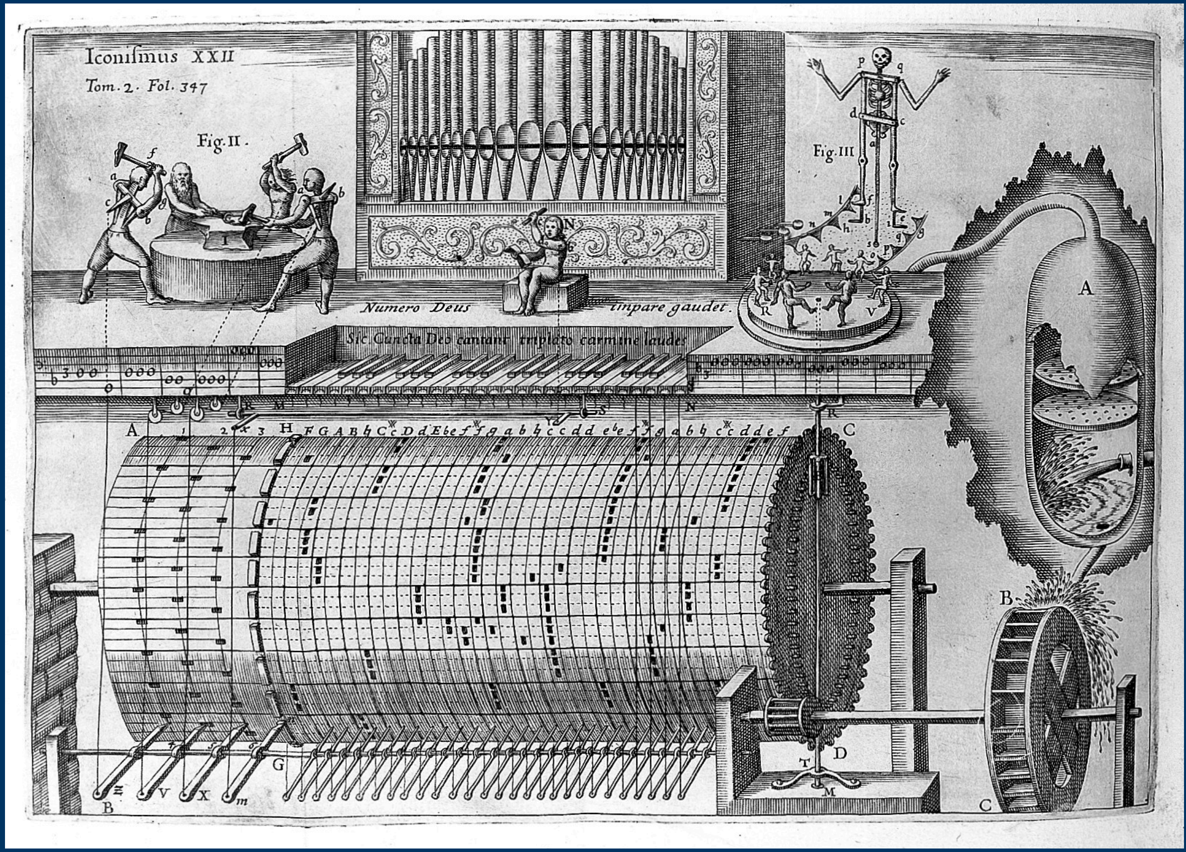
Music composition aided by symbolic AI

The Harmoniser project and the Fuxophone project

April 14, 2025
Peter Van Roy
peter.vanroy@uclouvain.be



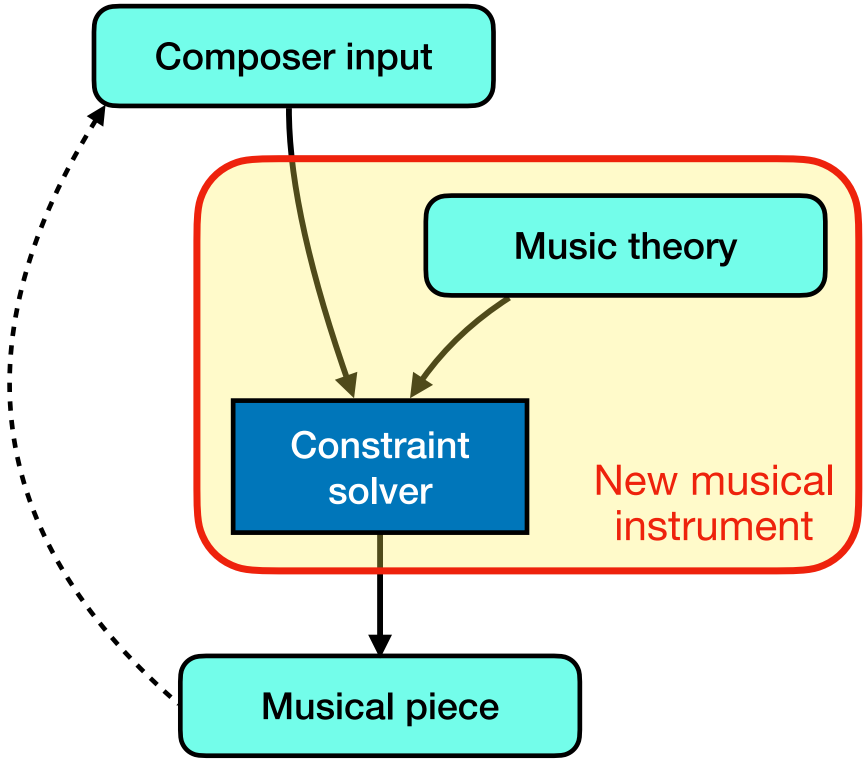
Pythagorean organ from Musurgia Universalis (Athanasius Kircher, 1650)

Motivation and overview

- We propose to build a practical tool for music composition using constraint programming
  - The tool gives freedom to the composer while respecting a musical style
  - Music is an appropriate application for constraint programming because it is highly combinatorial
  - Composers enter their musical ideas - as requirements and preferences - and the tool infers the optimal musical piece satisfying them according to a given musical style. The composer stays in the world of musical ideas whereas the tool does the heavy lifting of molding them into a concrete musical piece. While the creativity remains mostly with the composer, the tool does constraint solving which can give serendipitous results that complement the composer's creativity.
- Talk overview
  - Two music theories: Western tonal music and classical counterpoint
  - Musical examples for the two theories
  - Example formal rules for the two theories
  - Future work and conclusions

Creativity amplification

The composer “plays” an instrument defined by a music theory



- The composer input is given as musical ideas expressed through a GUI, internally encoded as constraints, preferences, and search heuristics
- The music theory is formalized as a constraint model that enforces the rules of a musical style
- The composer input and music theory are given as input to the constraint solver
- The solver finds an optimal solution that respects the music theory and returns it as a musical piece
- The composer evaluates the piece and iteratively updates the musical ideas

Two music theories

We have defined constraint models for two music theories

- **Harmoniser project** (Ph.D. research of Damien Sprockeels)
    - Music theory: Western tonal music (Gauldin 2004 + Duha 2016)
    - Composer gives global structure with added musical ideas
- **Fuxophone project** (Master's thesis projects of Thibault Wafflard, Anton Lamotte, Diego de Patoul, and Luc Cleenewerk)
    - Music theory: Counterpoint according to Johann Joseph Fux (1725)
    - Composer gives cantus firmus with added musical ideas
- Collaboration with IRCAM (Dr. Karim Haddad, composer and computer scientist)

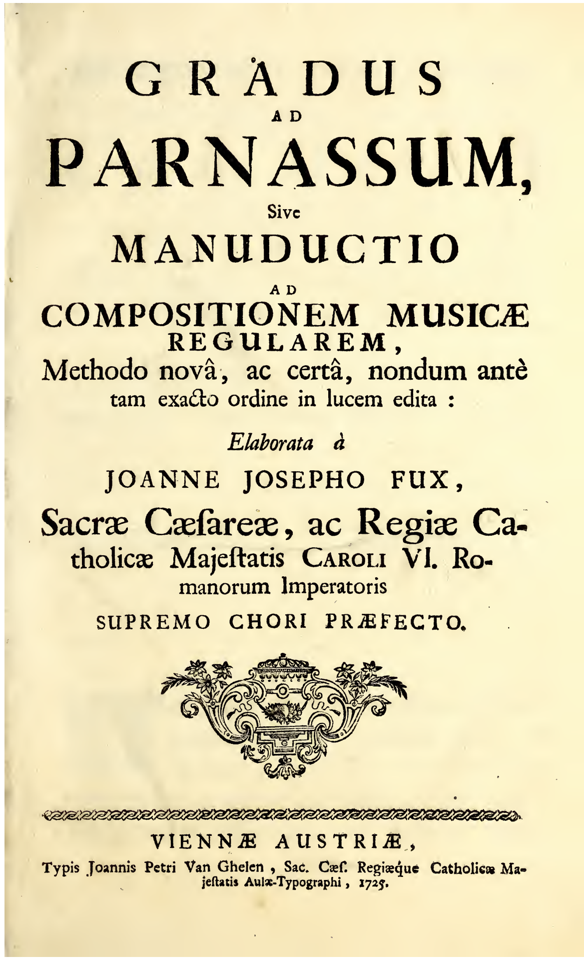
One constraint solver

Constraint programming is a form of symbolic AI

- Constraint programming is an approach to solve complex combinatorial problems based on formal logic
    - The constraint solver takes as input a conjunction of logical relations, called constraints, over finite domains of integers and gives concrete solutions that satisfy the relations
- The constraint solver does a combination of logical deduction with heuristic search
    - All solutions are *logical consequences* of the initial problem specification
    - Search can find *optimal solutions* according to a ranking (solutions of minimal total cost)
    - Search is *complete*, it can cover the whole space of solutions, which can find unexpected results
- We use the Gecode constraint solver
    - Gecode is a well-documented open-source software system written in C++ that has excellent performance and provides a wide variety of built-in constraints, combinators, and heuristics

Christian Schulte, Guido Tack, and Mikael Z. Lagerkvist. *Modeling and Programming with Gecode*. 2019. See: www.gecode.org

Counterpoint according to Fux


GRADUS
AD
PARNASSUM,
Sive
MANUDUCTIO
AD
COMPOSITIONEM MUSICÆ
REGULAREM,
Methodo novâ, ac certâ, nondum antè
tam exacto ordine in lucem edita :
Elaborata à
JOANNE JOSEPHO FUX,
Sacræ Cæſareæ, ac Regiæ Catholicæ Majeſtatis CAROLI VI. Romanorum Imperatoris
SUPREMO CHORI PRÆFECTO.

VIENNÆ AUSTRIÆ,
Typis Joannis Petri Van Ghelen, Sac. Cæſ. Regiæque Catholicæ Majeſtatis Aulæ-Typographi, 1725.

Johann Joseph Fux

- Counterpoint is a style of music consisting of two or more simultaneous voices that are harmonically interdependent. Counterpoint is a precursor to the tonal music that is considered to be the basis of Western classical music.
- The treatise *Gradus ad Parnassum* by Johann Joseph Fux published in 1725 is the most influential presentation of the rules of counterpoint in the Palestrinian style.
  - We have formalized this treatise as a constraint model in Gecode. We used the English translation by Alfred Mann (1965) and the French translation by Simonne Chevalier (2019).
- Counterpoint consists of a fixed melodic line of whole notes, the cantus firmus, accompanied by a number of counterpoint voices, where Fux presents 2, 3, and 4-voice counterpoint.
- In the Fux theory, each counterpoint voice can be in one of five species: whole notes, half notes, quarter notes, half notes with ligatures, and florid counterpoint (mixture of the four species)

Fuxophone musical example

Example by Thibault Wafflard

Composer input

- We give a cantus firmus in 17 measures, a basic melody with one whole note per measure, including chromatic steps not usual for counterpoint

- The FuxCP solver generates the counterpoint melody for two voices. This example tests the Fux theory outside of its original scope. A chromatic bass line (common in contemporary music) is used as cantus firmus. Instrumentation has been added by the composer to give texture but the melody is not altered. Two solutions with different preferences are juxtaposed.

- The result is surprising and shows the ability of the solver to adapt to a cantus firmus that is not classic. This demonstrates that this kind of solver can be used for freer music than just classical counterpoint.

Musical piece generated by FuxCP

https://www.youtube.com/watch?v=9yB4OGr4Cgk (example at 06:02)

Example rule for Fux counterpoint

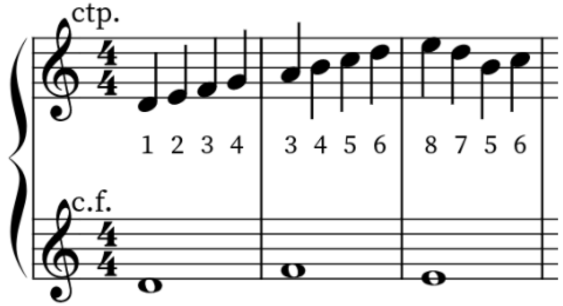
We give an example of a rule for the third species (quarter notes)

- Harmonic rule 3.H1 of the third species:
  If five notes follow each other by joint degrees in the same direction, then the harmonic interval of the third note must be consonant

- Chevalier's French translation says:
  If it happens that five quarter notes follow each other by joint degrees, either ascending or descending, the first one must be consonant, the second one may be dissonant, the third one again necessarily consonant, the fourth one may be dissonant if the fifth one is a consonance.

- The precise meaning of this rule is not immediately clear from the text of the treatise. For example, what five-note tuples are targeted? From the context, we infer that the rule targets the four quarter notes of each measure plus the first note of the next measure. Furthermore, given rule 2.H1 (first note of each measure must be consonant), we infer that there is only one additional constraint, on the third note.

- The complete formalization of 4-voice Fux counterpoint for all species consists of 62 rules and preferences, supplemented by auxiliary definitions, all defined in approximately 7500 lines of C++ for the Gecode constraint solver



Third species of two-voice counterpoint

$$\forall j \in [0, m-1)$$
$$\left(\bigwedge_{i=0}^{3} M[i,j] \leq 2\right) \wedge \left(\bigwedge_{i=0}^{3} M_{brut}[i,j] > 0 \vee \bigwedge_{i=0}^{3} M_{brut}[i,j] < 0\right)$$
$$\implies IsCons[2,j]$$

$M[i, j]$: absolute step for note i in measure j

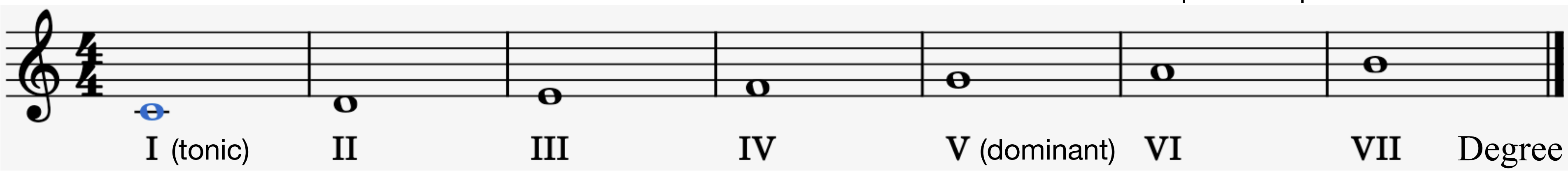
$M_{brut}[i, j]$: positive or negative step

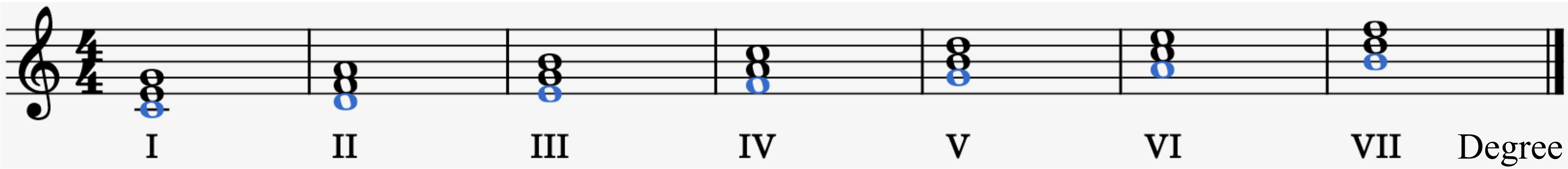
$IsCons[i, j]$: note i in measure j is consonant

Western tonal music (for Harmoniser)

Tonal music is organized around a central note and its scale

- Tonality = (key, mode)
    - Key: any of the twelve notes
    - Mode: major or minor
    - Scale: commonly 7 notes per octave
- Chord: a set of simultaneous notes
    - Chord = (root note, quality, state)
- Degree: role of a chord in a tonality
    - Determines root and possible qualities and states





C major tonality

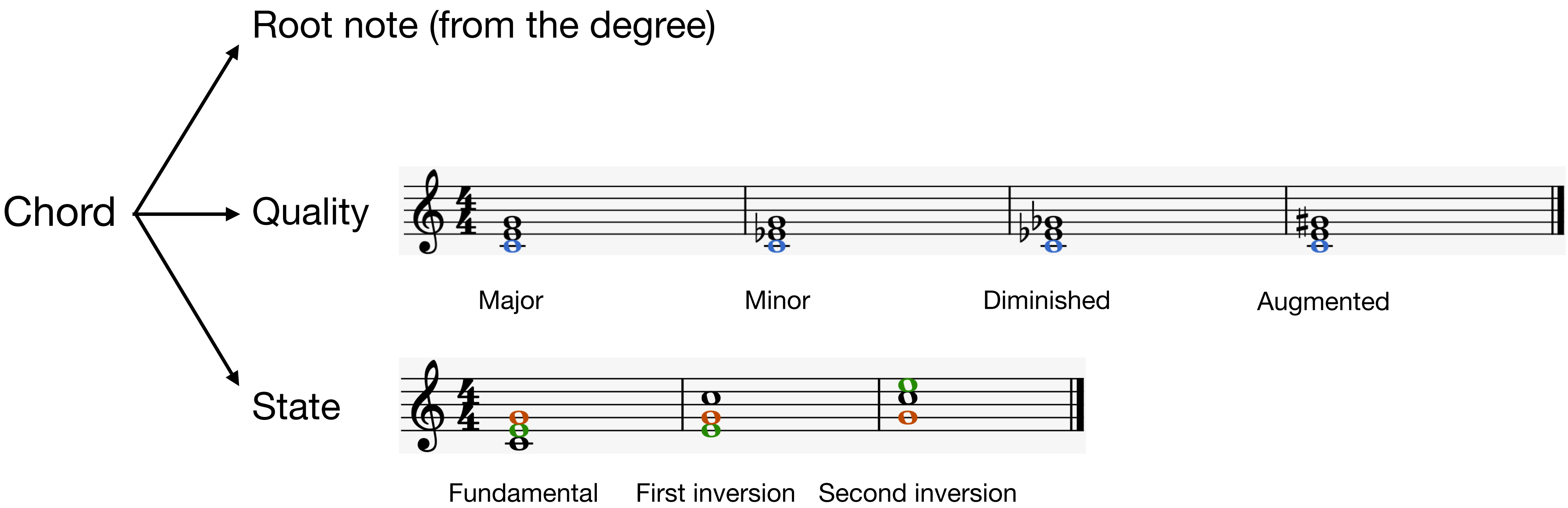
$$f(tonality, degree) = chord$$

Robert Gauldin. *Harmonic Practice in Tonal Music, second edition*. W.W. Norton & Co. Inc., 2004.

Isabelle Duha. *L'harmonie en liberté: de la mémoire à l'improvisation*. Gérard Billaudot, Armiane Imp., 2016.

Western tonal music (for Harmoniser)

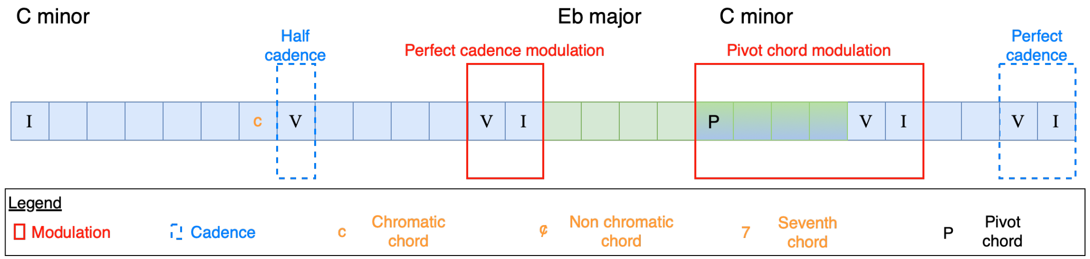
Chords as the basic abstraction



Harmoniser musical example

Example by Damien Sprockeels

Composer input



- We require a 14-chord progression in C minor, modulating to an 8-chord E♭ major progression and then back to a 6-chord C minor progression, with a perfect cadence modulation and a pivot chord modulation
- We add the following musical ideas:
  - Chord 1 is I
  - No III or VII except in pivot
  - Only dominant chords have a 7th
  - Chord 7 is chromatic
  - Chord 8 is half cadence
  - II only in first inversion
  - Perfect cadence in pivot modified to interrupted cadence
  - Perfect cadence at end

Musical piece generated by Harmoniser

Rhythm and ornaments added by the composer

https://soundcloud.com/anonymous123-414149293/ijcai-example

Harmoniser composition process

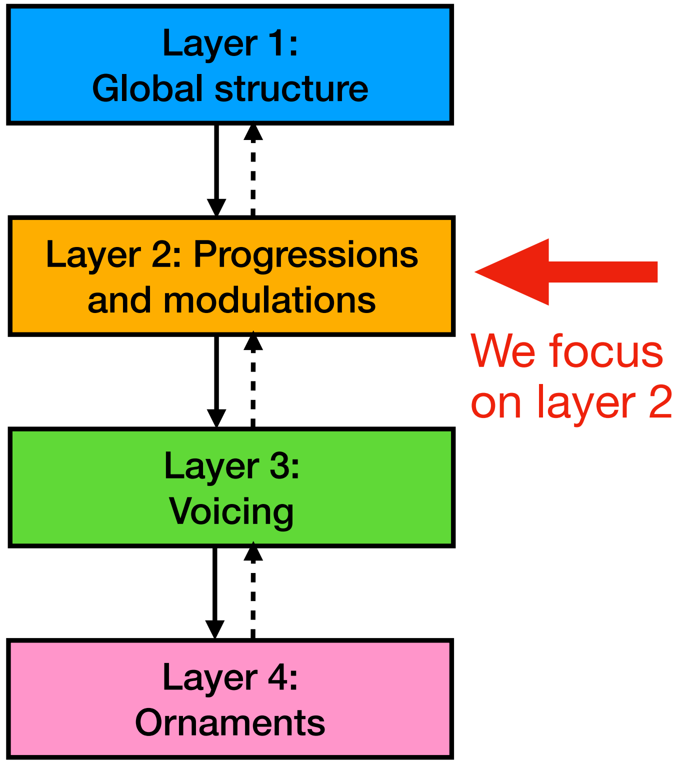
Music composition is a vast subject!

- We define a *four-layer process* that corresponds to what many composers do
  - This splits one complex monolithic problem into four simpler subproblems
- In each layer, the composer introduces musical ideas and the constraint solver determines the optimal solution that embodies them
  - The solution is passed to the next layer, where new ideas are added
  - The musical piece is complete after the fourth layer
  - The composer can move up and down the layers, adding and removing musical ideas, until the musical piece is satisfactory

Harmoniser composition process

Four layers that each embodies a subset of tonal music theory

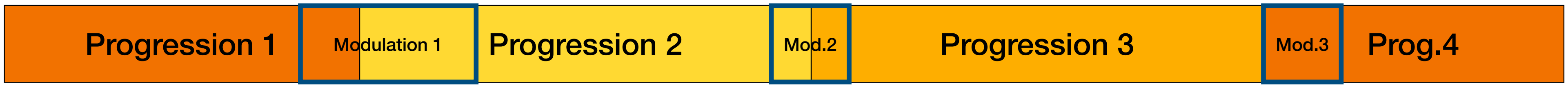
- *Global structure*: decomposition of the musical piece into progressions, which are sequences of chords each in a single tonality, connected by modulations
- *Progressions and modulations*: harmonic development within each progression as well as the transitions between progressions
- *Voicing*: assigning the actual notes played for each chord, typically using four voices, taking account interaction between notes in a chord and between notes in a voice
- *Ornaments*: local embellishments, such as passing notes or appoggiaturas, that add essential complexity to the piece



Layer 2: Chord progressions and modulations

Overview

- Given a structure for a musical piece, we generate a sequence of chord degrees
- Input:
  - Number of chords of the piece n
  - Number of progressions l
  - Tonality of each progression t_i
  - Modulation types $type_m$
  - Modulation start and endpoints s_m and e_m
- Output:
  - Chords are uniquely identified by a triple (r,q,s):
    - Chord roots $R[]$
    - Chord qualities $Q[]$
    - Chord states $S[]$
  - Chord degrees for each progression $D_i[]$



Damien Sprockeels and Peter Van Roy. Towards a Practical Tool for Music Composition: Using Constraint Programming to Model Chord Progressions and Modulations. In *International Joint Conference on Artificial Intelligence 2025 (submitted)*.

Layer 2: Chord progressions and modulations

Degree transition matrix *T* for chord progressions in a tonality

Second chord

	I	II	III	IV	V	VI	VII	Vda
I	1	1	1	1	1	1	1	
II	1	1		1	1			1
III						1		
IV	1	1		1	1		1	1
V	1			1	1	1		
VI		1		1	1			1
VII	1		1					
Vda	A				1			

First chord

- Only some chord degree transitions are allowed
- Using only diatonic chords can become repetitive

Constraint between adjacent chord degrees

$$\forall i \in [0,l[,\ c'_i \in [0,e_i - b_i[\quad T[D_i[c'_i], D_i[c'_i + 1]] = 1$$

Layer 2: Chord progressions and modulations

Degree transition matrix *T* for chord progressions in a tonality

	I	II	III	IV	V	VI	VII	Vda	V/II	V/III	V/IV	V/V	V/VI	V/VII
I	1	1	1	1	1	1	1		1	1	1	1	1	1
II	1	1		1	1			1			1	1		
III						1							1	
IV	1	1		1	1		1	1	1			1		1
V	1			1	1	1					1		1	
VI		1		1	1			1	1		1	1		
VII	1		1							1				
Vda	A				1				B					
V/II		1										1		
V/III			1										1	
V/IV				1										1
V/V					1									
V/VI						1			1					
V/VII	D						1		E	1				

- Only some chord degree transitions are allowed
- Using only diatonic chords can become repetitive
- Borrowed chords can be used to add color

$$\forall i \in [0,l[,\ c_i' \in [0,e_i - b_i[\quad T[D_i[c_i'], D_i[c_i' + 1]] = 1$$

Layer 2: Chord progressions and modulations

Degree transition matrix *T* for chord progressions in a tonality

	I	II	III	IV	V	VI	VII	Vda	V/II	V/III	V/IV	V/V	V/VI	V/VII	♭II	6△
I	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
II	1	1		1	1			1			1	1				
III						1							1			
IV	1	1		1	1		1	1	1			1		1	1	1
V	1			1	1	1					1		1			
VI		1		1	1			1	1		1	1			1	1
VII	1		1							1						
Vda	A				1				B						C	
V/II		1										1				
V/III			1										1			
V/IV				1										1		
V/V					1											
V/VI						1			1							
V/VII	D						1		E	1					F	
♭II					1			1								
6△	G				1			1	H						I	

- Only some chord degree transitions are allowed
- Using only diatonic chords can become repetitive
- Borrowed chords can be used to add color
- Chromatic chords related to the tonality can also be used

$$\forall i \in [0,l[,\ c'_i \in [0,e_i - b_i[\quad T[D_i[c'_i], D_i[c'_i + 1]] = 1$$

Layer 2: Chord progressions and modulations

Connecting chord degrees to chord qualities and states

	M	m	°	△	7	M7	m7	°7	Ø7	mM7	6△
I	1					1					
II		1					1				
III		1					1				
IV	1					1					
V	1				1			1			
VI		1					1				
VII			1						1		
Vda	1										
V/II	1				1			1			
V/III	1				1			1			
V/IV	1				1			1			
V/V	1				1			1			
V/VI	1				1			1			
V/VII	1				1			1			
♭II	1										
6△											1

Major tonalities

Link between chord degrees and qualities for major and minor tonalities

Minor tonalities

	M	m	°	△	7	M7	m7	°7	Ø7	mM7	6△
I		1					1				
II			1						1		
III	1					1					
IV		1					1				
V	1				1			1			
VI	1					1					
VII			1					1			
Vda		1									
V/II	1				1			1			
V/III	1				1			1			
V/IV	1				1			1			
V/V	1				1			1			
V/VI	1				1			1			
V/VII	1				1			1			
♭II	1										
6△											1

constraints

Possible states per chord degree P																
	I	II	III	IV	V	VI	VII	Vda	V/II	V/III	V/IV	V/V	V/VI	V/VII	♭II	6△
Fund.	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
1^{st}	1	1		1	1		1		1	1	1	1	1	1	1	
2^{nd}					1		1	1	1	1	1	1	1	1		
3^{rd}					1				1	1	1	1	1	1		
4^{th}																

Link between chord degrees and states

	I	II	III	IV	V	VI	VII	Vda	V/II	V/III	V/IV	V/V	V/VI	V/VII	♭II	6△
I	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1
II	1	1		1	1			1			1	1				
III						1							1			
IV	1	1		1	1		1	1	1			1		1	1	1
V	1			1	1	1					1		1			
VI		1		1	1			1	1		1	1			1	1
VII	1		1							1						
Vda	A				1				B						C	
V/II		1										1				
V/III			1										1			
V/IV				1										1		
V/V					1											
V/VI						1			1							
V/VII	D						1		E	1					F	
♭II					1			1								
6△	G				1			1	H						I	

Degree transition matrix *T*

constraints

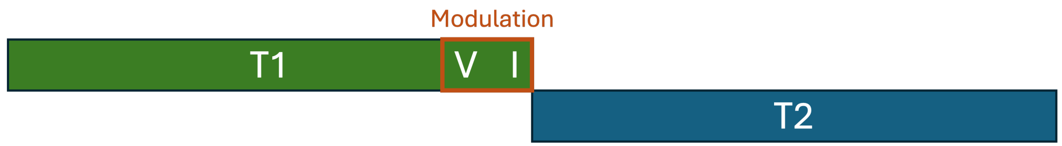
Link between chord degrees and chord notes

Chord composition L																
	I	II	III	IV	V	VI	VII	Vda	V/II	V/III	V/IV	V/V	V/VI	V/VII	♭II	6△
Fund./root	I	II	III	IV	V	VI	VII	I	VI	VII	I	II	III	VII	II	VI
1^{st}/third	III	IV	V	VI	VII	I	II	III	I	II	III	IV	V	II	IV	I
2^{nd}/fifth	V	VI	VII	I	II	III	IV	V	III	IV	V	VI	VII	IV	VI	III
3^{rd}/seventh	VII	I	II	III	IV	V	VI	VII	V	VI	VII	I	II	VI	I	V
4^{th}/ninth	II	III	IV	V	VI	VII	I	II	VII	I	II	III	IV	V	III	II

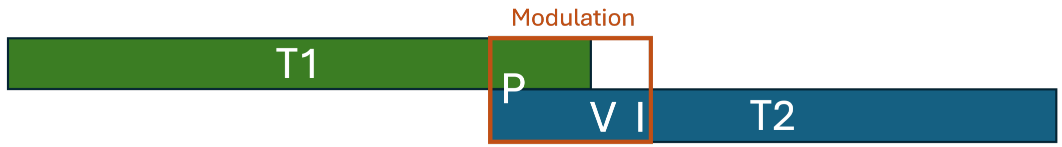
Layer 2: Chord progressions and modulations

Modulations connect adjacent progressions

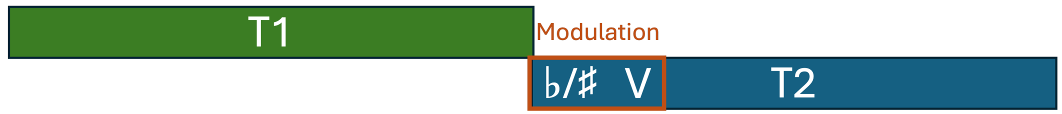
- Modulations to neighboring tonalities:
  - Perfect cadence modulation



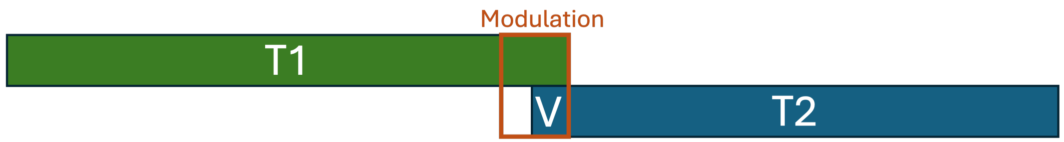
  - Pivot chord modulation



  - Alteration modulation



  - Chromatic modulation



Future work and conclusions

We have some promising results but a lot remains to be done

- Work completed
  - Harmoniser project: layer 2 model (progressions + modulations) and layer 3 model (voicing)
  - Fuxophone project: constraint model for 4-voice counterpoint all species
  - All constraint models are available as open-source software using Gecode
- Work in progress
  - Harmoniser project
    - Global structure (layer 1), ornaments (layer 4), rhythm (all layers)
    - Composer evaluation
  - Fuxophone project
    - Validation of Fux constraint model
    - Musical exploration and extensions
  - For both projects: Digital Audio Workstation support
    - Plug-ins with graphical user interface for practical use by composers

These projects can be continued in many ways, for example:

- Extend matrix T with additional degrees or extensional constraints
- Use statistical AI (machine learning) to learn new music theories

References

All documents available on the Web or upon request

- **Harmoniser project**
  - Damien Sprockeels and Peter Van Roy. Exprimer les Idées Musicales avec la Programmation par Contraintes: un Assistant pour Compositeurs. In *Journées Francophones de Programmation par Contraintes,* Lens, France, June 24-26, 2024.
  - Damien Sprockeels and Peter Van Roy. Expressing Musical Ideas with Constraint Programming using a Model of Tonal Harmony. In *International Joint Conference on Artificial Intelligence (IJCAI 2024),* Jeju Island, South Korea, Aug. 2024.
  - Damien Sprockeels and Peter Van Roy. Towards a Practical Tool for Music Composition: Using Constraint Programming to Model Chord Progressions and Modulations. In *International Joint Conference on Artificial Intelligence,* 2025 (submitted).
  - Damien Sprockeels and Peter Van Roy. The Harmoniser Project: A Practical Four-Layer Tool for Music Composition based on Constraints. In *Journées Francophones de Programmation par Contraintes,* 2025 (submitted).
  - Cédric Niyikiza. Plug-In to Support Music Composition with Constraint Programming. Master thesis, UCLouvain, June 2025 (in progress).

- **Fuxophone project**
  - Damien Sprockeels, Thibault Wafflard, Peter Van Roy, and Karim Haddad. A Constraint Formalization of Fux's Counterpoint. *Journées d'Informatique Musicale (JIM 2023),* Paris, France, May 24-26, 2023.
  - Thibault Wafflard. FuxCP: A Constraint Programming Based Tool Formalizing Fux's Musical Theory of Counterpoint. Master thesis, UCLouvain, June 5, 2023.
  - Anton Lamotte. FuxCP: Constraint Programming Formalisation of Three-Voice Counterpoint According to Fux. Master thesis, UCLouvain, Jan. 8, 2024.
  - Diego de Patoul and Luc Cleenewerk. FuxCP: A Constraint Programming Formalization of Fux's Musical Theory of Four-Voice Counterpoint. Master thesis, UCLouvain, Aug. 19, 2024.
  - Tom Lai. Validation and Exploration of a Constraint Model for Four-Voice Counterpoint According to Fux. Master thesis, UCLouvain, Aug. 2025 (in progress).
  - Chris Bakashika. Interactive Music Composition Interface: Integrating Fux's Musical Theory of Four-Voice Counterpoint with JUCE. Master thesis, UCLouvain, June 2025 (in progress).