A Model of Musical Motifs

Torsten Anders

MCM 2007
18 May 2007

Outline

Introduction: Music Constraint Programming

Research field

- Computational modelling of composition
- Explicitly modelling compositional knowledge

Successful technique: constraint programming

- User states music theory model – computer generates music which complies with this theory
- Music theory: a set of rules (constraints) applied to unknows (variables) in solution score
- Highly declarative definition (e.g. simple combination of music theories by conjunction)
- Efficient solvers

Research Goal

Long-term objective modelling of explicit knowledge of melody composition and musical form

Present goal modelling of fundamental melodic/formal concept: musical motifs as constraint satisfaction problem

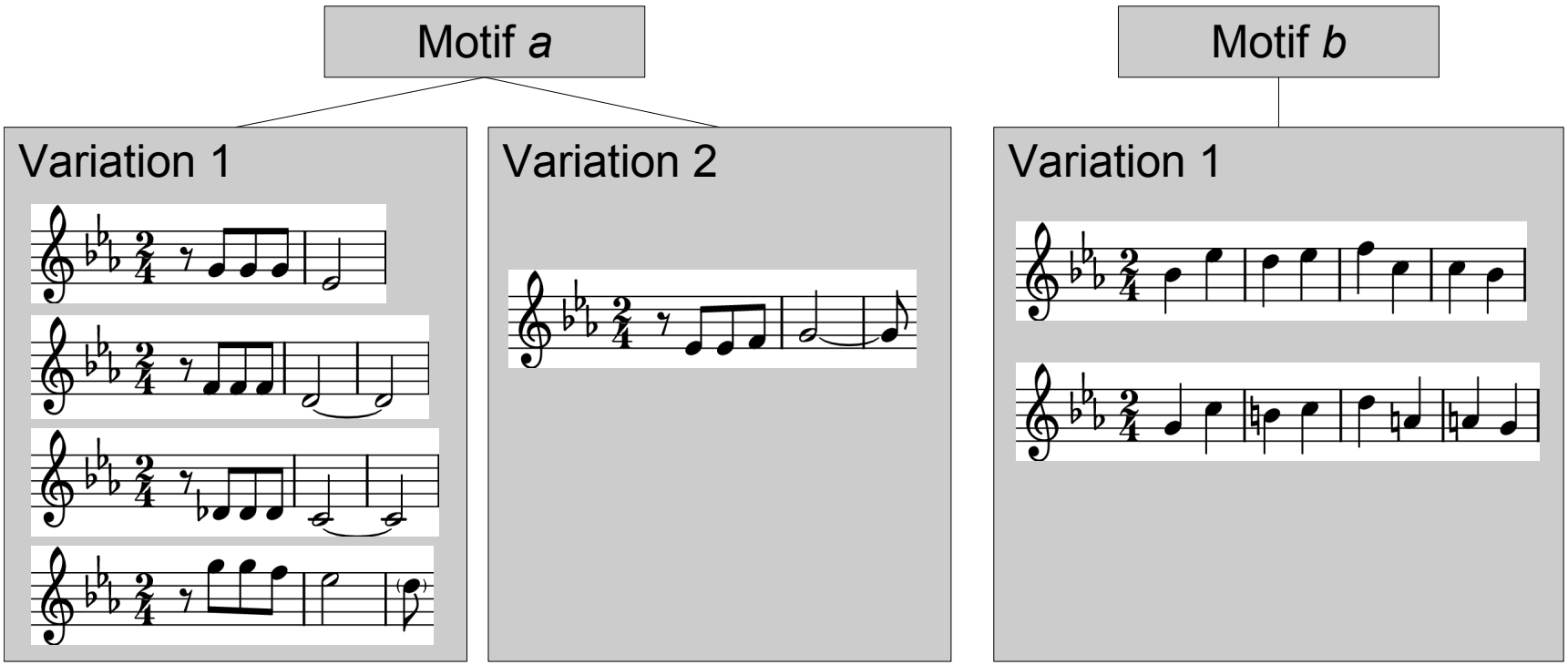
Motif Components

$$a\ motif := \langle representation, description, variation \rangle$$

A motif consists of

- Representation: stores score information
- Description: specifies motif's identity (e.g. a vs. b)
- Variation: specifies motif's variation (e.g. a^1 vs. a^2)

Example: Beethoven's Symphony No. 5



Motif Components: the Details

- Representation (contains variables): instance of some music representation
- Description (is variable): symbolic description (feature-value pairs)
- Variation (is variable):
  - function constraining relation between representation and description
  - convenient variant: feature-function pairs

Music Representation

- Music representation 'factored out'
- Requirements for the employed representation
  - Support for variables
  - Hierarchic representation: motifs are separate objects
  - Interface for accessing score information

Motif Components: the Details

- Representation (contains variables): instance of some music representation
- Description (is variable): symbolic description (feature-value pairs)
- Variation (is variable):
  - function constraining relation between representation and description
  - convenient variant: feature-function pairs

Motif Description

General form

$$a\ description := \langle feature_1 : variable\ list_1$$
$$feature_2 : variable\ list_2$$
$$\ldots \rangle$$
$$description := \bigvee description_a, description_b, \ldots$$

Beethoven example

$$description_a := \langle durations : (\text{♪}, \text{♪}, \text{♪}, (\bigvee \text{𝅗𝅥}, \ldots, \text{𝅝}))$$
$$pitch\ contour : (\rightarrow, (\bigvee \rightarrow, \searrow), \searrow)\rangle$$
$$description_b := \langle durations : (\text{♩}, \text{♩}, \text{♩}, \text{♩}, \text{♩}, \text{♩}, \text{♩}, \text{♩})$$
$$scale\ degree\ intervals : (3, -1, 1, 1, -3, 0, -1)\rangle$$

Motif Components: the Details

- Representation (contains variables): instance of some music representation
- Description (is variable): symbolic description (feature-value pairs)
- Variation (is variable):
  - function constraining relation between representation and description
  - convenient variant: feature-function pairs

Motif Variation

General Form

$$\begin{aligned} a\ variation := \langle & feature_1 : function_1 : motif \rightarrow variable\ list_1 \\ & feature_2 : function_2 : motif \rightarrow variable\ list_2 \\ & \ldots \rangle \end{aligned}$$

$$variation := \bigvee variation_1, variation_2, \ldots$$

Variable lists at matching features of description and variation are constrained to be equal.

Motif Variation (continued)

Beethoven Example

$$variation_1 := \langle durations : getNoteDurations$$
$$pitch\ contour : getPitchContour$$
$$scale\ degree\ intervals : getScaleDegreeIntervals \rangle$$
$$variation_2 := \langle durations : getNoteDurations$$
$$pitch\ contour : myPitchContourTransformation \rangle$$
$$myPitchContourTransformation(myMotif) :=$$
$$getDescription(myMotif) = description_a$$
$$\wedge\ inverse(getPitchContour(myMotif))$$

Beethoven's Symphony

Discussion

Motif model very flexible: allows for

- Score generation from a motivic analysis
- Application of various similarity formalisms (e.g. pitch contour equivalence)
- Non-motivic sections: non-motific variation function constrains nothing
- Contrapunctual combinations: motif description is searched for
- Combination with various other music theories (e.g. harmony, counterpoint)

Efficient implementation in Strasheela (implemented in Oz): uses constraint propagation and supports user-definable search strategy.

Conclusion

The motif model constrains relation between three motif properties

- Its score representation
- Its symbolic description (specifying its identity)
- Its variation: constraining relation between representation and description