Project Proposal

Advanced Functional Programming

Joris ten Tusscher, Cas van der Rest, Orestis Melkonian

1 Domain

1.1 Algorithmic Music Composition

1.2 Generation Techniques

1.3 Motivation

[1]

2 Problem

2.1 Music-Representation DSL

2.1.1 Euterpea

Euterpea[1]

2.1.2 Export to MIDI

Midi[2]

2.1.3 Render to music scores

Lilypond[3]

2.2 Generation DSL

2.2.1 Chaos Functions

[2]

2.2.2 L-Systems

[3]

[1] https://hackage.haskell.org/package/Euterpea
[2] http://hackage.haskell.org/package/midi
[3] https://hackage.haskell.org/package/lilypond

2.2.3 QuickCheck

2.3 Constraint DSL

As the solution space defined by our categorial grammar alone is huge, searching for solutions exhibiting specific desired properties (e.g. melodies involving notes from a certain scale) would be computationally infeasible.

To remedy this, we will implement a DSL that will allow the programmer to naturally express constraints, which will be respected by the musical artefacts we generate; these will model musical properties such as restricted pitch range. As you would expect, these constraints will not be applied posthumously as a filter, but integrated in the generation process, effectively pruning the search space.

2.4 Applications

Apart from the above, we also aim to implement several applications, showcasing the features of our library:

Music Representation We will provide code snippets that demonstrate one's ability to write concrete music pieces using our DSL and to export them in MIDI format or music notation.

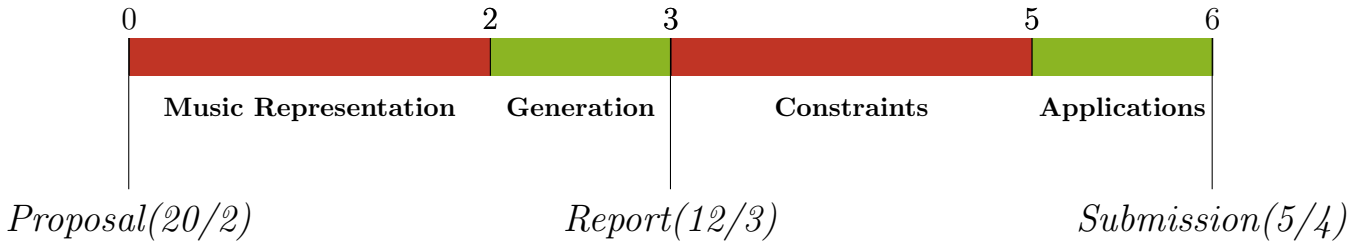
Generation We plan to implement several common generation techniques, such as creating melodies from chaotic/complex functions and structuring pieces via an L-system grammar.

Constraints We will demonstrate how our library can be used to for automatic generation of musical exercises, utilizing a variety of constraints.

An important property of our library that we wish to demonstrate with our examples, is that it is not geared specifically towards single-voice melodies, but can be used as easily to generate rhythm, harmony or anything combining these three principal elements of music. If time permits, we will also implement a simple web interface, which runs our library on the back-end and allows the user to select a number of pre-defined constraints in order to generate, for instance, musical exercises. Last but not least, the library will shipped with its own "Prelude", providing common patterns/techniques for algorithmic music composition.

3 Planning

Below we give the estimated schedule across the six weeks available:



References

[1] H. Young, "A categorial grammar for music and its use in automatic melody generation," in *Proceedings of the 5th ACM SIGPLAN International Workshop on Functional Art, Music, Modeling, and Design*, pp. 1–9, ACM, 2017.

[2] R. Bidlack, "Chaotic systems as simple (but complex) compositional algorithms," *Computer Music Journal*, vol. 16, no. 3, pp. 33–47, 1992.

[3] J. McCormack, "Grammar based music composition," *Complex systems*, vol. 96, pp. 321–336, 1996.