

Demo: Counterpoint by Construction

Youyou Cong and John Leo

Composition Rules and Typing Rules

- ▶ Composition rules make music sound good
- ▶ Typing rules make programs meaningful

Well-typed music does not sound wrong
(Szamozvancev & Gale, Haskell '17)

Dependent Types as Precise Specification

Length-Indexed Vector Type

$v : \text{Vec } \mathbb{N} \ n$ means “ v has n natural numbers”

$\text{nth} : \forall m. \text{Vec } \mathbb{N} \ m \rightarrow \{n : \mathbb{N} \mid n \leq m\} \rightarrow \mathbb{N}$

$\text{nth } v \ n = \dots$

$\text{nth } [1] \ 1 \ \checkmark$

$\text{nth } [1] \ 2 \ \times$

Dependent Types for Music Rules

- ▶ Length-indexed vector type
(for composition of parallel voices)
- ▶ Mode-indexed datatypes
(for composition of phrases)

This Work

Music Tools: An Agda Library for Music Composition

- ▶ Full dependent types for encoding rules
- ▶ Haskell FFI for generating MIDI
- ▶ Demo: First-species counterpoint

The Music Tools Library

Basic Ingredients

```
data Pitch : Set where  
  pitch :  $\mathbb{N} \rightarrow$  Pitch
```

```
data Duration : Set where  
  duration :  $\mathbb{N} \rightarrow$  Duration
```

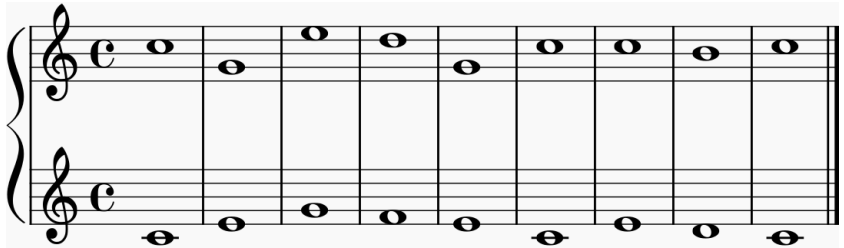
```
data Note : Set where  
  note : Duration  $\rightarrow$  Pitch  $\rightarrow$  Note  
  rest : Duration  $\rightarrow$  Note
```

Building Music

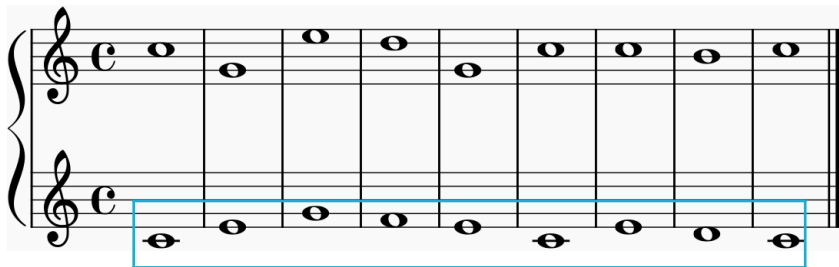
```
data Music : Set where
  -- single note (base case)
  note : Note → Music
  -- sequential composition
  _::_ : Music → Music → Music
  -- parallel composition
  _||_ : Music → Music → Music
```


First-Species Counterpoint

Counterpoint: Interaction of Multiple Voices



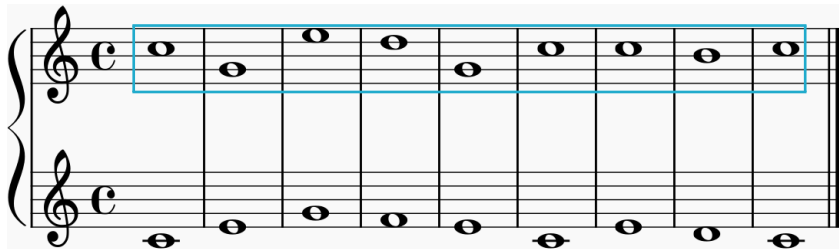
Counterpoint: Interaction of Multiple Voices



cantus firmus

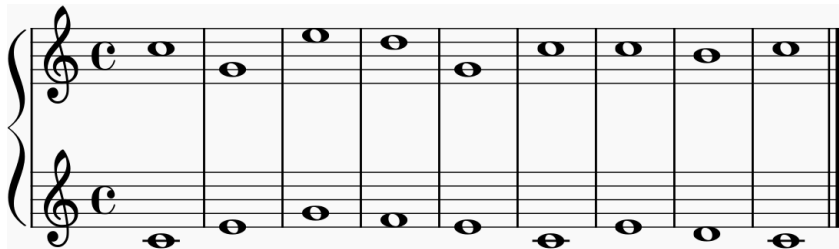
Counterpoint: Interaction of Multiple Voices

counterpoint



Intervals and Motion

per8 min3 maj6 maj6 min3 per8 maj6 maj6 per8



Rules for First-Species Counterpoint

All intervals must be *consonant*

A musical staff with two staves, both in treble clef and common time (C). The top staff contains a series of half notes: C4, D4, E4, F4, G4, A4, and B4. The bottom staff contains a series of half notes: C4, D4, E4, F4, G4, A4, and B4. Below the staff, the intervals between the notes are labeled: 2nd, 3rd, 4th, 5th, 6th, 7th, and 8th. Below each label is a red 'X' or a green checkmark indicating whether the interval is consonant or dissonant.

Interval	Consonant
2nd	×
3rd	✓
4th	×
5th	✓
6th	✓
7th	×
8th	✓

Representing Valid Intervals

```
data IntervalQuality : Set where
```

```
  min3  : IntervalQuality
```

```
  maj3  : IntervalQuality
```

```
  per5  : IntervalQuality
```

```
  min6  : IntervalQuality
```

```
  maj6  : IntervalQuality
```

```
  per8  : IntervalQuality
```

```
  min10 : IntervalQuality
```

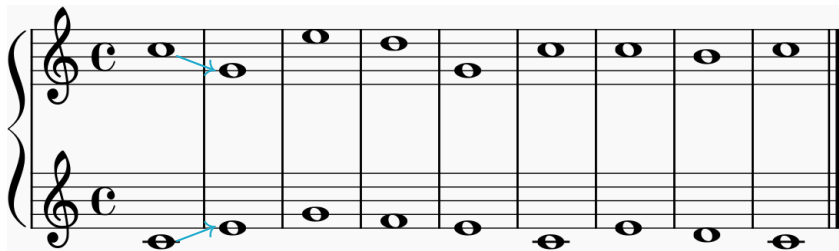
```
  maj10 : IntervalQuality
```

```
PitchInterval : Set
```

```
PitchInterval = Pitch × IntervalQuality
```

Intervals and Motion

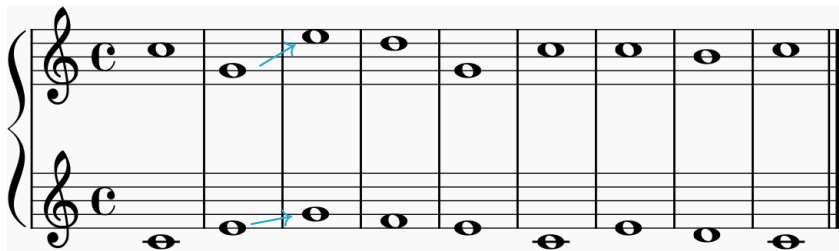
ctr sim par sim ctr obl par ctr



where ctr: contrary, sim: similar, par: parallel, obl: oblique

Intervals and Motion

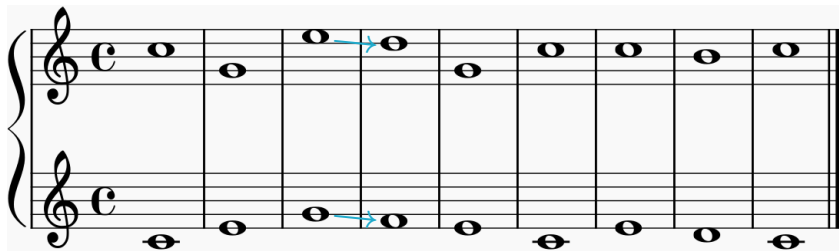
ctr **sim** par sim ctr obl par ctr



where ctr: contrary, sim: similar, par: parallel, obl: oblique

Intervals and Motion

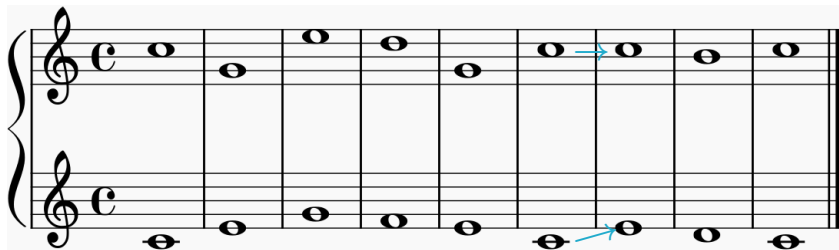
ctr sim **par** sim ctr obl par ctr



where ctr: contrary, sim: similar, par: parallel, obl: oblique

Intervals and Motion

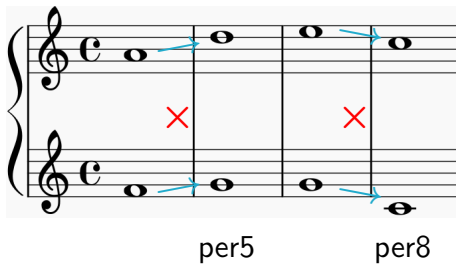
ctr sim par sim ctr **obl** par ctr



where ctr: contrary, sim: similar, par: parallel, obl: oblique

Rules for First-Species Counterpoint

Perfect intervals must be approached
by ***non-similar motion***

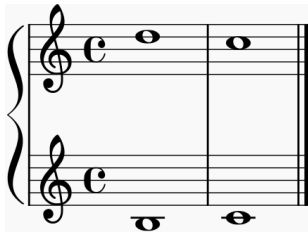


Constraining Motion

```
motionOK : (i1 i2 : Interval) → Set
motionOK i1 i2 with motion i1 i2
  | isPerfectInterval i2
motionOK i1 i2 | contrary | _ = T
motionOK i1 i2 | oblique  | _ = T
motionOK i1 i2 | parallel | false = T
motionOK i1 i2 | parallel | true  = ⊥
motionOK i1 i2 | similar  | false = T
motionOK i1 i2 | similar  | true  = ⊥
```

Rules for First-Species Counterpoint

Final interval must be approached by
contrary step-wise motion



Building Correct Counterpoint

head interval

```
data FirstSpecies : PitchInterval → Set where  
  :
```

Building Correct Counterpoint

```
data FirstSpecies : PitchInterval → Set where
  cadence2 : (p : Pitch) →
    FirstSpecies (transpose (+ 2) p , maj6)
  cadence7 : (p : Pitch) →
    FirstSpecies (transpose -[1+ 0 ] p , min10)
  :
```


Building Correct Counterpoint

```
data FirstSpecies : PitchInterval → Set where
  cadence2 : (p : Pitch) →
    FirstSpecies (transpose (+ 2) p , maj6)
  cadence7 : (p : Pitch) →
    FirstSpecies (transpose -[1+ 0 ] p , min10)
_::_ : (pi : PitchInterval) →
  {pj : PitchInterval} →
  {- : motionOK pi pj} → -- implicit
  FirstSpecies pj →
  FirstSpecies pi
```

Demo

Future Work

- ▶ Higher-species counterpoint
 - ▶ Counterpoint has more notes than cantus firmus
 - ▶ Dissonant intervals are allowed on weak beats
- ▶ Automatic generation of counterpoint
 - 😊 Imperfect intervals, contrary motion
 - 😞 Big intervals, repeats
- ▶ Harmonic analysis and synthesis