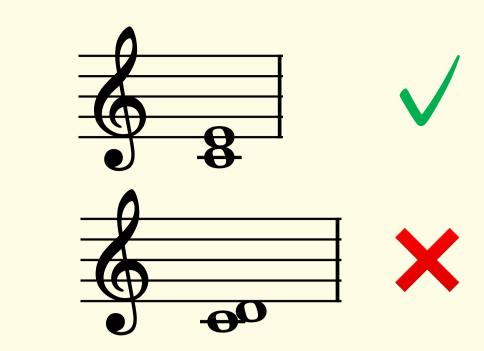
Agdaによる対位法の形式化

叢悠悠(東京工業大学)

背景:プログラミング言語における型規則 vs. 音楽理論における作曲規則

 出現する和音はすべて協和音(1,3,5,6,8度)

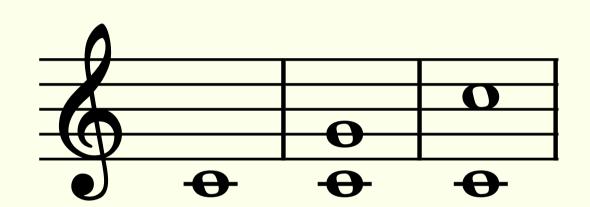


貢献:音楽理論における対位法の規則を Agda で形式化

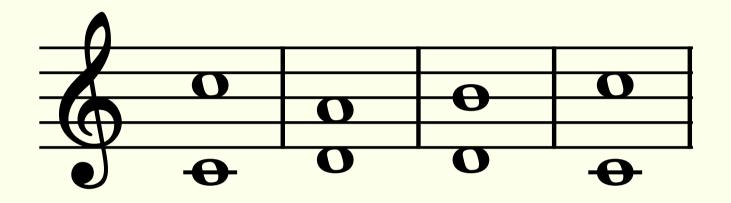
スローガン: Dependently-typed music does not sound wrong

対位法:複数の旋律を重ねる技法[1]

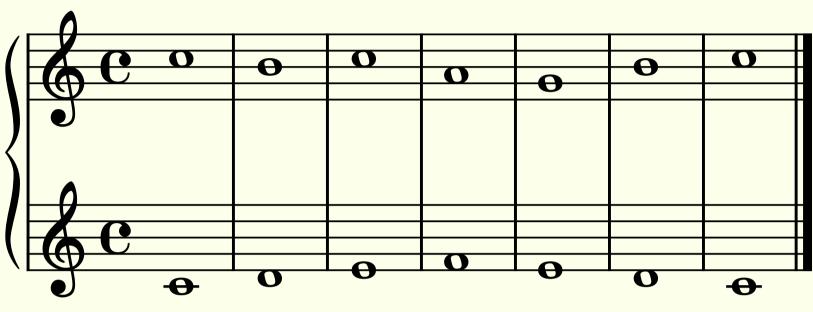
★ 規則1:開始の明確さ 最初の和音は1度,5度,or8度



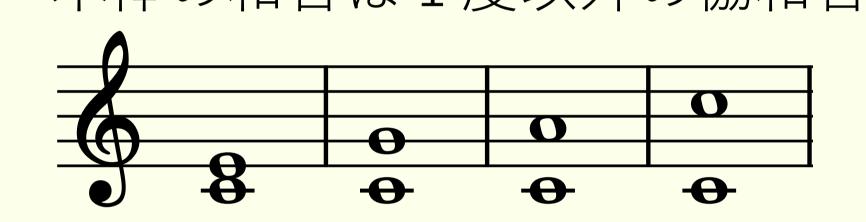
★ 規則3:進行のスムーズさ
5度と8度への移動は逆方向から



規則を守った例:



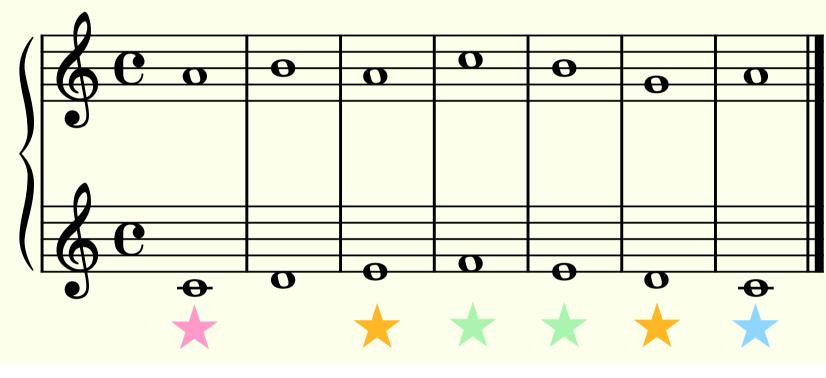
★ 規則2:和音の響きの良さ 本体の和音は1度以外の協和音



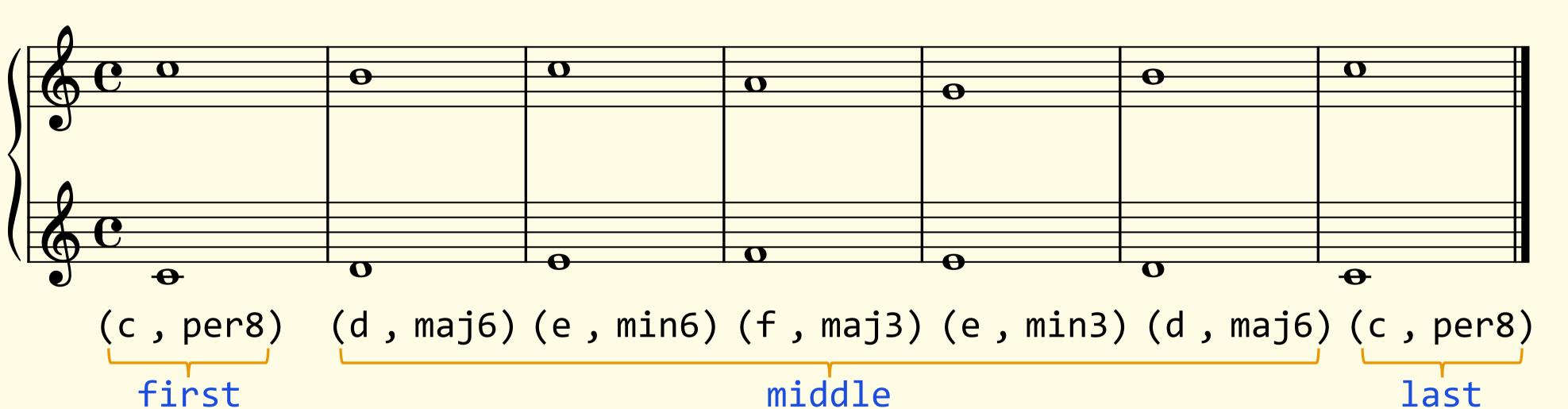
★ 規則4:終了の明確さ 最後の2和音は終止形



規則を破った例:



提案:依存レコードを用いた良い音楽の表現



和音のデータ: ベースの音程 (Pitch型) + 2音の距離 (Interval型)

record FirstSpecies : Set where constructor firstSpecies

field

firstBar : PitchInterval
middleBars : List PitchInterval
lastBar : PitchInterval

★ beginningOk : checkBeginning firstBar = nothing

fs : FirstSpecies

fs = firstSpecies first middle last refl refl refl

フィールド:和音のデータ+正しさの証明

data IntervalError : Set where

dissonant : Interval → IntervalError -- 不協和音 unison : Pitch → IntervalError -- 1度の和音

intervalCheck : PitchInterval → Maybe IntervalError
intervalCheck (p , i) with isConsonant i | isUnison i
intervalCheck (p , i) | false | _ = just (dissonant i)
intervalCheck (p , i) | true | true = just (unison p)
intervalCheck (p , i) | true | false = nothing

checkIntervals : List PitchInterval → List IntervalError
checkIntervals = mapMaybe intervalCheck

今後の課題

- ・ 芸術的な面白さの考慮 [2]
- ・規則を破った音楽の許容
- 証明付きはもりラインの自動生成

参考文献

- [1] J. J. Fux. The Study of Counterpoint. W. W. Norton & Company, 1965.
- [2] M. Komosinski and P. Szachewicz. Automatic species counterpoint by means of the dominance relation. J. Math. Music, 2015.