COMPUTERGESTÜTZTE MUSIKFORSCHUNG 1

Institut für Musikinformatik und Musikwissenschaft
Wintersemester 2025–26

Christophe Weis
christophe.weis@stud.hfm.eu

Woche 01
14.10.2025

Organisation

wöchentlich, Di. 14.30–16.00, K10 Raum 309

Modul **Music Processing**

- **BA MI (HF)/MW (EF), wiss. Schwerpunkt:** Pflicht (4. Semester)
- **BA MI (HF)/MW (EF), künstl. Schwerpunkt:** Wahlpflicht (6. Semester)
- **BA MW (HF)/MI (EF):** Pflicht (4. Semester) – reduzierter Arbeitsaufwand
- **BA MI/MW (KF):** Pflicht (4. Semester)
- **BA:** Wahlfach

Projektarbeit

- eine selbstständige praktische Arbeit aus den Bereichen Musikkodierung, symbolbasierte Musikverarbeitung und –analyse mit Dokumentation (ca. 5000 Zeichen)

Übungen

- Tutorin: Joanna Friedrich-Sroka
- wöchentlich, ..., K10 Raum 309

01.

RESSOURCEN

Einige Ressourcen

Digital Bibliography & Library Project (DBLP)

- https://dblp.org
- Bibliografien-Datenbank mit Artikeln/Papers/Veröffentlichungen im Bereich Informatik
- z. B. mit den Suchbegriffen „symbolic streamid:conf/ismir:“

arXiv

- https://arxiv.org
- Preprints aus verschiedenen wissenschaftlichen Bereichen

JSTOR

- https://www.jstor.org
- Digitale Bibliothek mit wissenschaftlichen Fachzeitschriften und Büchern

Academia.edu

- https://www.academia.edu
- Plattform für wissenschaftliche Papers

ResearchGate

- https://www.researchgate.net
- Datenbank für wissenschaftliche Publikationen

Badische Landesbibliothek

- https://www.blb-karlsruhe.de
- Zugang zu Fachzeitschriften und E-Book-Sammlungen

Einige Ressourcen

Konferenzen

- **International Society for Music Information Retrieval (ISMIR)**
  - https://www.ismir.net
  - Music Information Retrieval
- **Music Encoding Conference (ICMC)**
  - https://music-encoding.org/conference
  - Digitale Darstellung von Musik
- **International Conference on Technologies for Music Notation and Representation (TENOR)**
  - https://www.tenor-conference.org
  - Darstellung/Visualisierung/Notation von Musik
- **Mathematics and Computation in Music (MCM)**
  - http://www.smcm-net.info
  - Schnittstellen zwischen Musik, Mathematik und Informatik
- **Sound and Music Computing Conference (SMC)**
  - https://smcnetwork.org
  - Erforschung/Modellierung/Generierung von Musik/Klängen

Einige Ressourcen

- Meinard Müller. ***Fundamentals of Music Processing: Using Python and Jupyter Notebooks*** (Vol. 2). Cham: Springer, 2021.
- David Meredith (Hg.). ***Computational Music Analysis.*** Heidelberg: Springer, 2016.

02.

DARSTELLUNGEN VON MUSIK

Darstellungen von Musik

Darstellungen im **„Bildformat“:**

- Notenschrift, Partituren, Tabulaturen, ...
- (Funktionale oder freie) grafische Notation, Musik-Nachzeichnungen

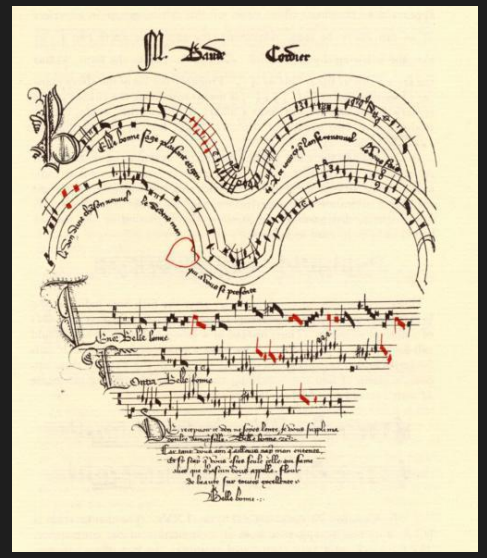
Darstellungen von Musik

Darstellungen im **„Bildformat“:**

- Notenschrift, Partituren, Tabulaturen, ...

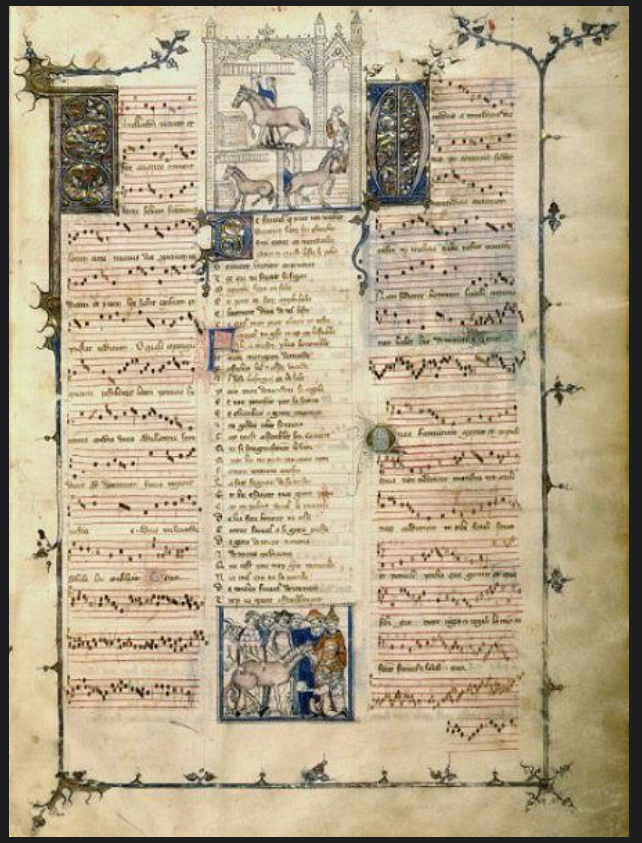
Jeongganbo-Notation
(14.–15. Jahrhundert)

https://en.wikipedia.org/wiki/File:%EC%8B%9C%EC%9A%A9%ED%96%A5%EC%95%85%EB%B3%B4_(%E6%99%82%E7%94%A8%E9%84%95%E6%A8%82%E8%AD%9C).jpg



Baude Cordier. *Belle, bonne, sage.*
(ca. 1400)

https://upload.wikimedia.org/wikipedia/commons/0/09/CordierColor.jpg



Roman de Fauvel: Neumenschrift
(ca. 1310–1314)

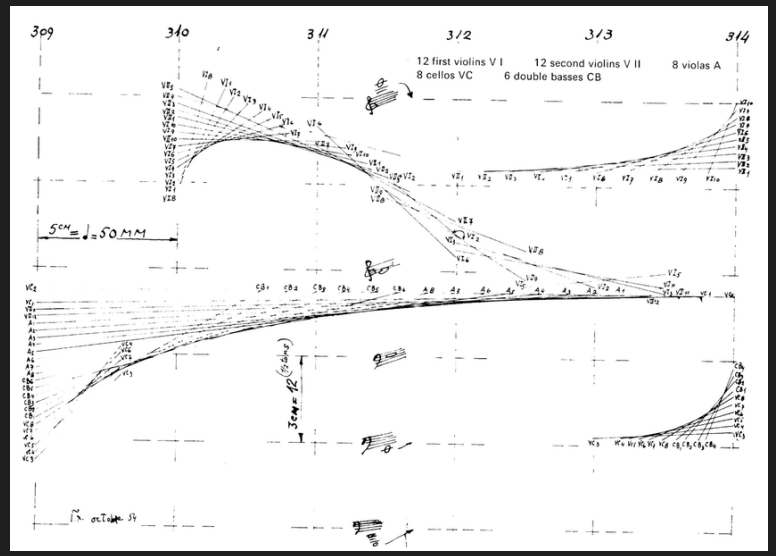
https://upload.wikimedia.org/wikipedia/commons/b/bf/Roman_de_Fauvel.jpg

- (Funktionale oder freie) grafische Notation, Musik-Nachzeichnungen

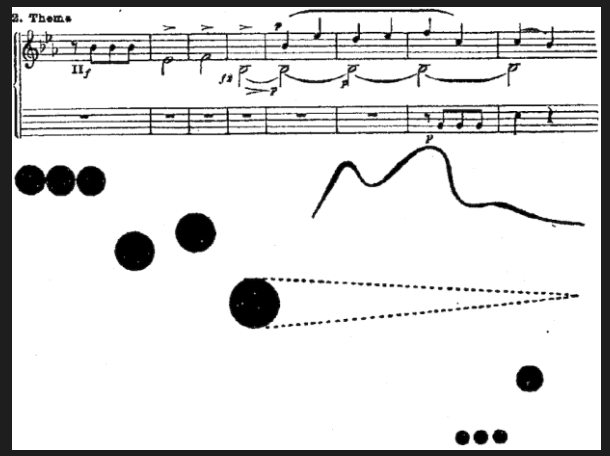
Darstellungen von Musik

Darstellungen im **„Bildformat“:**

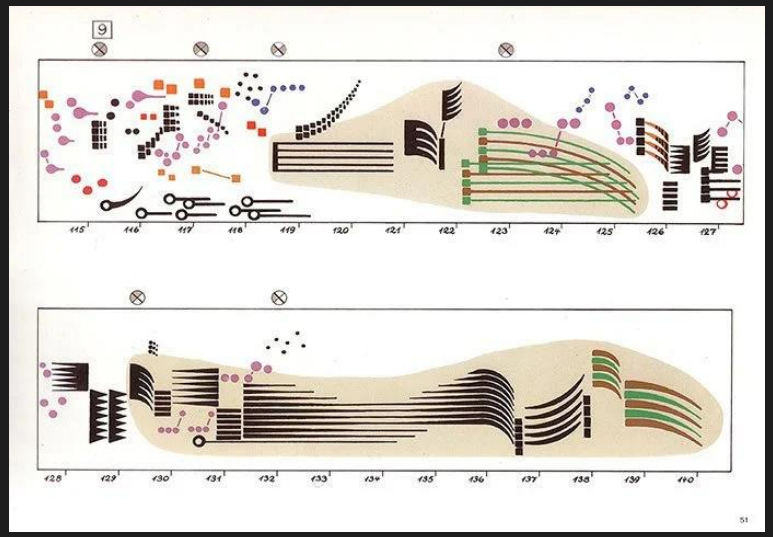
- Notenschrift, Partituren, Tabulaturen, ...
- (Funktionale oder freie) grafische Notation, Musik-Nachzeichnungen



Iannis Xenakis. Skizze zu *Metastaseis* (1954)



Wassily Kandinsky. *Punkt und Linie zu Fläche* (1926)



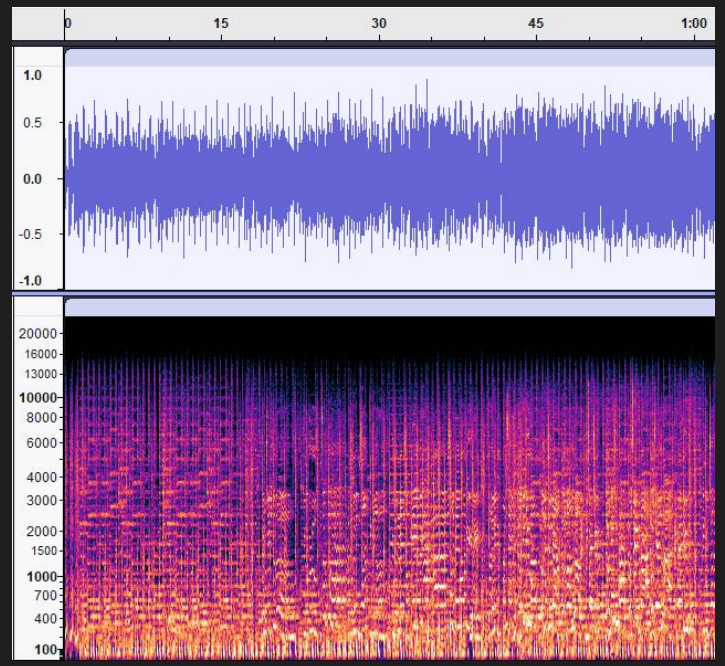
György Ligeti. *Artikulation* (1958)
Hör-Partitur von Rainer Wehinger (1970)
https://youtu.be/71hNl_skTZQ

Darstellungen von Musik

Audio-basierte Darstellungen

- (Digitale oder analoge) Darstellungen von akustischen Wellen
- „Invertierbare“ Kodierung von zeitlichen und klanglichen Informationen für eine akustische Umsetzung
- Keine explizite Darstellung von Notenparametern und Aufführungsinformationen
- Wenig unmittelbare Informationen über strukturelle, formale Beziehungen zwischen musikalischen Elementen

→ *Computergestützte Musikforschung 2* (im Sommersemester 2026)



Wellenform und **Spektrogramm** eines Audio-Signals, dargestellt mithilfe des Programms *Audacity*

Darstellungen von Musik

Symbolische Darstellungen

- Explizite, diskrete Darstellung von musikalischen Parametern (Tonhöhen, Rhythmus, Dynamik, ...)
- „Higher-Level“-Kodierung, Abstraktion von zeitlichen und klanglichen Informationen
- Formate, die maschinenlesbar – und vom Menschen interpretierbar sind
- Einsatz in der computergestützten Musikanalyse, der Musikgenerierung, der computergestützten Komposition, der Modellierung von Strukturen, der Genre-Klassifikation, ...

→ *Computergestützte Musikforschung 1*

Darstellungen von Musik

Symbolische Darstellungen

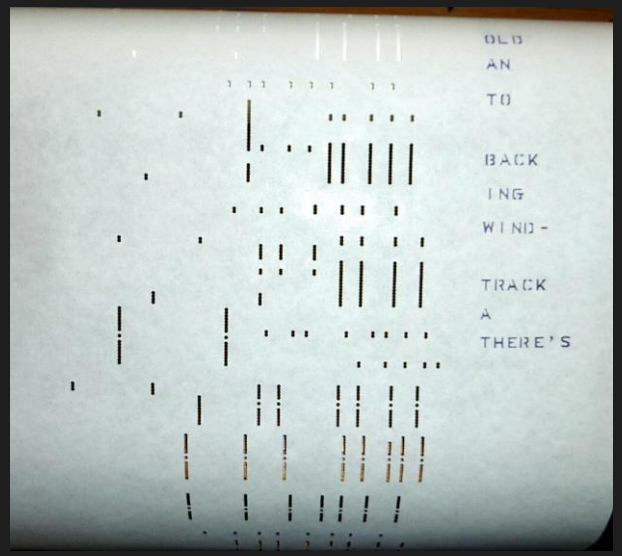
- Gängige Formate:
  - Piano-Roll
  - MIDI
  - MusicXML
  - MEI (Music Encoding Initiative)
  - **kern (Humdrum)
  - LilyPond
  - NoteTuple
  - ...
- Im erweiterten Sinn:
  - Alle mit „symbolischen“ Labels versehenen musikalischen Sequenzen
  - Geometrische, Graphen-basierte, algebraische, Darstellungen von musikalischen Parametern
  - ...

03.

SYMBOLISCHE MUSIK-DARSTELLUNGEN

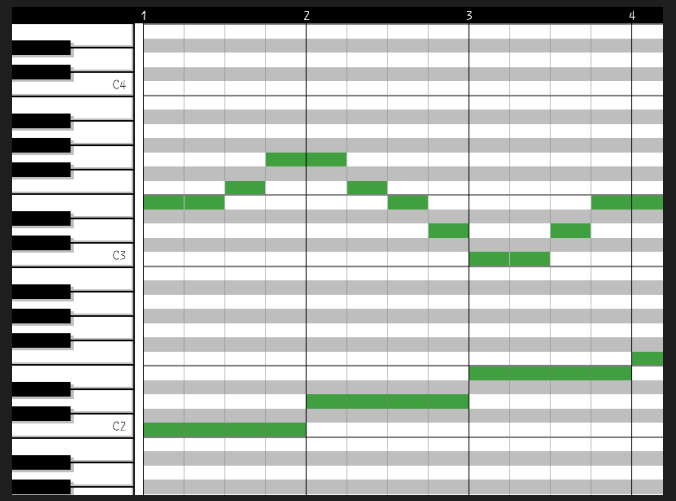
Piano-Roll

- Einsatz zur Steuerung von selbstspielenden Klavieren (beliebt Ende 19. / Anfang 20. Jahrhundert)
- Kodierung von Notenkontrolldaten als **Lochschrift**
- Möglichkeit zur Aufnahme und Wiedergabe von Aufführungen (z. B. Gustav Mahler, George Gershwin, ...)



https://upload.wikimedia.org/wikipedia/commons/thumb/b/b8/Mastertouch_Piano_Roll_Australian_Dance_Gems.jpg/1024px-Mastertouch_Piano_Roll_Australian_Dance_Gems.jpg

Gustav Mahler – Piano-Roll-Aufnahmen
https://youtu.be/PScJkkQPwwE



https://upload.wikimedia.org/wikipedia/commons/b/bb/Computer_music_piano_roll.png

Piano-Roll

- Einsatz zur Steuerung von selbstspielenden Klavieren (beliebt Ende 19. / Anfang 20. Jahrhundert)
- Kodierung von Notenkontrolldaten als **Lochschrift**
- Möglichkeit zur Aufnahme und Wiedergabe von Aufführungen (z.B. Gustav Mahler, George Gershwin, ...)
- Eine mögliche „musikinformatische“ Definition:
  - **geometrische, zweidimensionale** Visualisierung von Notenkontroll-Informationen
  - Kodierung der **Zeit** entlang der horizontalen Achse, Kodierung der **Tonhöhe** entlang der vertikalen Achse
  - Darstellung von Noten als Rechtecke, Kodierung von 3 Notenparametern:
    - **Onset:** x-Koordinate der linken Seite eines Rechtecks
    - **Notendauer:** Breite eines Rechtecks
    - **Tonhöhe:** y-Koordinate der unteren Seite eines Rechtecks

MIDI

- 1981/82 eingeführt, um gleichzeitiges Spielen/Arbeiten mit elektronischen Instrumenten verschiedener Hersteller zu erlauben.

- Die ***Standard-MIDI-File***-Spezifikation bestimmt das Format von MIDI-Dateien als Liste von MIDI-Befehlen. Wichtigste MIDI-Befehle: ***note-on, note-off***

- *note-on*-Events und *note-off*-Events werden gesendet mit:

  - einer MIDI-Zahl:
    → Ganzzahl zwischen 0 und 127, codiert die Tonhöhe

  - der Velocity:
    → Ganzzahl zwischen 0 und 127, codiert die Anschlagsdynamik (bei *note-on*-Events), das Abklingen/Decay („Release-Velocity“ bei *note-off*-Events)

  - einer MIDI-Kanal-Spezifikation:
    → Ganzzahl zwischen 0 und 15, bestimmt das einem spezifischen Kanal zugewiesene Instrument

  - einem Zeitstempel:
    → Ganzzahl, bestimmt Anzahl der zu wartenden Ticks vor der Ausführung des Events

MIDI

- **Ticks** oder **Clock-Pulses** definieren die zeitliche Auflösung von MIDI-Dateien.
- Die Anzahl der Ticks pro Viertelnote (**PPQN**, *pulses per quarter note*) wird im Header-Chunk (am Dateianfang) der MIDI-Datei definiert.
- Der **Header-Chunk** definiert u. a. Dateiformat, Anzahl der Spuren und PPQN nach dem Schema:

  `header_chunk = „MThd“ + <header_length> + <format> + <number_of_tracks> + <PPQN>`

- Der **Track-Chunk** definiert die Abfolge der MIDI-Events nach dem Schema:

  `track_chunk = „MTrk“ + <length> + <track_event> (+ <track_event> + ...)`

MIDI

- MIDI-Varianten:
  - Standard MIDI
    → siehe vorherige Slides
  - MIDI 2.0
    → veröffentlicht 2020
    → Erweiterung des Standard-Protokolls
  - REMI (**RE**vamped **MI**DI-derived events)
    → MIDI-Tokenizer: Umwandlung von MIDI-Scores in Abfolgen von diskreten Tokens
  - MIDI-like
    → MIDI-Tokenizer

MusicXML

- Als universelles Format entwickelt, um Austausch zwischen verschiedenen Notensatzprogrammen zu ermöglichen.
- Mittlerweile allgemein von Notensatzprogrammen und einigen Sequenzern unterstützt.
- Basiert auf *XML* (Extensible Markup Language)
  → Kodierung, die maschinenlesbar und menschenlesbar ist
- Kodierung, die Notations- und Layout-orientiert ist
- Aufbau einer MusicXML-Datei:
  - **Elemente** definieren die wichtigsten Einheiten einer *MusicXML*-Struktur und enthalten entweder Text oder weitere Elemente.
  - **Tags** dienen zur Auszeichnung von Elementen:
    - Starttags kennzeichnen Anfang eines Elements nach dem Modell: `<Elementname>`
    - Endtags kennzeichnen Ende eines Elements nach dem Modell: `</Elementname>`
    - leere Tags kennzeichnen Elemente ohne Inhalt nach dem Modell: `<Elementname/>`
  - **Attribute** definieren zusätzliche Eigenschaften einzelner Elemente nach dem Modell:
    `<Elementname Attributname="Wert">`

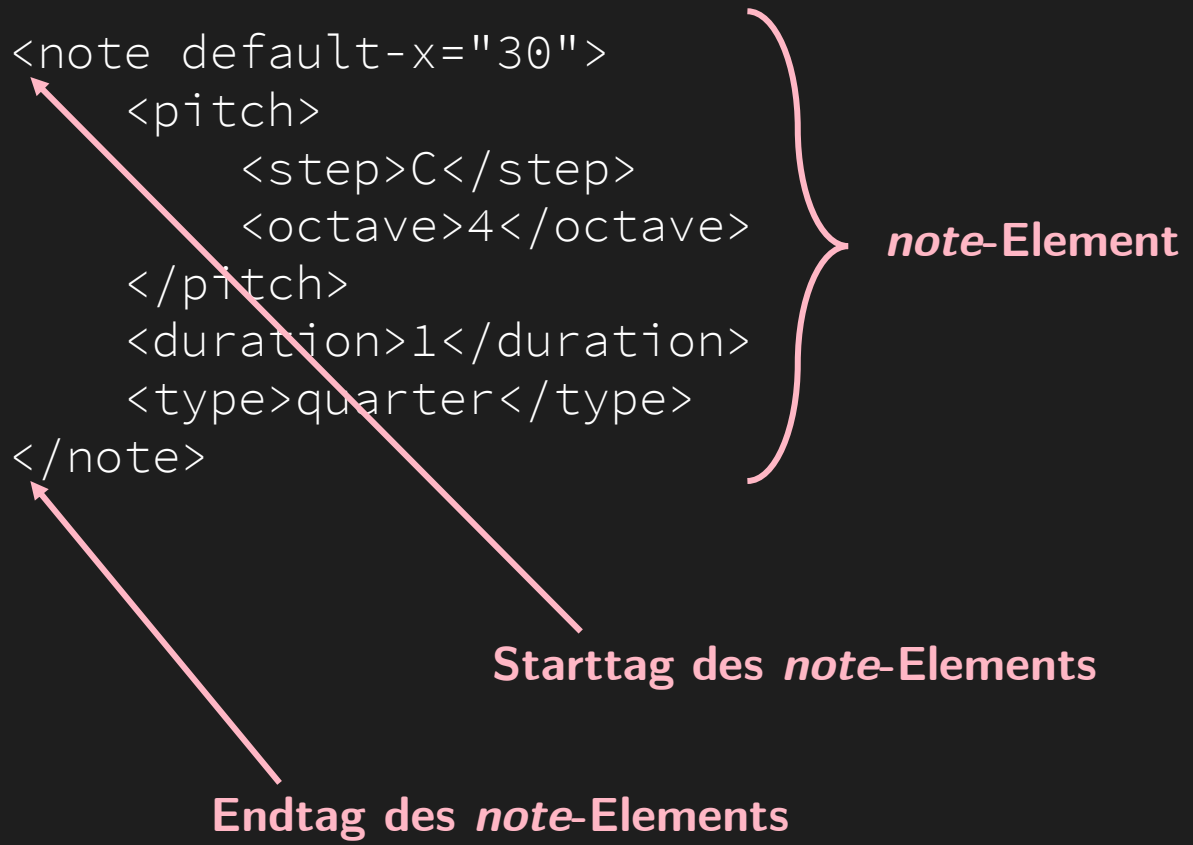
MusicXML

- *Beispiel:* MusicXML-Codierung eingestrichenes C als Viertelnote

```
<note default-x="30">
    <pitch>
        <step>C</step>
        <octave>4</octave>
    </pitch>
    <duration>1</duration>
    <type>quarter</type>
</note>
```

MusicXML

- *Beispiel:* MusicXML-Codierung eingestrichenes C als Viertelnote



MusicXML

- *Beispiel:* MusicXML-Codierung eingestrichenes C als Viertelnote

```
<note default-x="30">
    <pitch>
        <step>C</step>
        <octave>4</octave>
    </pitch>
    <duration>1</duration>
    <type>quarter</type>
</note>
```

***pitch*-Element**

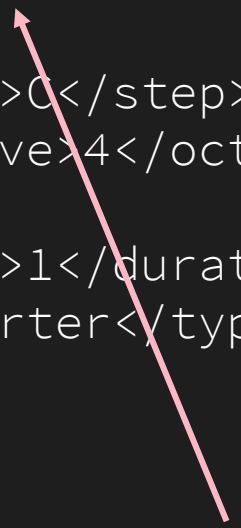
Starttag des *pitch*-Elements

Endtag des *pitch*-Elements

MusicXML

- *Beispiel:* MusicXML-Codierung eingestrichenes C als Viertelnote

```
<note default-x="30">
    <pitch>
        <step>C</step>
        <octave>4</octave>
    </pitch>
    <duration>1</duration>
    <type>quarter</type>
</note>
```



ein mögliches Attribut des *note*-Elements (hier: *default-x* zur Bestimmung der horizontalen Position)

MusicXML

- Grundlegende Eigenschaften von MusicXML-Dokument: **Wohlgeformtheit** und **Validität**

- Ein MusicXML-Dokument ist **wohlgeformt**, wenn:

  - es genau ein Wurzelelement (äußeres Element) besitzt,
  - alle Elemente (bis auf leere Elemente) sind durch einen Start- und einen Endtag ausgezeichnet,
  - alle Elemente auf derselben Ebene geöffnet und geschlossen werden (Einhalten einer Baumstruktur),
  - kein Element mehrere Attribute mit demselben Namen besitzt,
  - Werte von Attributen in Anführungszeichen stehen,
  - Start- und Endtags jeweils Groß- und Kleinschreibung beachten.

- Vordefinierte **Grammatiken/Schemata** erlauben den Austausch von Daten mittels MusicXML-Dateien.

- Ein MusicXML-Dokument ist **valide**, wenn:

  - es wohlgeformt ist,
  - es zusätzlich einen Verweis auf ein solches Schema enthält und dieses einhält.

MusicXML

- Beispiele von MusicXML-Dateien:
  https://www.musicxml.com/music-in-musicxml/example-set/

- Weitere ausführliche Referenzen:
  https://www.w3.org/2021/06/musicxml40/musicxml-reference/

MEI

- Open-Source-Projekt, um Codierung, Archivierung und Austausch von Musik zwischen verschieden wissenschaftlichen Communities zu erlauben (betreut durch die Akademie der Wissenschaften und der Literatur)
- Basiert auf *XML*
- kodiert musikalische Inhalte auf inhaltlicher Ebene, statt als rein grafische Anweisung (z. B. zusätzliche editorische Hinweise, Quellenangaben, Varianten in verschiedenen Handschriften, ...)
- Anwendungsgebiete in wissenschaftlichen und editorischen Bereichen (kritische Editionen, Archivierung, musikwissenschaftliche Forschung, ...)
- Das von Theoretiker*innen und Historiker*innen entwickeltes MEI-Schema definiert Kodierungsregeln für alle wesentlichen Notationsformen der westlichen Musiknotation: https://music-encoding.org/resources/schemas.html
- Eine Reihe von Open-Source-Tools erlauben das Arbeiten mit MEI: https://music-encoding.org/resources/tools.html

MEI

- mei-friend:
  - https://mei-friend.mdw.ac.at
  - browserbasierter MEI-Editor

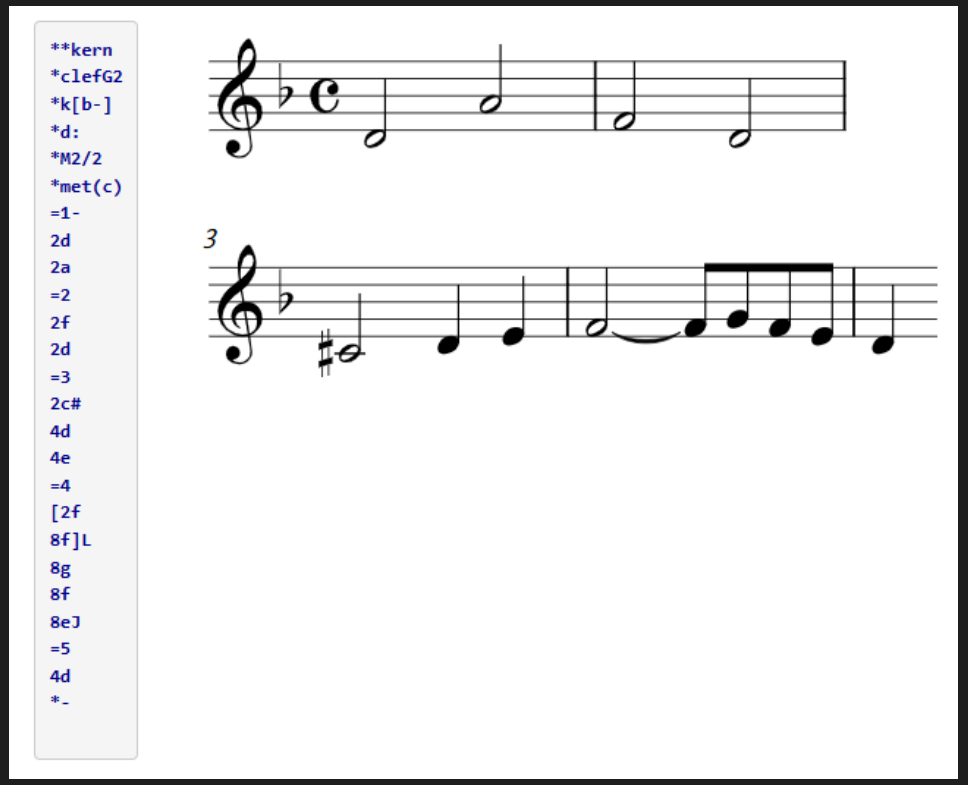
- MerMEId:
  - https://github.com/edirom/mermeid
  - entwickelt zum Erstellen von Katalogen und Hinzufügen von Metadaten zu Datensätzen

- MEIse:
  - https://de.dariah.eu/mei-score-editor
  - Editor zum Anzeigen und Bearbeiten von MEI-Dokumenten

- Verovio:
  - https://www.verovio.org/index.xhtml
  - Library zum Erstellen von MEI-basierten Partituren

**kern (Humdrum)

- Format zur Darstellung wesentlicher musikalischer Parameter innerhalb des zur computergestützten Analyse entwickelten *Humdrum*-Toolkits.
- Dokumentation: https://www.humdrum.org/rep/kern/
- Beispiele: https://www.humdrum.org/guide/ch02/



https://www.humdrum.org/guide/ch02

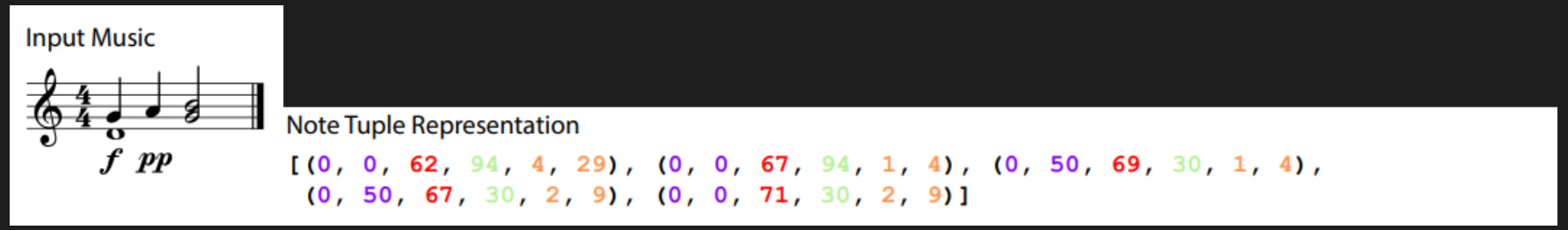
NoteTuple

- Zugehöriges Paper: C. Hawthorne et al. (2018). *Transformer-NADE for Piano Performances.*

- Darstellung von Noten als **Tupel** mit 4 Attributen:
  - **Time Offset** (Dargestellt mit 2 Werten für größere und kleinere Ticks/Zeiteinheiten)
  - **Pitch**
  - **Velocity**
  - **Duration** (Dargestellt mit 2 Werten für größere und kleinere Ticks/Zeiteinheiten)

- Bei Akkorden: Auflistung der Noten-Tupel in der Reihenfolge der aufsteigenden Tonhöhe.

- Vorteil (insbesondere im Machine-Learning) gegenüber anderen Formaten: alle zu einer Note gehörenden Attribute sind an derselben Stelle kodiert.

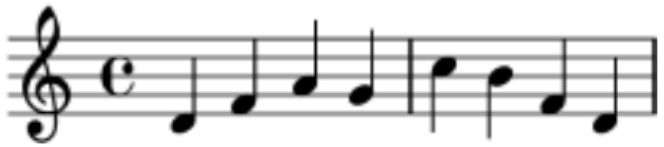


C. Hawthorne et al. (2018). *Transformer-NADE for Piano Performances.*

LilyPond

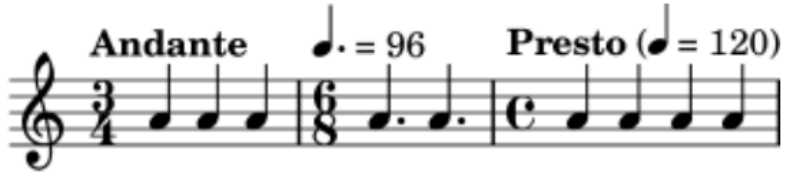
- Textbasiertes Notensatzprogramm
- Dokumentation: https://lilypond.org/manuals.de.html

```
\relative {
  d' f a g
  c b f d
}
```



https://lilypond.org/doc/v2.24/Documentation/learning/simple-notation

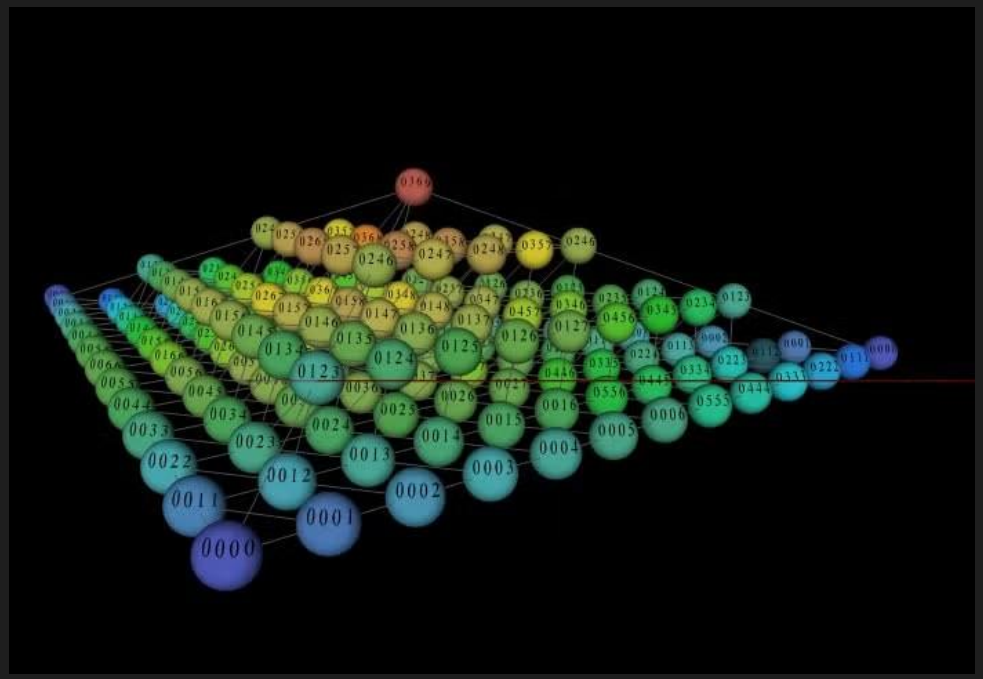
```
\relative {
  \time 3/4
  \tempo "Andante"
  a'4 a a
  \time 6/8
  \tempo 4. = 96
  a4. a
  \time 4/4
  \tempo  "Presto" 4 = 120
  a4 a a a
}
```

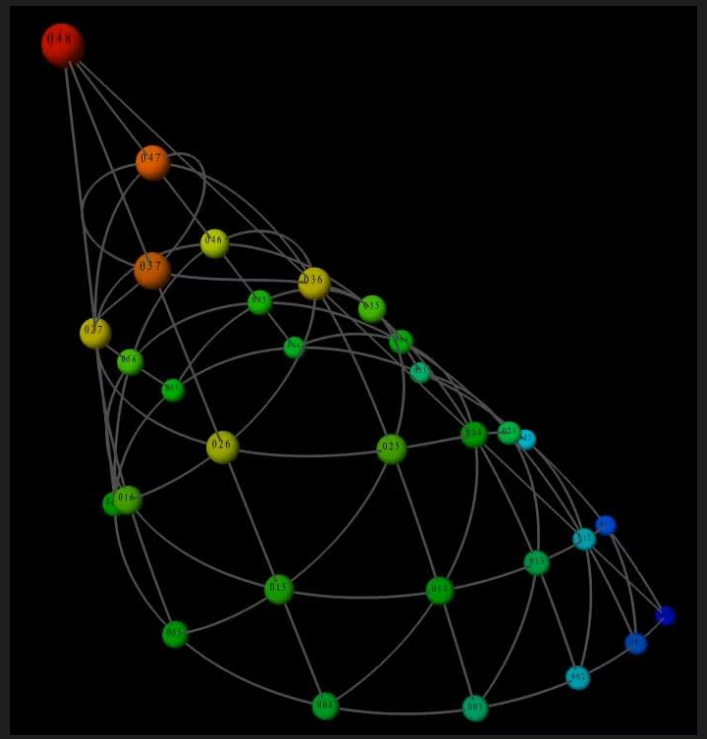


Geometrische Darstellungen

- Erforschung von „musikalischen Räumen“ zur Darstellung/Visualisierung von Eigenschaften und Zusammenhängen, die auf anderen (klassischen) Wegen nicht unmittelbar zugänglich oder offensichtlich sind

- *Beispiel:* Darstellung aller möglichen 3- und 4-stimmigen Akkord-Typen in **„Voice-Leading Spaces“**



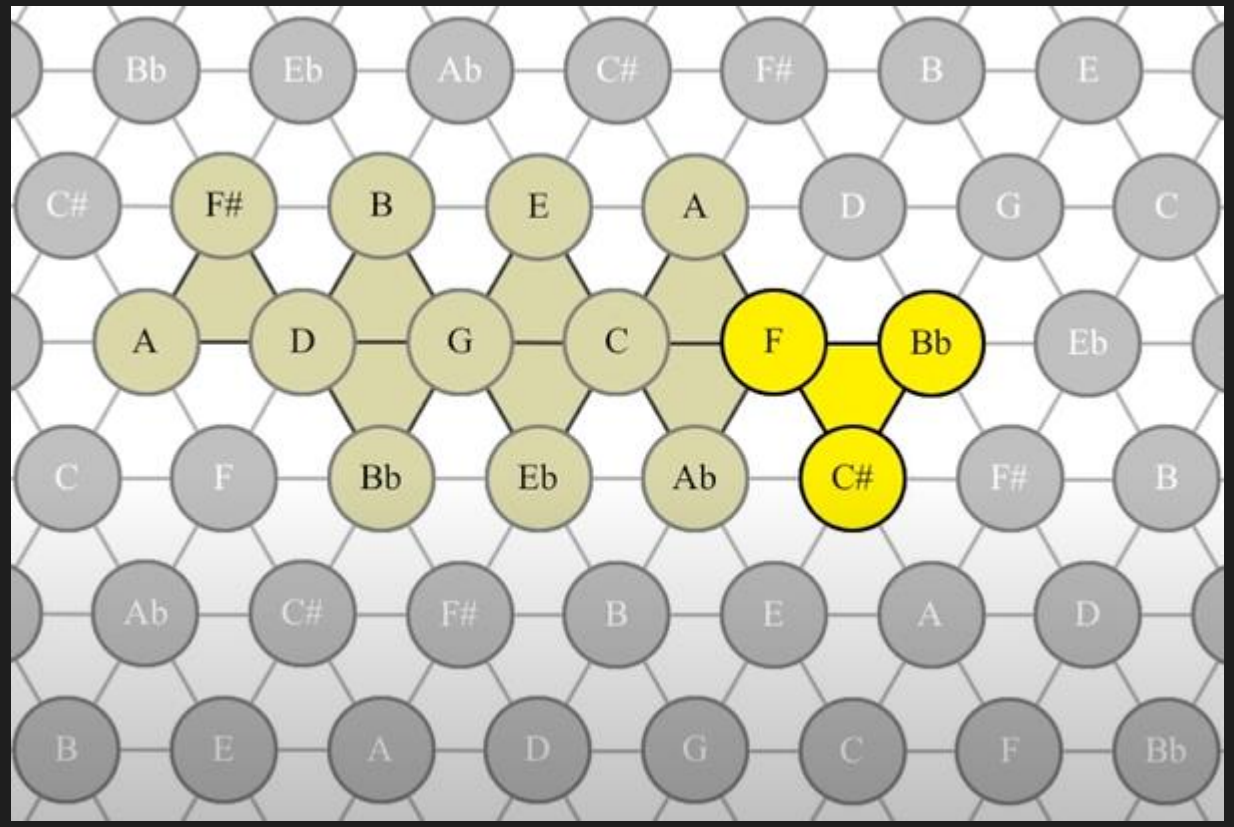


[D. Tymoczko (2011). *A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice.*

Geometrische Darstellungen

- Erforschung von „musikalischen Räumen“ zur Darstellung/Visualisierung von Eigenschaften und Zusammenhängen, die auf anderen (klassischen) Wegen nicht unmittelbar zugänglich oder offensichtlich sind

- *Beispiel:* Darstellung von Akkordfolgen als **Trajektorien im „Tonnetz“**



L. Bigo et al. (2013). *Computation and Visualization of Musical Structures in Chord-Based Simplicial Complexes.*

https://youtu.be/NQ7LkWCzKxI

04.

FRAMEWORKS & DATENSÄTZE

Einige Frameworks

- **music21**
  - Python-Library zur computergestützten Musikanalyse
  - Import und Export verschiedener symbolischer Formate, insbesondere MIDI & MusicXML
  - Häufig eingesetzte Standardlibrary in der Musikforschung
  - Dokumentation: https://www.music21.org/music21docs/
  - https://github.com/cuthbertLab/music21

- **Partitura**
  - Python-Library zur computergestützten Musikanalyse
  - Import von MIDI, MusicXML, MEI & **kern, Export von MIDI & MusicXML
  - https://github.com/CPJKU/partitura

- **pretty_midi**
  - Python-Library zur Analyse von MIDI-Dateien
  - https://craffel.github.io/pretty-midi/

- **jSymbolic 2.2**
  - Open-Source-Plattform zur Feature-Extraction aus symbolischen Formaten
  - Verfügbarkeit von aktuell 246 Features
  - https://jmir.sourceforge.net/jSymbolic.html

Einige Frameworks

- **MIDI Toolbox**
  - In *MATLAB* implementierte Funktionen zur Analyse und Visualisierung von MIDI-Dateien
  - https://github.com/miditoolbox
- **Humdrum**
  - Framework zur computergestützten Musikanalyse
  - https://www.humdrum.org
- **Melisma Music Analyzer**
  - Programm zur computergestützten Musikanalyse
  - http://davidtemperley.com/melisma-v2/

Einige Datensätze

- Auflistungen von MIDI-Datasets:
  - https://paperswithcode.com/datasets?mod=midi
  - https://magenta.tensorflow.org/datasets/
- music21-Korpus:
  - https://www.music21.org/music21docs/about/referenceCorpus.html
- Josquin Research Project:
  - https://josquin.stanford.edu
  - Polyphone Musik aus dem Zeitraum ca. 1420 – ca. 1520

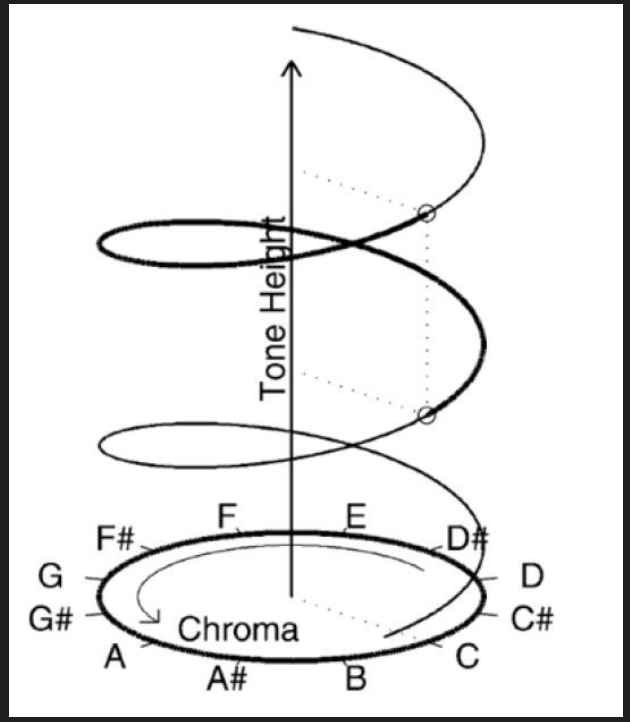
05.

Musiktheoretische Grundlagen

Chroma und Pitch-Class

- **Tonhöhenspirale** nach Géza Révész (1953)
- Darstellung der Tonhöhen (in einem 12-tönigen, gleichstufigen System) als 2-dimensionale Struktur:

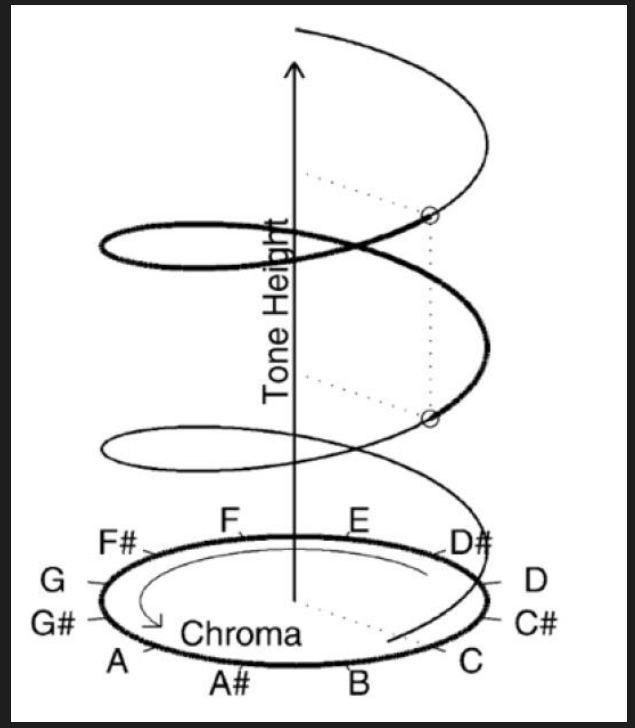
  - Pitch = Chroma + 12 · Tonal Height
  - „Tonhöhe = Tonigkeit (oder Toncharakter) + 12 · Oktavzahl“

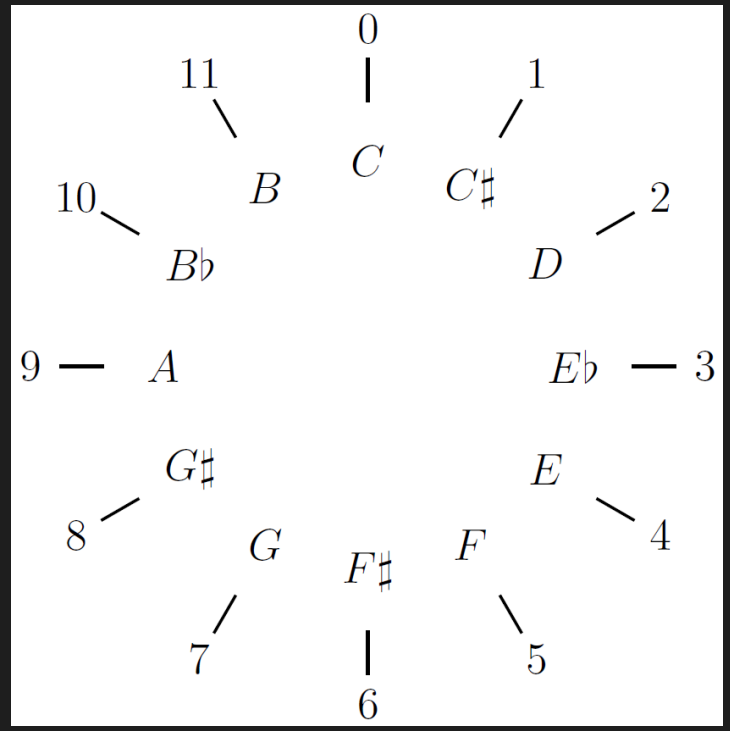


Zhiyao Duan & Emmanouil Benetos. Automatic Music Transcription, Tutorial at ISMIR 2015.

Chroma und Pitch-Class

- Eine Pitch-Class („Tonhöhen-Klasse“) ist definiert als die Menge aller Tonhöhen mit gleichem Chroma.
- Chroma und Pitch-Class im Wesentlichen als Synonyme verwendet.





Der Pitch-Class-Circle:
Eine Kreisförmige Darstellung
des Pitch-Class-Raums

Pitch-Class-Sets

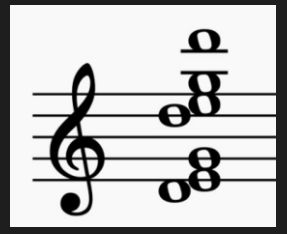
- Jede beliebige Menge von Pitch-Classes (ohne Verdopplung) bildet ein **Pitch-Class-Set**.

  Beispiele für Pitch-Class-Sets:

  [0, 4, 7] → gebildet aus den Pitch-Classes 0 (C), 4 (E), 7 (G)

  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] → gebildet aus allen Pitch-Classes

- Jeder beliebige Akkord kann als Pitch-Class-Set dargestellt werden.

  Beispiel:  → dazugehöriges Pitch-Class-Set:

Pitch-Class-Sets

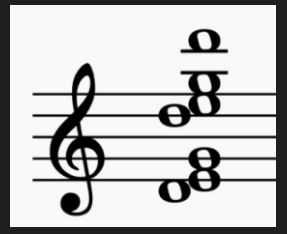
- Jede beliebige Menge von Pitch-Classes (ohne Verdopplung) bildet ein **Pitch-Class-Set**.

  Beispiele für Pitch-Class-Sets:

  [0, 4, 7] → gebildet aus den Pitch-Classes 0 (C), 4 (E), 7 (G)

  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11] → gebildet aus allen Pitch-Classes

- Jeder beliebige Akkord kann als Pitch-Class-Set dargestellt werden.

  Beispiel:  → dazugehöriges Pitch-Class-Set: [2, 4, 7]

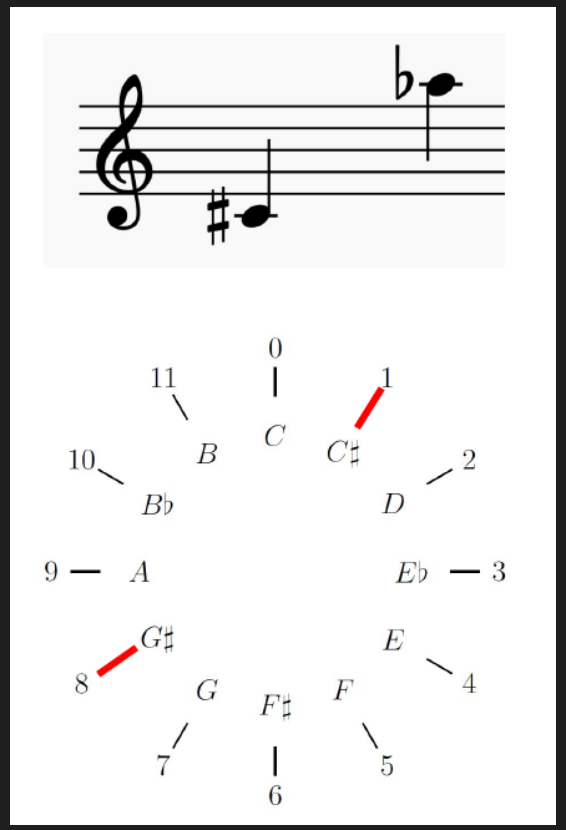
Pitch-Class-Sets

- Klassifikation aller möglichen Typen von Pitch-Class-Sets durch den Musiktheoretiker Allen Forte:
  → Jedem Pitch-Class-Set-Typ ist eine sogenannte **Forte-Number** zugewiesen.

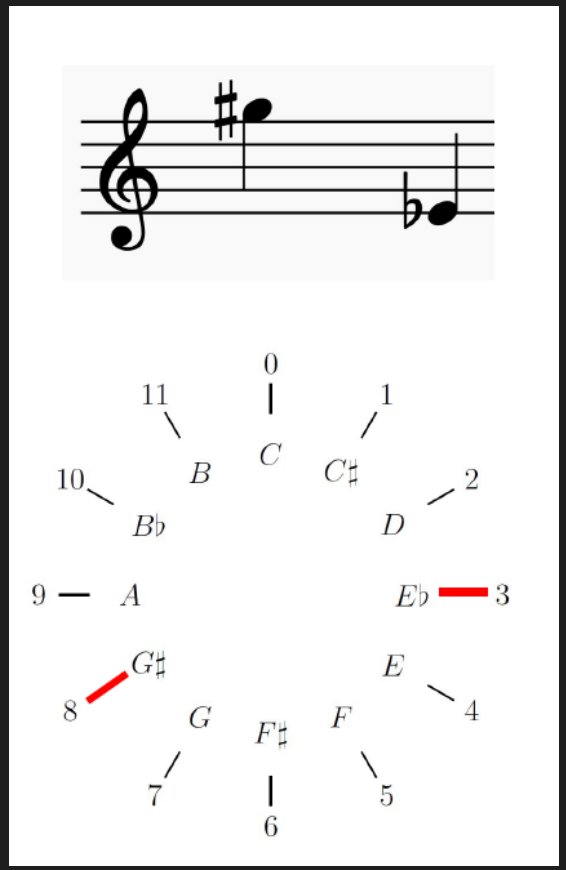
  Beispiele für verschiedene Typen von Pitch-Class-Sets:
  - Dur-Dreiklang → Forte-Number: 3-11A
  - Moll-Dreiklang → Forte-Number: 3-11B
  - Dominantseptakkord → Forte-Number: 4-27B

- Überblick über alle Typen von Pitch-Class-Sets: https://en.wikipedia.org/wiki/List_of_set_classes

Intervallklassen



- Ordered Pitch Interval: +19
- Unordered Pitch Interval: 19
- Ordered Pitch Class Interval: 7
- Unordered Pitch Class Interval: 5



- Ordered Pitch Interval: -17
- Unordered Pitch Interval: 17
- Ordered Pitch Class Interval: 7
- Unordered Pitch Class Interval: 5

06.

music21

Grundlagen

- Installation & Upgrade:
  - `pip install music21`
  - `pip install music21 --upgrade`
- Python-Bibliothek importieren:
  - `import music as m21`
  - oder: `from music21 import *`
- Beispiel für die Konfiguration der Kompatibilität mit *MuseScore:*

```
us = m21.environment.UserSettings()
us['musescoreDirectPNGPath'] = "C:/Program Files/MuseScore 4/bin/MuseScore4.exe"
us['musicxmlPath'] = "C:/Program Files/MuseScore 4/bin/MuseScore4.exe"
```