

Double Interpreter for Musica Programming Language

Formal Languages and Compilers

AGH University of Science and Technology
Kaung Sithu | Min Khant Soe Oke

Abstract:

This project aims to develop a double interpreter for the Musica Programming Language, which is designed specifically for algorithmic music composition. The interpreter will generate both PDF scores and sound files (WAV and MP3) based on the compositions written in Musica. The project requires thorough understanding of the Musica language syntax and semantics, creation of an ANTLR4 grammar, implementation of a Python interpreter, integration with LilyPond and other music generation tools, and error handling mechanisms.

Project Overview:

The Musica Programming Language is a simple language designed for algorithmic music composition. It allows users to create melodies, rhythms, and harmonies using a variety of instructions. The language also supports the generation of PDF scores and sound files.

This project will develop a double interpreter for Musica. The first interpreter will read Musica code and translate it into a representation that can be used by a music generation engine. The second interpreter will read the representation of the music and generate PDF scores and sound files.

Project Requirements:

- Understand the syntax and semantics of the Musica Programming Language
- Create an ANTLR4 grammar for the Musica language
- Implement a Python interpreter for the Musica language
- Integrate the Python interpreter with LilyPond for generating PDF scores
- Integrate the Python interpreter with Timidity++ or another MIDI-to-WAV converter for generating WAV files
- Implement error handling mechanisms to catch and report errors

Project Deliverables:

- A Python interpreter for the Musica Programming Language
- A Makefile or similar build system to compile and run the interpreter

- A test suite to ensure the correctness of the interpreter
- A demonstration of the interpreter using examples from the Musica documentation

Project Success Criteria:

- The interpreter should be able to correctly interpret all Musica code from the documentation
- The interpreter should generate PDF scores that match the examples in the documentation
- The interpreter should generate WAV files that accurately represent the melodies, rhythms, and harmonies of the Musica code
- The interpreter should handle errors gracefully and provide informative error messages

Project Challenges:

- The Musica Programming Language is a relatively simple language, but there are a number of edge cases that need to be handled correctly by the interpreter.
- The generation of PDF scores requires careful formatting and placement of musical elements.
- The generation of WAV files requires accurate conversion of MIDI data into audio samples.

Project Benefits:

- The double interpreter for Musica will provide a valuable tool for musicians and composers who want to create music algorithmically.
- The project will help us to develop our skills in programming and music theory.
- The project will contribute to the development of the Musica Programming Language.

Conclusion:

The development of a double interpreter for Musica is a challenging but rewarding project that will provide significant benefits to the music community. The project will also give us valuable experience in programming and music theory.