

Proposal: ABCD - ABC Music Notation extended with programming language features*

Trent Jeffery

The University of British Columbia
Vancouver, B.C.
y3f0b@ugrad.cs.ubc.ca

Victoria Wang

The University of British Columbia
Vancouver, B.C.
y7x9a@ugrad.cs.ubc.ca

Ziyang Jin

The University of British Columbia
Vancouver, B.C.
f4a0b@ugrad.cs.ubc.ca

Yifan Yang

The University of British Columbia
Vancouver, B.C.
p7w9a@ugrad.cs.ubc.ca

ABSTRACT

This is our project proposal for our new Domain Specific Language - ABCD, the extended language that adds programming features to the classic ABC notation.

KEYWORDS

Programming Language, Music, ABC Music Notation

ACM Reference Format:

Trent Jeffery, Ziyang Jin, Victoria Wang, and Yifan Yang. 2017. Proposal: ABCD - ABC Music Notation extended with programming language features. In *Proceedings of The University of British Columbia (CPSC 311 2017W1)*. ACM, New York, NY, USA, 1 page. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 HOW IT COULD BE A FULL 100% PROJECT

We intend to design a Domain-Specific Language extending the ABC music notation language. ABC is similar in power and purpose to \LaTeX : it allows a user to hand-type a document that represents a piece of sheet music, and can be compiled to a PDF or played aloud using appropriate software.

Our language, ABCD, will enhance ABC with variables, functions, conditionals, and loops, empowering users to programmatically modify existing ABC documents and generate or author new music more easily. Variables will allow snippets of music code to be built up, modified, and used multiple times. Functions will provide a way to transform or generate music code in a way that is reusable over different pieces of music code. Conditionals and loops provide control flow mechanisms that make it much easier to apply changes selectively or repeatedly.

ABCD will be used to write scripts that compile into ABC files after modification or creation. By compiling to ABC files, our users can leverage existing software that works with ABC, including music players and pdf compilers. ABCD should have native support

for importing ABC files, allowing users to easily modify existing ABC scores. To support the use of "library files" containing useful definitions, the language should also support importing other ABCD files.

An ABCD file may contain definitions of variables and functions as well as ABC notation. When compiled, functions will modify, add, or remove ABC code in the script, and the script will evaluate to one or more ABC documents.

A full implementation of the language would involve:

- (1) An EBNF defining concrete syntax for:
 - Variables
 - Functions
 - Loops
 - Conditionals
 - Import statements
 - ABCD data types (especially types that are not in ABC)
 - Any other syntax constructs required to support these features
- (2) An ABCD parser
- (3) An abstract syntax implementation
- (4) A compiler to produce ABC files from the ABCD abstract syntax

We would likely write our parser and compiler in Python to harness its string processing capabilities, as ABC has a plain-text format where characters often need to be considered individually. For a final project, we would likely also write a small ABCD program to demonstrate a possible use case for ABCD and showcase some of its capabilities. This would probably be a program that applies some transposition, addition of harmonies to certain notes, and some rhythm changes, and we would demonstrate by applying the program to a small ABC tune and playing the result or printing it as a pdf.

*The authors' names are ordered by last name.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CPSC 311 2017W1, 2017, Vancouver, BC, Canada

© 2017 Copyright held by the owner/author(s).

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>