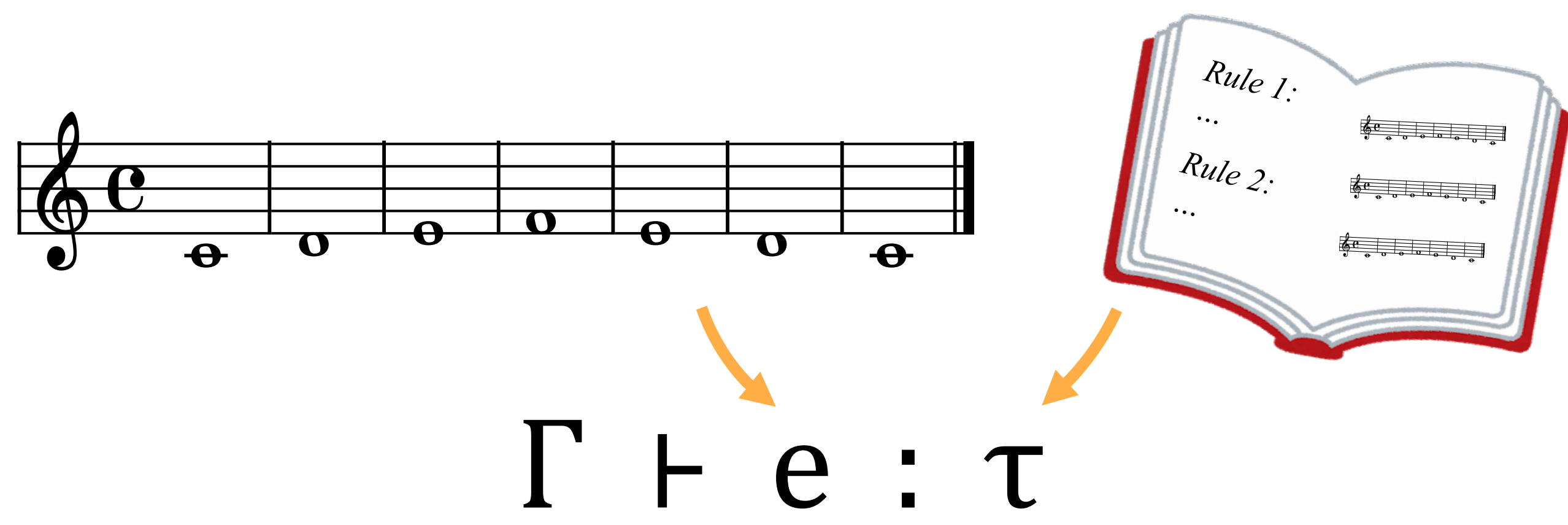


型を利用した音楽自動生成に向けて

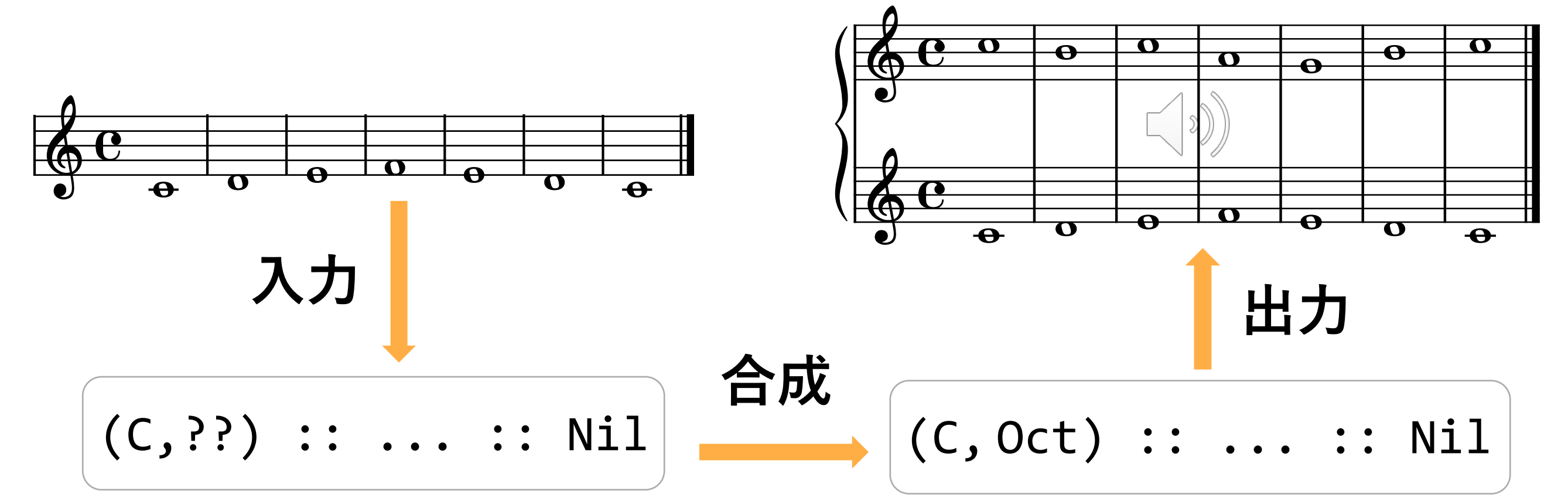
叢 悠悠（東京工業大学）

背景：型による作曲規則の表現^[1]



型が付いた音楽 = 正しさの証明

目標：型を用いた和音列の自動生成



応用：教育用譜例、正しい音楽のコーパス

篩型による対位法^[2]の形式化

-- ピッチ

data Pitch where

C :: Pitch

Cis :: Pitch

D :: Pitch

⋮

-- 距離

data Interval where

Uni :: Interval

Min2 :: Interval

Maj2 :: Interval

⋮

-- 距離の規則（すべての和音は協和音）

measure allConsonant :: PIs -> Bool where

Nil -> True

Cons pi pis -> ...

-- 和音はピッチと距離のペア

data PI where

Pair :: Pitch -> Interval -> PI

-- 和音列は和音のリスト

data PIs where

Nil :: PIs

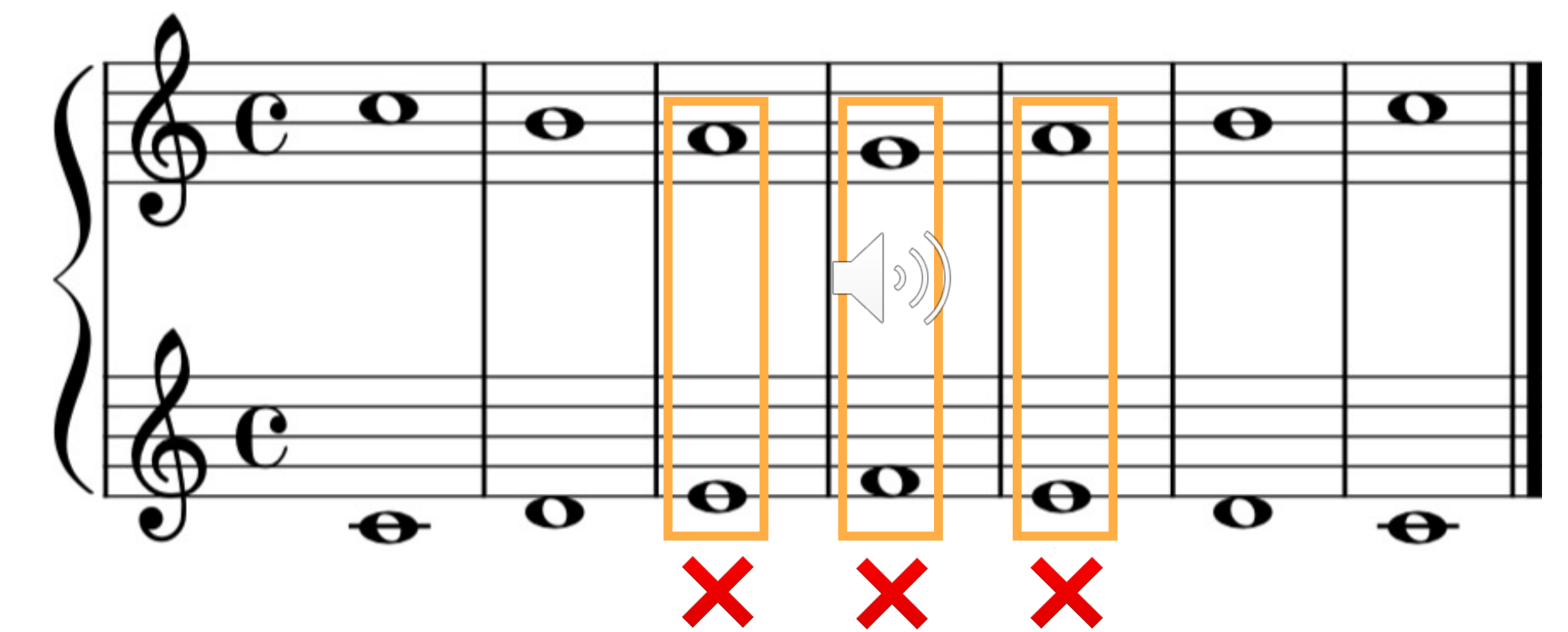
Cons :: PI -> PIs -> PIs

-- 対位法に従った和音列

type CP = {PIs | allConsonant _v && motionOK _v}

篩型の制約内で
使用できる関数

対位法の規則
による絞り込み

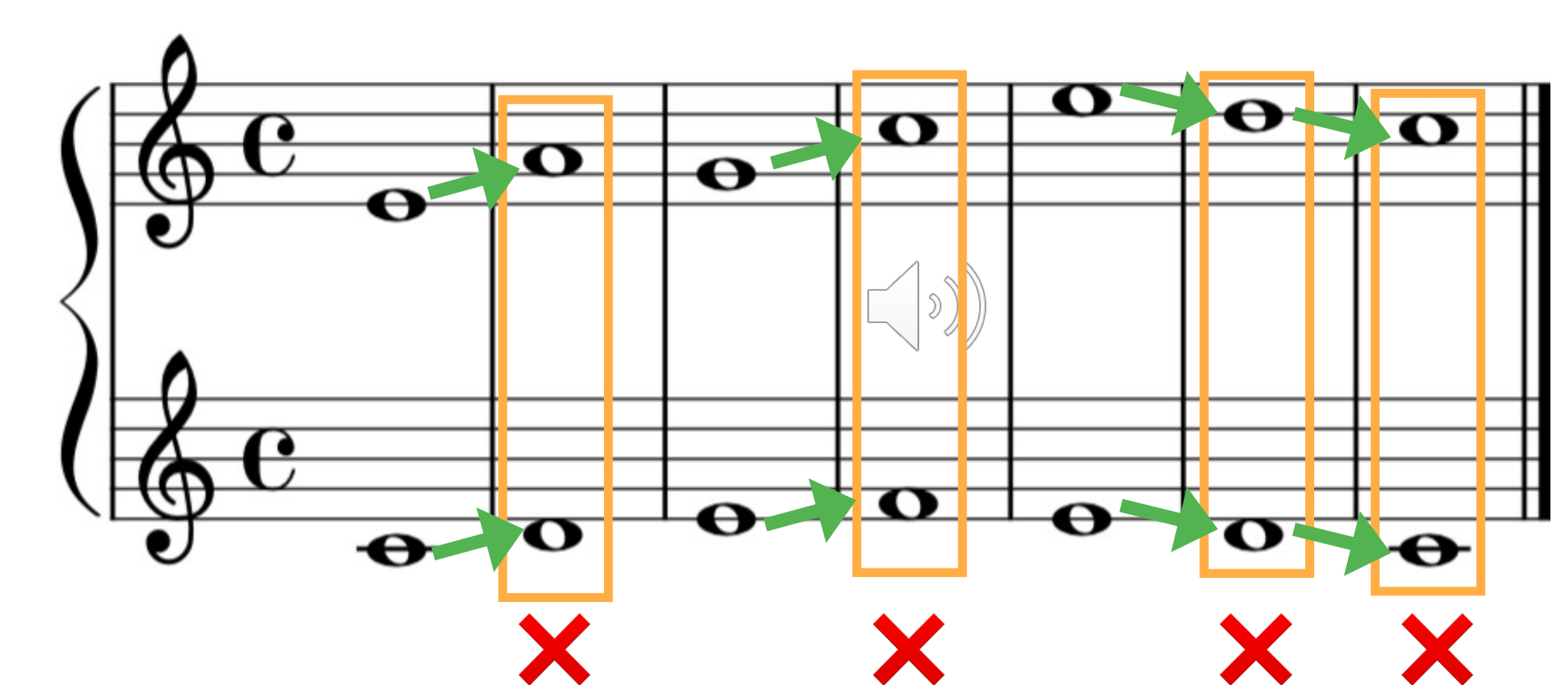


-- 動きの規則（平行5度・8度を含まない）

measure motionOK :: PIs -> Bool where

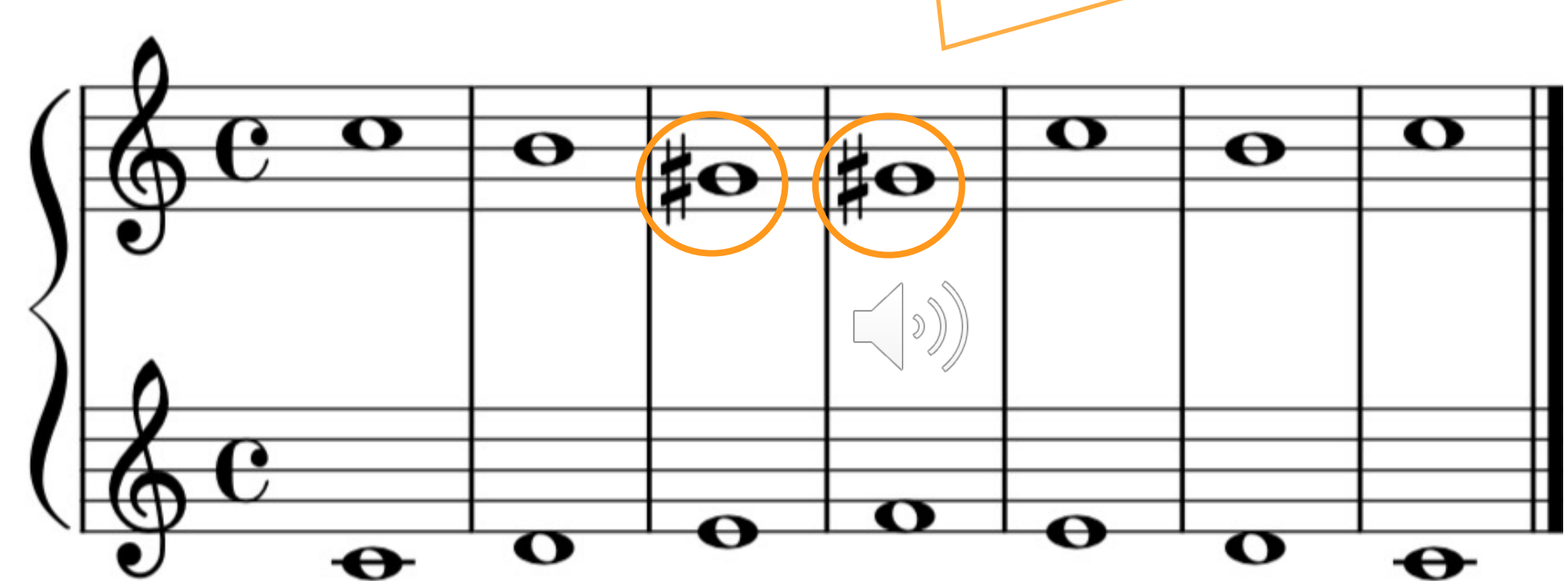
Nil -> True

Cons pi pis -> ...



SYNQUID^[3] を用いた実験

- 音階上でない
- 重複



→ 常識（必須でない規則）の考慮

提案：重み付き篩型

必須 : 重み ∞
非必須 : 重み $n \in \mathbb{N}$

各和音
の制約

隣り合う
和音の制約

type CP = List (Pitch * Interval) < r1, r2 >

where r1 = (isConsonant, ∞) \wedge (isScaleNote, 80)
r2 = (motionOK, ∞) \wedge (notRepeated, 60)

疑問：重み付き篩型はプログラミングにも有用？

[1] Szamozvancev & Gale “Well-typed Music Does Not Sound Wrong”, Haskell ’17.

[2] Fux “Gradus ad Parnassum”, 1725.

[3] Polikarpova et al. “Synthesizing Programs from Polymorphic Refinement Types”, PLDI ’16.