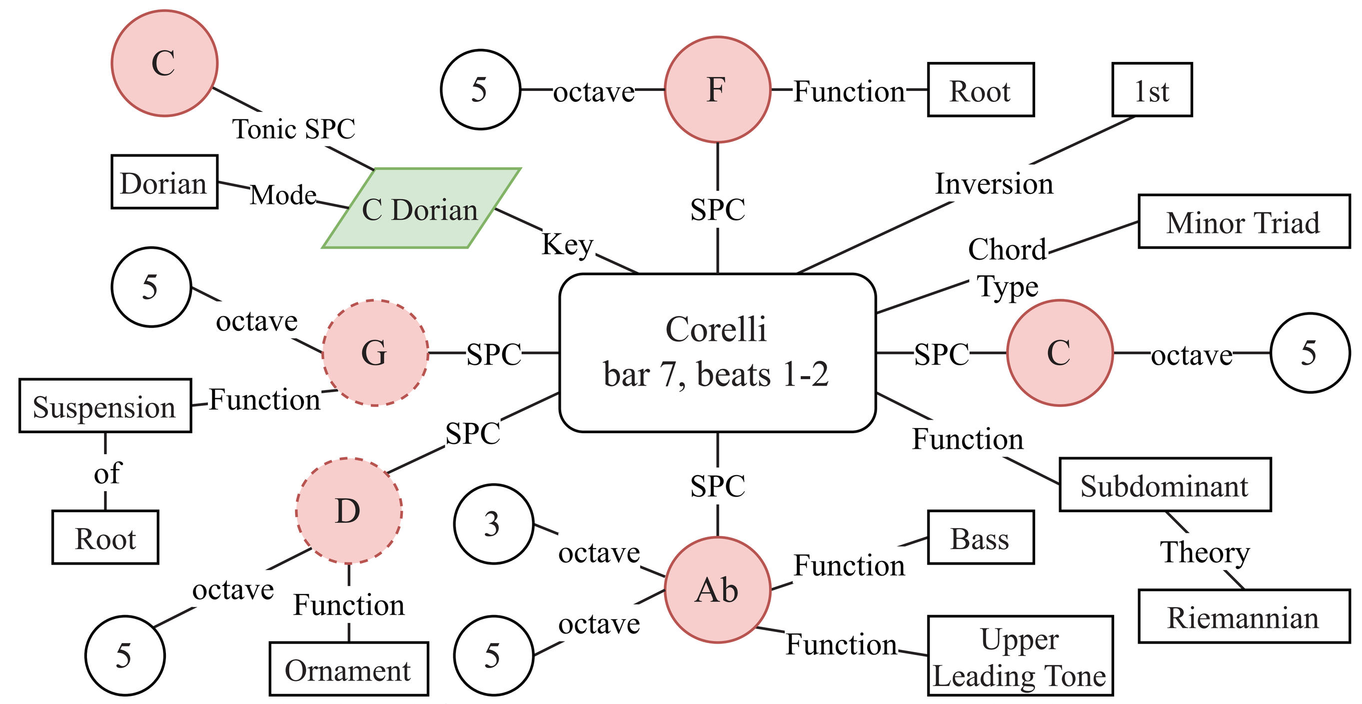
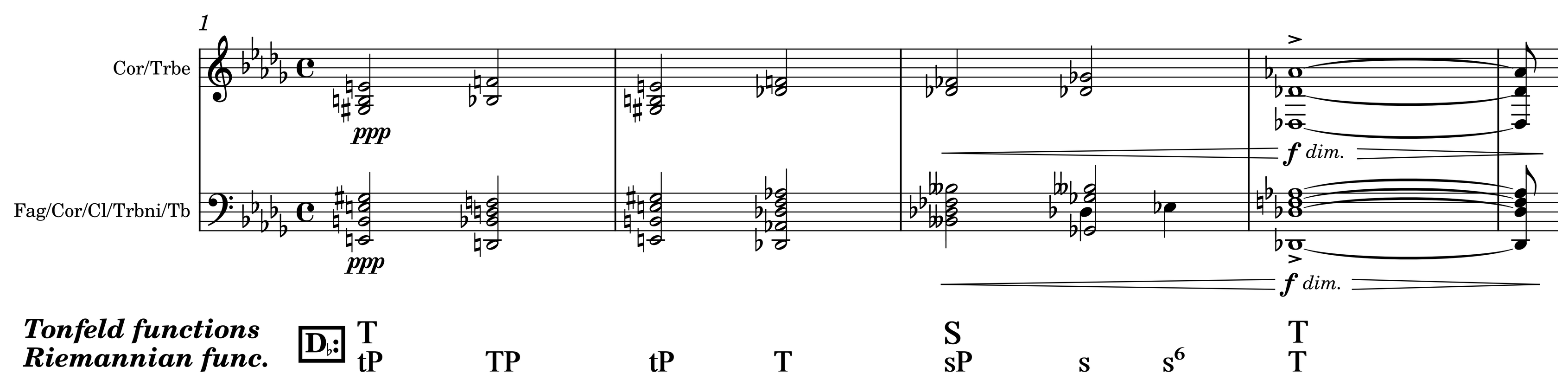
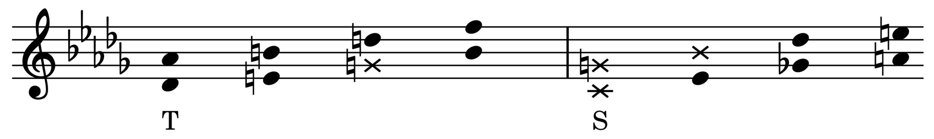


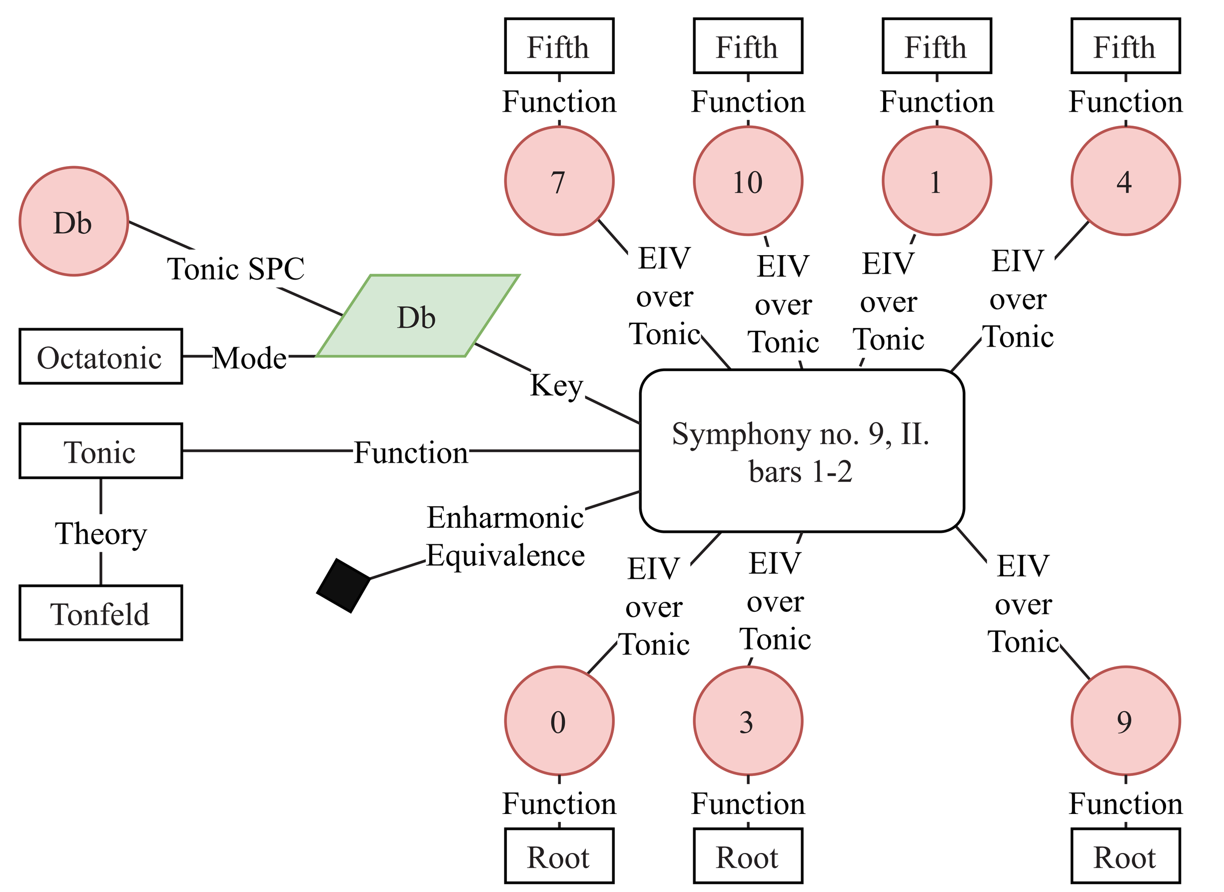
Fig. 2: The graph shows our chord model's least abstract representation of the first harmony in Fig. 1. It displays the pitches on the score surface as Specific Pitch Classes (SPC) with octave information and, where applicable, pitch functions attached. The presence of information on the key, chord type, Riemannian function, and inversion enables the model to convert between a multitude of annotation standards.



(a) A. Dvořák, *Symphony no. 9* (1893), II. Largo, mm. 1–5, score reduction, annotated with Riemannian and Tonfeld functions [3].



(b) Tonfeld functions tonic (T) and subdominant (S) for the Dvořák example above. Non-occurring notes crossed out.



(c) Graph representing the tonic Tonfeld function in mm. 1–2 above. Pitch classes are expressed as enharmonic intervals (EIVs) relative to D♭. Pitch functions correspond to the octatonic field based on D♭.

Fig. 3: Example for a Tonfeld function constituted under enharmonic equivalence. For instance, the G♯/A♭ in m. 2 is appropriately represented as D♭ (or enharmonic equivalent) + 7 semitones.

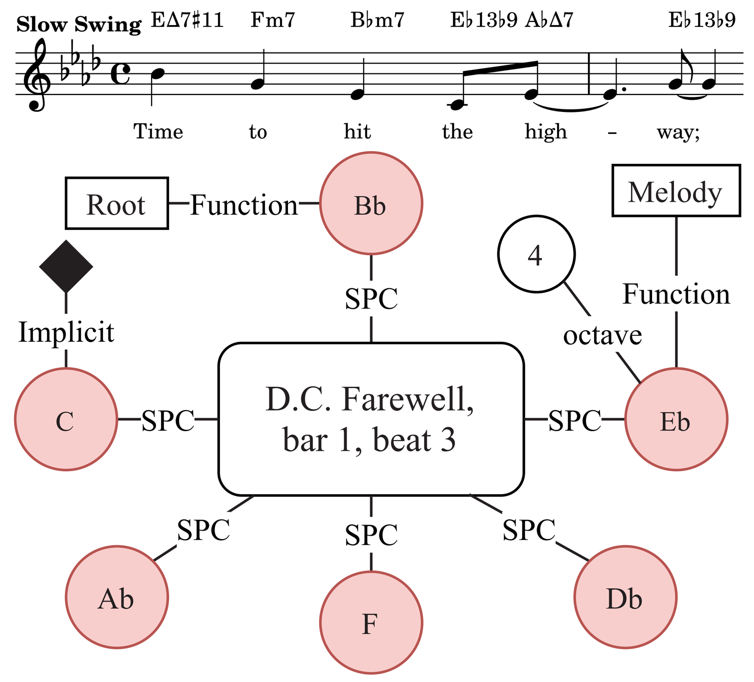


Fig. 4: R. Cole, *D. C. Farewell* (1976), mm. 1–2, an example where a chord comprises two PCs more than indicated by the absolute chord label B♭m7.

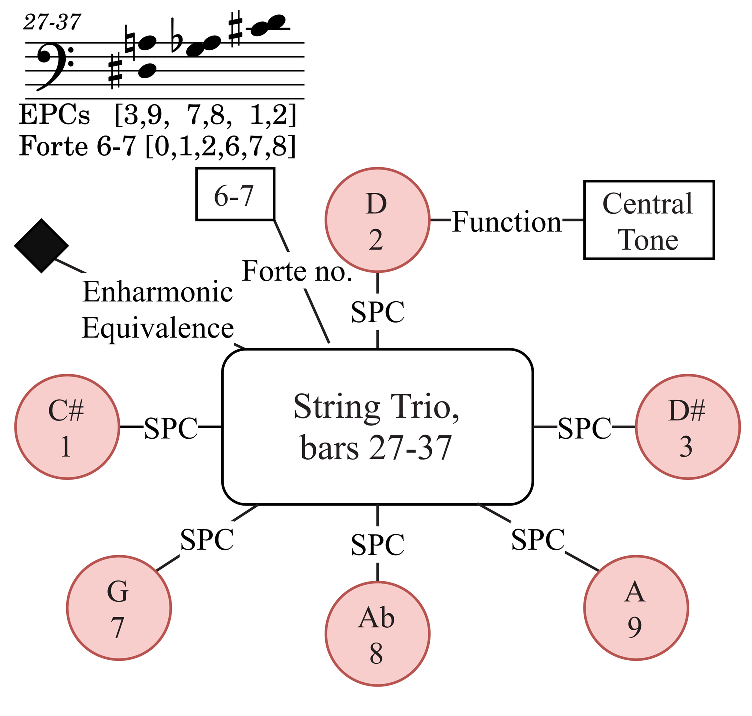


Fig. 5: S. Gubaidulina, *String Trio* (1988), hexachord sounding throughout mm. 27–37. Pitch classes are given as both specific and enharmonic. D4 is the central tone of the passage starting in m. 21 with an unisono.

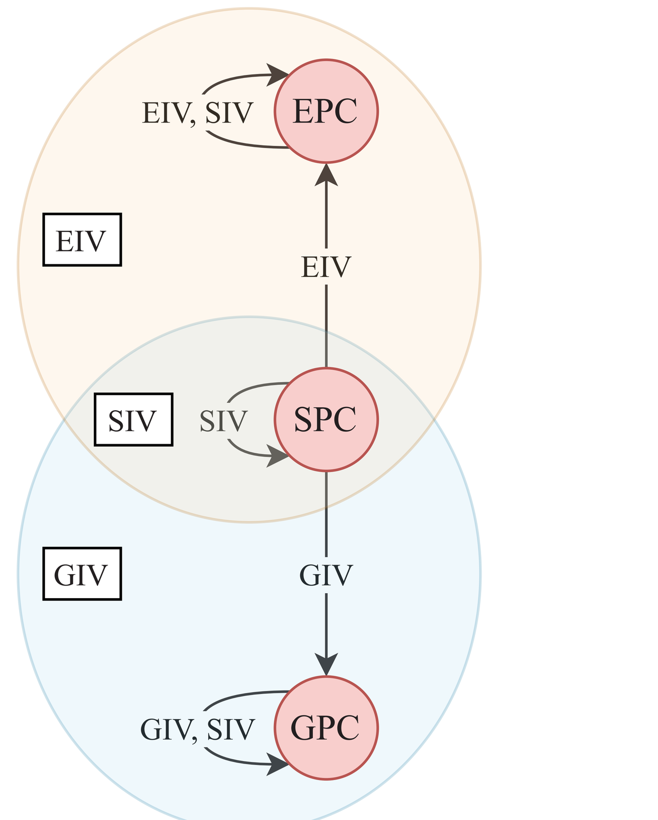


Fig. 6: The different pitch class (PC) types in our model, together with the result of transposing them by some interval (IV) type (the arrows), and the interval type that measures the distance between two PC types (the labeled ovals). Both PCs and IVs can be generic (G), specific (S), or enharmonic (E).

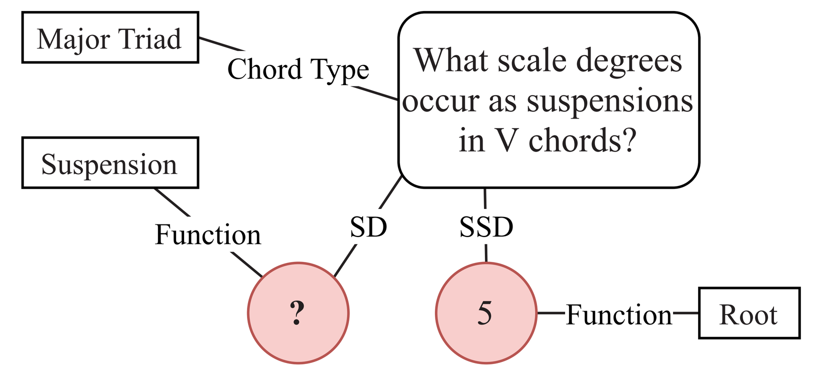


Fig. 7: Representation of a graph query where the question mark symbolizes the queried variable. In order for a pitch class to be expressed as scale degree (SD), at least its GPC and a key need to be defined for the respective chord; a specific scale degree (SSD) requires an SPC (compare Fig. 6 and the model specification below).

Abstract

We propose a model which allows for the representation of chords at multiple levels of abstraction: from chord realizations on the score level (if available), to pitch-class collections (including a potential application of different equivalences, such as enharmonic or octave equivalence), to pitch- and chord-level functions and higher-order abstractions. Importantly, our proposed model is also well-defined for theories which do not specify information at each level of abstraction (e.g., some theories make no claims about harmonic function), representing only those harmonic properties that are included and inducing others where possible (e.g., deriving scale degrees from root and key information). Our model thus represents an important step towards a unified representation of harmony and its various applications.

Problem Setting

Chord-based harmony is an important aspect of many types of Western music, across genres, regions, and historical eras. However, the consistent representation and comparison of harmony across a wide range of styles (e.g. classical music, Jazz, Rock, or Pop) is a challenging task. Moreover, even within a single musical style, multiple theories of harmony may exist (see, for example, the annotations in **Fig. 1**), each relying on its own (possibly implicit) assumptions and leading to harmonic analyses with a distinct focus (e.g. on the root of a chord vs. its bass note) or representation (e.g. spelled vs. enharmonic pitch classes). Cross-stylistic comparisons (as well as comparisons within a single style involving multiple annotation systems) are therefore even more difficult, particularly in a large-scale computational setting that requires a common overarching representation. The figures above exemplify this challenge through four stylistically distinct musical contexts with various harmonic annotations.

Elements of the Unified Chord Model

Our proposed model represents chords as graphs, where the modelled chord and its position in a piece form a central node, and its properties are labeled edges with attached nodes representing values. Boolean `true` values (flags) are represented as black diamonds in the figures above.

Pitch Classes (PCs, red circles). Fundamentally, the model is based on viewing a chord as a selection of pitch classes (PCs): Generic Pitch Classes (GPCs; A-G), Specific Pitch Classes (SPCs; GPC plus accidentals), and Enharmonic Pitch Classes (EPCs; MIDI note number mod 12). An SPC can be converted into an EPC or a GPC, but not vice versa (see **Fig. 6**). Each of a chord's PCs can also be represented as the interval above some reference PC (e.g., the chord's root or bass; see for example **Fig. 3c**) and, analogously to PCs, an interval may be generic (GIV; the difference between two GPCs, e.g. any 3rd), specific (SIV; the difference between two SPCs, e.g. a major 3rd), or enharmonic (EIV; the difference between two EPCs, e.g. 5 semitones). The convertibility between various PC representations ensures that the same chord information can be expressed in many annotation standards.

Score level (PCs + octave information). This least abstract representation of a chord consists of the set of pitches that are taken from all the notes within a given score segment (e.g., **Fig. 2**). On this level, pitches are typically represented as SPCs, although we can also model annotated MIDI files at this level with pitches viewable only as EPCs. Information beyond the pitches contained in a score segment stems from one or several chord labels and are shown as properties of individual pitch classes or of the entire chord.

Pitch equivalences. The model allows for abstraction from the score level by applying different equivalence operations, such as octave and enharmonic equivalence. To apply octave equivalence, the octave information of a pitch is simply ignored. In cases where octave information is not available because a symbolic pitch representation is missing or does not include all pitches expressed by a chord label, octave equivalence is therefore necessarily assumed (e.g. in **Fig. 4** for all PCs except E♭). Enharmonic equivalence is represented as a flag that may be associated either with individual PCs or the entire chord, converting the corresponding PCs to EPCs (e.g. in **Fig. 3c & 5**).

Pitch functions. PCs can be assigned functions within the chord. Importantly, each PC can be classified as either a chord tone or a non-chord tone (represented as dotted circles in **Fig. 2**). The possibility of ignoring non-chord tones, such as suspensions or ornaments, is common to many annotation standards. Other common pitch functions are, for example, root, bass note, and leading tone, but this set of categorical pitch functions can easily be extended.

Chord functions and properties. Since many music theories analyze chords within their tonal contexts, information about an underlying pitch collection, such as a key or a tonal hierarchy, features prominently among the chord-level properties that may be added (green parallelograms in **Fig. 2 & 3c**). This information is of particular importance for representing or converting scale degrees, which are used by certain theories (e.g. Roman numerals and bass degrees in **Fig. 1**) and in the graph query example of **Fig. 7**. Other typical chord properties include chord type, inversion, and chord function (e.g., tonic, dominant, predominant); functions pertaining to different theories may be differentiated (see, for example, **Fig. 3c**).

Model Specification

```
CHORD       := <POS, HARMONY>
POS         := <[bar: (int | int-int)], [beat: (int | int-int)], [time: float]>
HARMONY     := <NOTE*, [KEY], [(CHORDFUNC [of: THEORY])*], [CHORDTYPE], [INV], [ENH], [FORTE], ...>
NOTE        := <PITCH, [(NOTEFUNC [of: THEORY])*], [ENH], [IMP], ...>
PITCH       := (PC | INTERVAL | SD) [OCTAVE]
PC          := GPC | SPC | EPC
GPC         := A | B | C | D | E | F | G
SPC         := GPC [ACCIDENTAL]
ACCIDENTAL  := b* | #*
INTERVAL    := GIV | SIV | EIV
EPC, EIV    := {0..11}
SIV         := (Perfect | Maj | Min | Aug* | Dim*) GIV
SD          := GSD | SSD
GIV, GSD    := {1..7}
SSD         := GSD [ACCIDENTAL]
OCTAVE      := int
NOTEFUNC    := Suspension [of: (NOTEFUNC | PITCH)] | Root | Bass | Melody | UpperLeadingTone | CentralTone | Ornament | ...
THEORY      := Riemannian | Tonfeld | FigBass | RomanNumeralAnalysis | Forte | ...
ENH         := Enharmonic: (True | False)
IMP         := Implicit: (True | False)
FORTE       := Forte No: String
KEY         := <tonic: PITCH, mode: MODE, [type: KEYTYPE], [KEY]>
MODE        := Maj | Min | Dor | ... | INTERVAL*
KEYTYPE     := Global | Local | Secondary
CHORDFUNC   := Tonic| Subdominant | Dominant | ...
CHORDTYPE   := Maj | Min | Dim | Aug | MajMaj7 | MajMin7 | MinMaj7 | MinMin7 | Dim7 | HalfDim7 | AugMaj7 | AugMin7 | ... | INTERVAL*
INV         := {0..N}
```

References

[1] L. Holtmeier. "Funktionale Mehrdeutigkeit, Tonalität und arabische Stufen. Überlegungen zu einer Reform der harmonischen Analyse". In: *Zeitschrift der Gesellschaft für Musiktheorie [Journal of the German-Speaking Society of Music Theory]* 8.3 (2011), pp. 465–487. DOI: 10.31751/655.

[2] R. Cohn. "Introduction to Neo-Riemannian Theory: A Survey and a Historical Perspective". In: *Journal of Music Theory* 42.2 (1998), pp. 167–180. DOI: 10.2307/843871.

[3] M. Polth. "The Individual Tone and Musical Context in Albert Simon's Tonfeldtheorie". In: *Music Theory Online* 24.4 (2018). DOI: 10.30535/mto.24.4.15.

Acknowledgements

This publication is possible thanks to funding by the Swiss National Science Foundation (Grant no. 182811) and by Mr. Claude Latour (Latour Chair in Digital Musicology at EPFL).

Swiss National Science Foundation