Judah Daniels

Inferring Harmony from Free Polyphony

Computer Science Tripos – Part II

Clare College

July, 2023

Declaration of originality

I, Judah Daniels of Clare College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose. I am content for my dissertation to be made available to the students and staff of the University.

Signed Judah Daniels

Date January 21, 2023

Proforma

Candidate Number:	**2200D**
Project Title:	**Inferring Harmony from Free Polyphony**
Examination:	**Computer Science Tripos – Part II, July, 2023**
Word Count:	**1587**[1]
Code Line Count:	**1000**
Project Originator:	**Christoph Finkensiep**
Supervisor:	**Dr Peter Harrison**

Original Aims of the Project

Work Completed

All that has been completed appears in this dissertation.

Special Difficulties

None

[1] This word count was computed by `detex diss.tex | tr -cd '0-9A-Za-z \n' | wc -w`

Contents

List of Figures

Acknowledgements

This document owes much to an earlier version written by Simon Moore [?]. His help, encouragement and advice was greatly appreciated.

Chapter 1

Introduction

1.1 Motivation

The introduction should explain the principal motivation for the project and show how the work fits into the broad area of surrounding computer science and give a brief survey of previous related work.
Clear motivation, justifying potential benefits of success.

1.2 Related Work

It should generally be unnecessary to quote at length from technical papers or textbooks. If a simple bibliographic reference is insufficient, consign any lengthy quotation to an appendix.

1.3 Aims

Chapter 2

Preparation

Description: Principally, this chapter should describe the work which was undertaken before code was written, hardware built or theories worked on. It should show how the project proposal was further refined and clarified, so that the implementation stage could go smoothly rather than by trial and error.

Throughout this chapter and indeed the whole dissertation, it is essential to demonstrate that a proper professional approach was employed.

2.1 Starting Point

It is essential to declare the starting point. This states any existing codebase or materials that your project builds on. The text here can commonly be identical to the text in your proposal, but it may enlarge on it or report variations. For instance, the true starting point may have turned out to be different from that declared in the proposal and such discrepancies must be explained.

2.1.1 Relevant courses and experience

2.1.2 Existing codebase

2.2 The Protovoice Model

Mention complicated theories or algorithms which required understanding.
Clear presentation of challenging background material covering a range of computer science topics beyond Part IB.

2.3 Heuristic Search Algorithms

2.4 Data Processing

2.5 Requirements Analysis

Underlining the professional approach, this chapter will very likely include a section headed ”Requirements Analysis” and refer to appropriate software engineering techniques used in the dissertation.
Good or excellent requirements analysis;

2.5.1 Main deliverables

Main deliverables with a risk analysis.

2.5.2 Dependency Analysis

Implementation modules with a dependency analysis.

2.6 Software Engineering Techniques

Justified and documented selection of suitable tools; good engineering approach.

2.6.1 Development model

Include Gantt chart.

2.6.2 Languages, libraries and tools

The chapter will also cite any new programming languages and systems which had to be learnt

Table 2.1: Languages, libraries and tools

Tool	Purpose	License
Haskell	Main language	...
GHC	Compiling and profiling to inspect time performance and memory usage	GPL-3.0+
Haskell-Musicology	...	...
Python	...	...
Numpy	...	...
Pandas	...	...
MS3	...	...
Musescore 3	...	...
Protovoice Annotation Tool	...	...
Git	Version Control, Continuous Integration	...

Chapter 3

Implementation

Description: This chapter should describe what was actually produced: the programs which were written, the hardware which was built or the theory which was developed. Any design strategies that looked ahead to the testing stage should be described in order to demonstrate a professional approach was taken.

Descriptions of programs may include fragments of high-level code but large chunks of code are usually best left to appendices or omitted altogether. Analogous advice applies to circuit diagrams or detailed steps in a machine-checked proof.

The implementation chapter should include a section labelled ”Repository Overview”. The repository overview should be around one page in length and should describe the high-level structure of the source code found in your source code repository. It should describe whether the code was written from scratch or if it built on an existing project or tutorial. Making effective use of powerful tools and pre-existing code is often laudable, and will count to your credit if properly reported. Nevertheless, as in the rest of the dissertation, it is essential to draw attention to the parts of the work which are not your own.

It should not be necessary to give a day-by-day account of the progress of the work but major milestones may sometimes be highlighted with advantage.

Mark Scheme: Contribution to the field. Application of extra-curricular reading and original interpretation of previous work from academia or industry. Challenging goals and substantial deliverables with excellent selection and application of appropriate mathematical, scientific and/or engineering techniques. Clear and justified repository overview. At most minor faults in execution or understanding.

3.1 Repository Overview:

The following describes the protovoices-haskell repository, and where my code contribution will lie:

```
protovoices-haskell
 └─ app
     └─ MainHeuristicSearch.hs <- My code
```

```
│
│   │
│   └── ...
├── src
│   ├── Heuristics <- My code
│   │   └── ...  <- My code
│   └── ...
├── test
├── testdata
└── ...
```

Chapter 4

Evaluation

Description: This is where Assessors will be looking for signs of success and for evidence of thorough and systematic evaluation. Sample output, tables of timings and photographs of workstation screens, oscilloscope traces or circuit boards may be included. Care should be employed to take a professional approach throughout. For example, a graph that does not indicate confidence intervals will generally leave a professional scientist with a negative impression. As with code, voluminous examples of sample output are usually best left to appendices or omitted altogether.

There are some obvious questions which this chapter will address. How many of the original goals were achieved? Were they proved to have been achieved? Did the program, hardware, or theory really work?

Assessors are well aware that large programs will very likely include some residual bugs. It should always be possible to demonstrate that a program works in simple cases and it is instructive to demonstrate how close it is to working in a really ambitious case.

Mark Scheme (Evaluation and Conclusions): Conclusions provide an effective summary of work completed along with good future work. Clearly presented argument demonstrating success criteria met. Good or excellent evidence of critical thought and interpretation of the results which substantiate any claims of success, improvements or novelty.

My thoughts No terminal output, just diagrams and tables?

Chapter 5

Conclusions

This chapter is likely to be very short and it may well refer back to the Introduction.

5.1 Achievements

5.2 Future Work

5.3 Lessons learned

Personal reflection on the lessons learned. It might offer a reflection on the lessons learned and explain how you would have planned the project if starting again with the benefit of hindsight.

Appendix A

Latex source

A.1 metadata.tex

```
% Change these

\newcommand{\mcandidate}{2200D}
\newcommand{\mfullname}{Judah Daniels}
\newcommand{\mcollege}{Clare College}
\newcommand{\mtitle}{Inferring Harmony from Free Polyphony}
\newcommand{\mexamination}{Computer Science Tripos -- Part II}
\newcommand{\mdate}{July, 2023}
\newcommand{\moriginator}{Christoph Finkensiep}
\newcommand{\msupervisor}{Dr Peter Harrison}
\newcommand{\mwordcount}{1587}
\newcommand{\mlinecount}{1000}
% Consent to the dissertation made available to University members
\newcommand{\mconsent}{I am content for my dissertation to be made available to the students and staff of the University.}
% For the Declaration of originality
\newcommand{\msignature}{Judah Daniels}
```

A.2 main.tex

```
% Template for a Computer Science Tripos Part II project dissertation
\documentclass[12pt,a4paper,twoside,openright]{report}
\usepackage[pdfborder={0 0 0}]{hyperref}    % turns references into hyperlinks
\usepackage[margin=25mm]{geometry}  % adjusts page layout
\usepackage{graphicx}  % allows inclusion of PDF, PNG and JPG images
\usepackage{verbatim}
\usepackage{docmute}   % only needed to allow inclusion of proposal.tex
\usepackage[utf8]{inputenc}
\usepackage{mathtools}
\usepackage{changepage}
\usepackage{url}
\usepackage{blindtext}
\usepackage{tabularx,booktabs}
\usepackage{dirtree}
\usepackage{cite}
\usepackage{float} % Prevent Latex from repositioning tables
\usepackage{amsmath}
\usepackage{graphicx}

%-------------------------------------------------------------------------------------------

% Formatting Commands
\newcommand{\keyword}[1]{\textbf{#1}}
\newcommand{\tabhead}[1]{\textbf{#1}}
```

```
\newcommand{\code}[1]{\texttt{#1}}
\newcommand{\file}[1]{\texttt{\bfseries#1}}
\newcommand{\option}[1]{\texttt{\itshape#1}}

%----------------------------------------------------------------------------------------

% Language setting
\usepackage[english]{babel}
%\raggedbottom                           % try to avoid widows and orphans
\sloppy
\clubpenalty1000%
\widowpenalty1000%

\renewcommand{\baselinestretch}{1.1}    % adjust line spacing to make
                                        % more readable

\begin{document}

\include{metadata}

\bibliographystyle{plain}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\thispagestyle{empty}

\rightline{\LARGE \textbf{\mfullname}}

\vspace*{60mm}
\begin{center}
\Huge
\textbf{\mtitle} \\[5mm]
\mexamination \\[5mm]
\mcollege \\[5mm]
\mdate  % today's date
\end{center}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proforma, table of contents and list of figures
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\pagestyle{plain}

\newpage
\newpage
\section*{Declaration of originality}

I, \mfullname{} of \mcollege, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissert

\bigskip
\leftline{Signed \msignature}
\bigskip
\leftline{Date \today}

\chapter*{Proforma}

{\large
\begin{tabular}{ll}
Candidate Number:   & \bf \mcandidate                    \\
Project Title:      & \bf \mtitle                        \\
Examination:        & \bf \mexamination, \mdate          \\
Word Count:         & \bf \mwordcount\footnotemark[1]    \\
Code Line Count:    & \bf \mlinecount                    \\
Project Originator: & \bf \moriginator                   \\
Supervisor:         & \bf \msupervisor                   \\
\end{tabular}
}

\footnotetext[1]{This word count was computed
by \texttt{detex diss.tex | tr -cd ’0-9A-Za-z $\tt\backslash$n’ | wc -w}
}
\stepcounter{footnote}


\section*{Original Aims of the Project}
% At most 100 words



\section*{Work Completed}
% At most 100 words

All that has been completed appears in this dissertation.

\section*{Special Difficulties}
% At most 100 words

None

\newpage

\tableofcontents

\listoffigures

\newpage
\section*{Acknowledgements}

This document owes much to an earlier version written by Simon Moore
\cite{Moore95}.  His help, encouragement and advice was greatly
appreciated.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% now for the chapters

\pagestyle{headings}

\chapter{Introduction}

\section{Motivation}
The introduction should explain the principal motivation for the project and show how the work fits into the broad area of s
Clear motivation, justifying potential benefits of success.
\section{Related Work}
It should generally be unnecessary to quote at length from technical papers or textbooks.
If a simple bibliographic reference is insufficient, consign any lengthy quotation to an appendix.
\section{Aims}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Preparation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\chapter{Preparation}
\paragraph{Description: }
Principally, this chapter should describe the work which was undertaken before code was written, hardware built or theories
It should show how the project proposal was further refined and clarified, so that the implementation stage could go smoothl
\par
Throughout this chapter and indeed the whole dissertation, it is essential to demonstrate that a proper professional approac
\par

\section{Starting Point}
It is essential to declare the starting point.
This states any existing codebase or materials that your project builds on.
```

```
The text here can commonly be identical to the text in your proposal, but it may enlarge on it or report variations.
For instance, the true starting point may have turned out to be different from that declared in the proposal and such discre

\subsection{Relevant courses and experience}

\subsection{Existing codebase}

\section{The Protovoice Model}
Mention complicated theories or algorithms which required understanding. \\
Clear presentation of challenging background material covering a range of computer science topics beyond Part IB.

\section{Heuristic Search Algorithms}

\section{Data Processing}

\section{Requirements Analysis}
Underlining the professional approach, this chapter will very likely include a section headed "Requirements Analysis" and re
Good or excellent requirements analysis;
\par
\subsection{Main deliverables}
Main deliverables with a risk analysis.
\subsection{Dependency Analysis}
Implementation modules with a dependency analysis.

\section{Software Engineering Techniques}
Justified and documented selection of suitable tools; good engineering approach.

\subsection{Development model}
Include Gantt chart.

\subsection{Languages, libraries and tools}
The chapter will also cite any new programming languages and systems which had to be learnt

\begin{table}
  {
  \small
  \caption{Languages, libraries and tools}
  \label{Languages}
  \begin{center}
    \begin{tabularx}{.9\textwidth}{cXc}
      Tool & Purpose & License \\
      \toprule
      Haskell & Main language & ... \\
      \midrule
      GHC & Compiling and profiling to inspect time performance and memory usage  & GPL-3.0+ \\
      \midrule
      Haskell-Musicology & ... & ... \\
      \midrule
      Python & ... & ... \\
      \midrule
      Numpy & ... & ... \\
      \midrule
      Pandas & ... & ... \\
      \midrule
      MS3 & ... & ... \\
      \midrule
      Musescore 3 & ... & ... \\
      \midrule
      Protovoice Annotation Tool & ... & ... \\
      \midrule
      Git & Version Control, Continuous Integration & ... \\
      \bottomrule
    \end{tabularx}
  \end{center}
  }
\end{table}
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Implementation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\chapter{Implementation}

\paragraph{Description: }
This chapter should describe what was actually produced: the programs which were written, the hardware which was built or th
\par
Descriptions of programs may include fragments of high-level code but large chunks of code are usually best left to appendic
\par
The implementation chapter should include a section labelled "Repository Overview". The repository overview should be around
\par
It should not be necessary to give a day-by-day account of the progress of the work but major milestones may sometimes be hi

\paragraph{Mark Scheme: }
Contribution to the field.
Application of extra-curricular reading and original interpretation of previous work from academia or industry.
Challenging goals and substantial deliverables with excellent selection and application of appropriate mathematical, scienti
Clear and justified repository overview.
At most minor faults in execution or understanding.

\section{Repository Overview:}
The following describes the protovoices-haskell repository, and where my code contribution will lie:
\par
\medskip
\dirtree{%
.1 protovoices-haskell.
.2 app.
.3 MainHeuristicSearch.hs <- My code.
.3 ....
.2 src.
.3 \textbf{Heuristics} <- My code.
.4 ...  <- My code.
.3 ....
.2 test.
.2 testdata.
.2 ....
}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Evaluation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\chapter{Evaluation}

\paragraph{Description:}
This is where Assessors will be looking for signs of success and for evidence of thorough and systematic evaluation.
Sample output, tables of timings and photographs of workstation screens, oscilloscope traces or circuit boards may be includ
Care should be employed to take a professional approach throughout.
For example, a graph that does not indicate confidence intervals will generally leave a professional scientist with a negati
As with code, voluminous examples of sample output are usually best left to appendices or omitted altogether.
\par
There are some obvious questions which this chapter will address. How many of the original goals were achieved? Were they pr
\par
Assessors are well aware that large programs will very likely include some residual bugs. It should always be possible to de

\paragraph{Mark Scheme (Evaluation and Conclusions): }
Conclusions provide an effective summary of work completed along with good future work.
Clearly presented argument demonstrating success criteria met.
Good or excellent evidence of critical thought and interpretation of the results which substantiate any claims of success, i

\paragraph{My thoughts}
No terminal output, just diagrams and tables?

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Conclusions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
\chapter{Conclusions}
This chapter is likely to be very short and it may well refer back to the Introduction.
\section{Achievements}

\section{Future Work}

\section{Lessons learned}
Personal reflection on the lessons learned.
It might offer a reflection on the lessons learned and explain how you would have planned the project if starting again with

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the bibliography
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\addcontentsline{toc}{chapter}{Bibliography}
\bibliography{refs}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the appendices
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\appendix

\chapter{Latex source}

\section{metadata.tex}
{\scriptsize\verbatiminput{metadata.tex}}

\section{main.tex}
{\scriptsize\verbatiminput{main.tex}}

\section{proposal.tex}
{\scriptsize\verbatiminput{proposal.tex}}

\chapter{Makefile}

\section{makefile}\label{makefile}
{\scriptsize\verbatiminput{makefile.txt}}

\section{refs.bib}
{\scriptsize\verbatiminput{refs.bib}}


\chapter{Project Proposal}

\input{proposal}

\end{document}
```

A.3 proposal.tex

```
\documentclass{article}

% Packages
\usepackage{tabularx,booktabs}
\usepackage{dirtree}
\usepackage{cite}
\usepackage{float} % Prevent Latex from repositioning tables
\usepackage{amsmath}
\usepackage{graphicx}
\usepackage[colorlinks=true, allcolors=blue]{hyperref}

%----------------------------------------------------------------------------------------
```

```
% Formatting Commands
\newcommand{\keyword}[1]{\textbf{#1}}
\newcommand{\tabhead}[1]{\textbf{#1}}
\newcommand{\code}[1]{\texttt{#1}}
\newcommand{\file}[1]{\texttt{\bfseries#1}}
\newcommand{\option}[1]{\texttt{\itshape#1}}

%-------------------------------------------------------------------------------------

% Language setting
\usepackage[english]{babel}

% Set page size and margins
% Replace `letterpaper' with `a4paper' for UK/EU standard size
\usepackage[a4paper,top=2cm,bottom=2cm,left=3cm,right=3cm,marginparwidth=1.75cm]{geometry}

\title{Judah Daniels : Inferring Harmony from Free Polyphony}
\author{}
\date{\parbox{\linewidth}{\centering%
  \today\endgraf\bigskip
  DOS: Prof. Larry Paulson\endgraf\medskip
  Crsid: jasd6 \endgraf
  College: Clare College}}
\begin{document}
\maketitle


\section{Abstract}
% Further Explanation of the background and the objectives.
A piece of music can be described using a sequence of chords, representing a higher level harmonic structure of a piece. The
\par
The paper \textit{Modeling and Inferring Proto-voice Structure in Free Polyphony} describes a generative model that encodes
%A reduction is represented by a sequence of slices, where each slice is a set of pitches with a pointer to a starting and e
\par
Christoph Finkensiep suggests in his paper that the proto-voice model may be an effective way to infer higher level latent e
  \item Accuracy: can the model successfully emulate how experts annotate harmonic progressions in musical passages?
  \item Practicality: can the model be used to do this within a reasonable time frame?
\end{itemize}
% Harmonic annotations segment a piece, marking each segment with a chord label. The goal is then to obtain these labels fro
While the original model could in theory be used to generate harmonic annotations, its exhaustive search strategy would be p
% My project will extend the use of this model into field of harmonic analysis by inferring harmonic annotations for a piece
% \par
% As the protovoice model is generative, this harmonic analysis will consist of applying the inverse of the generative opera
% The starting point is a parser provided with the paper which exhaustively searches for a solution, leading to an exponenti
% the goal is basically
% just to find a reduction of each segment to a single slice, where
% ideally (a) this slice reflects the chord label and (b) the reduction
% itself is a "plausible" derivation (

%The general approach would be based on (heuristic) search. The search
% space would be partial reductions, the starting point is the (unreduced
% and segmented) surface, and the goal would be any reduction of the
% segments to single slices. The connections between different states are
% the reduction steps defined by the PV model.

% My project will involve development of a parser(s) that can parse pieces of music using the grammar described in the paper

% The dissertation includes code for a random parser which generates a derivation of the piece according to the protovoice m
%
% The specific problem of inferring harmonic annotations is not addressed in the dissertation, so theres is an open question
%

\section{Substance and Structure}
% Give details of specific goals to be achieved
% Give precise characterisations of the methods that will be used
% Data structures and algorithms
% Key concepts, major work items, their relations and relative importance
% Specify what it means for the project to be a success.
% Identify the main sub tasks
% Outline the main algorithms or techniques to be adopted in completed them
\subsection{Core: Search}
The core of this project is essentially a search problem characterised as follows:
\begin{itemize}
  \item The state space $S$ is the set of all possible partial reductions of a piece along with each reduction step that has
  \item We have an initial state $s_o \in S$, which is the empty reduction, corresponding to the unreduced surface of the pi
  \item We have a set of actions, $A$ modelled by a function $action: A \times S \to S$. These actions correspond to a singl
    \begin{itemize}
      \item The reduction steps are the inverses of the operations defined by the generative proto-voice model.
    \end{itemize}
  \item Finally we have a goal test, $goal: S \to \{true,false\}$ which is true iff the partial reduction $s$ has exactly on
    \begin {itemize}
  \item This means the partial reduction $s$ contains a sequence of slices which start and end positions corresponding to th
    \end {itemize}
  \item At the first stage, this will be implemented using a random graph search algorithm, picking each action randomly, ac
\end{itemize}
\par

\subsection{Core: Evaluation}

The second core task is to create an evaluation module that iterates over the test dataset, and evaluates the partial reduct
\par
In order to do this I will make use of the statistical harmony model from Finkensiep's thesis, \textit{The Structure of Free
\par

% from above by measuring the likelihood that the chord label produced the slice notes.

\subsection{Extension}
Once the base search implementation and evaluation module have been completed, the search problem will be tackled by heurist
\begin{itemize}
  \item How the chord types relate to the pitches used.
  \item How the chord types relate which notes are used as ornamentation, and the degree of ornamentation.
  \item Contextual information about neighboring slices
\end{itemize}

\subsection{Overview}
\noindent
The main work packages are as follows:
\par
\medskip
\keyword{Preliminary Reading} -- Familiarise myself with the proto-voice model, and read up on similar models and their impl
\par
\medskip
\keyword{Dataset Preparation} -- Pre-process the Annotated Beethoven Corpus into a suitable representation for my algorithm.
\par
\medskip
\keyword{Basic Search} -- Implement a basic random search algorithm that takes in surface and segmentations, and outputting
\par
\medskip
\keyword{Evaluation Module} -- Implement an evaluation module to evaluate the output from the search algorithm.
% Chord Profiles for chord tones and ornaments
% Existing set of probabilities: refer to the thesis.
\par
\medskip
\keyword{End-to-end pipeline} -- Implement a full pipeline from the data to the evaluation that can be used to compare diffe
\par
\medskip
\keyword{Heuristic Design} -- Extension -- Trial different heuristics and evaluate their performance against each other.
\par
\medskip
\keyword{Dissertation} -- I intend to work on the dissertation throughout the duration of the project. I will then focus on

\section{Starting Point}
% Record any signifcant bodies of coe that will form a basis for your project that already exists
% Describe state of existing software
```

```
The following describes existing code and languages that will be used for this project:
\par
\bigskip
\keyword{Haskell} -- I will be using Haskell for this project as it is used in the proto-voice implementation. It must be no
\par
\bigskip
\keyword{Python} -- Python will be used for data handling. I have experience coding in Python.
\par
\bigskip
\keyword{Prior Research} - Over the summer I have been reading the literature on computational models of music, as well as v
\par
\bigskip
\keyword{Protovoices-Haskell} -- The paper \textit{Modeling and Inferring Proto-Voice Structure in Free Polyphony}\cite{fink
This repository includes as parsing module which will be used to perform the actions in the search space of partial reductio
\par
\bigskip
\keyword{MS3} -- This is a library for parsing MuseScore Files and manipulating labels\cite{johannes_ms3_2021}, which I will
\par
\bigskip
\keyword{ABC} -- The \textit{Annotated Beethoven Corpus}\cite{neuwirth_annotated_2018} contains analyses of all Beethoven st
\par
%
% The following describes the protovoices-haskell repository, and where my code contribution will lie:
% \par
% \medskip
% \dirtree{%
% .1 protovoices-haskell.
% .2 app.
% .3 MainExamples.hs.
% .3 MainISMIR.hs.
% .3 MainLearning.hs.
% .3 MainHeuristicSearch.hs <- My code.
% .3 ....
% .2 src.
% .3 \textbf{Heuristics} <- My code.
% .4 ...  <- My code.
% .3 ....
% .2 test.
% .2 testdata.
% .2 ....
% }
\section{Success Criteria}
% Give critia that can be used to test if you've achieved goals, and explain the form evidence will be included

This project will be deemed a success if I complete the following tasks:
\begin{itemize}
  \item Develop a baseline search algorithm that uses the proto-voice model to output a partial reduction of a piece of musi
  \item Create an evaluation module that can take the output of the search algorithm and quantitatively evaluate its accurac
    % \\ To make this concrete, an expert will manually create partial reductions, some that represent infeasible interpreta
  \item Extension: Develop one or more search algorithms that use additional heuristics to inform the search, and compare th
\end{itemize}

\section{Timetable}
\setlength{\extrarowheight}{.4em}
\begin{tabularx}{\textwidth}{@{}l  p{180pt} p{110pt} @{}}
  \toprule
  Time frame       & Work & Evidence   \\
  \midrule
  \textbf{Michaelmas} (Oct 4 to Dec 2)  & &    \\
  Oct 14 to Oct 24 & \textit{Oct 14}: Final proposal deadline.
                     Preparation work: familiarise myself with the dataset and the proto-voice model implementation.
                     Work on manipulating reductions using the proto-voice parser provided by the paper. & None       \\
  Oct 24 to Nov 7 &  Dataset preparation and handling. & Plot useful metrics about the dataset using Haskell        \\
  Nov 7 to Nov 21 &  Random Search implementation &   None     \\
  Nov 21 to Dec 5 &  Evaluation Module. Continue with search implementation. & Evaluate a manually created derivation and pl
  \midrule
  \textbf{Vacation} (Dec 3 to Jan 16)  & &    \\
  Dec 5 to Dec 11 &  Evaluate performance of random search. Begin to work on extensions & Plot results       \\
  Dec 10 to Dec 21 &  Trial different heuristics. Implement an end-to-end pipeline from input to evaluation. & None    \\
  Dec 21 to Dec 27 &  None & None    \\
  Dec 27 to Jan 10 &  Continue trialing and evaluating heuristics & \textit{Fulfill success criterion: At least one heuristi
  \midrule
  \textbf{Lent} (Jan 17 to  Mar 17)   &  &    \\
  Jan 4 to Jan 20 &  Buffer Period to help keep on track& None       \\
  Jan 20 to Feb 3
                  &
                \textit{Feb 3}: Progress Report Deadline.
                  Write progress report and prepare presentation.
                  Write draft \textit{Evaluation} chapter  & Progress Report (approx. 1 page)      \\

                  Feb 3 to Feb 17 & Prepare presentation. & \textit{Feb 8 -- 15}: Progress Report presentation       \\
  Feb 17 to Mar 3 & \textit{Feb 17}: How to write a Dissertation briefing. Write draft Introduction and Preparation chapters
  Mar 3 to Mar 17 & Write draft Implementation chapters. Incorporate feedback on Introduction and Preparation chapters.& Sen
  \midrule
  \textbf{Vacation} (Mar 18 to Apr 24)  & &    \\
  Mar 18 to Mar 31 & Complete draft dissertation. & Send draft dissertation to supervisor       \\
  Mar 31 to April 15 & Give time for supervisor to read dissertation & None \\
  April 15 to April 25 & Incorporate feedback, make final revisions and checks of the whole dissertation & Submit dissertati
  \midrule
  \textbf{Easter} (Apr 25 to Jun 16)  &  &    \\
   April 25 to May 12 & None      & None             \\
  \bottomrule
\end{tabularx}
% \section{Evaluation}
% Given the goal is to infer the harmonic annotations given a piece, the main metric by which this project will be evaluated
% This metric will be calculated using perceptually-informed chord label evaluation, comparing the inferred annotations with

\section{Resources}
I plan to use my own laptop for development: MacBook Pro 16-inch, M1 Max, 32GB Ram, 1TB SSD, 24-core GPU.

All code will be stored on a GitHub repository, which will guarantee protection from data loss. I will easily be able to swi

The project will be built upon work that has been done in the DCML (Digital cognitive musicology lab) based in EPFL. The fil

\section{Supervisor Information}
Peter Harrison, head of Centre for Music and Science at Cambridge, has agreed to supervise me for this.
We have agreed on a timetable for supervisions for this year. I am also working with Christoph Finkensiep, a PHD student at
Professor Larry Paulson has agreed to be the representative university teaching officer.

\bibliography{lib}{}
\bibliographystyle{plain}

\end{document}

% Work Packages:
% \begin{enumerate}
%   \item (14 Oct - 24 Oct) Dataset preparation
%     \begin{itemize}
%       \item Determine a representation of segments and annotations that can be used by the system and evaluation module.
%     \end{itemize}
%   \item (24 Oct - 5 Dec) Initial search pipeline
%     \begin{itemize}
%       \item Implement a basic graph search algorithm that can reduce a piece up to a single slice for each given segment
%         \begin{itemize}
%           \item The first iteration of this will reduce each segment independently based on just local information.
%         \end{itemize}
%     \end{itemize}
%   \item (5 Dec - 21 Dec) Evaluation module
%     \begin{itemize}
%       \item Iterate over the 'ground truth' dataset(contains the score and chord labels), and evaluates how well the label
%     \end{itemize}
%   \item (21 Dec - 4 Jan) Progress report
%   \item (4 Jan - 25 Jan) Improved heuristic search approach
%     \begin{itemize}
%       \item Incorporate contextual information and heuristics to create an improved system to infer harmonic annotations
%       \item Ideally I will have 2-3 different systems by this point that can be evaluated against each other, each with in
%     \end{itemize}
%   \item (25 Jan - 1 Feb) Evaluation
%   \item (1 Feb - 12 March) Dissertation write up (6 weeks)
% \end{enumerate}
```

Appendix B

Makefile

B.1 makefile

```
# This is the makefile for the Part II demonstration dissertation
#
# Note that continuation lines require ’\’ and
# that a TAB character preceeds any shell command line

.DELETE_ON_ERROR:

# Rules for building LaTeX documents (see Unix Tools course)
%.pdf %.aux %.idx: %.tex
pdflatex -halt-on-error $<
while grep ’Rerun to get ’ $*.log ; do pdflatex $< ; done
%.ind: %.idx
makeindex $*
%.bbl: %.aux
bibtex $*
# Rules for exporting xfig diagrams into PDF or EPS
%.pdf: %.eps
epstopdf --outfile=$@ $<
%.eps: %.fig
fig2dev -L eps $< $@
%.pdftex %.pdftex_t: %.fig
fig2dev -L pdftex_t -p $*.pdftex $< $*.pdftex_t
fig2dev -L pdftex $< $*.pdftex

help:
@echo
@echo "USAGE:"
@echo
@echo "make                display help information"
@echo "make proposal.pdf   format the proposal as PDF"
@echo "make main.pdf       format the dissertation as PDF"
@echo "make all            make proposal.pdf and main.pdf"
@echo "make view-proposal  format and view the proposal"
@echo "make view-main      format and view the dissertation"
@echo "make count          display an estimated word count"
@echo "make pub            put demodiss.pdf onto your homepage"
@echo "make clean          delete all intermediate files"
@echo

view-%: %.pdf
( okular --unique $< || evince $< ) &

main.pdf: figs/cuarms.pdf figs/diagram.pdf makefile.txt proposal.tex main.bbl

makefile.txt: makefile
```

```
expand makefile >makefile.txt

count:
detex main.tex | tr -cd '0-9A-Za-z \n' | wc -w

all: proposal.pdf main.pdf

pub: main.pdf
rsync -t $+ $(HOME)/public_html/demodiss.pdf

clean:
rm -f *.aux *.log *.err *.out
rm -f *~ *.lof *.toc *.blg *.bbl
rm -f makefile.txt

distclean: clean
rm -f figs/*.pdf proposal.pdf main.pdf
```

B.2 refs.bib

```
@BOOK{Lamport86,
TITLE = "{LaTeX} --- a document preparation system --- user's guide
and reference manual",
AUTHOR = "Lamport, L.",
PUBLISHER = "Addison-Wesley",
YEAR = "1986"}

@REPORT{Moore95,
TITLE = "How to prepare a dissertation in LaTeX",
AUTHOR = "Moore, S.W.",
YEAR = "1995"}

@article{goodman_semiring_1999,
title = {Semiring {Parsing}},
volume = {25},
url = {https://aclanthology.org/J99-4004},
number = {4},
urldate = {2022-08-01},
journal = {Computational Linguistics},
author = {Goodman, Joshua},
year = {1999},
note = {Place: Cambridge, MA
Publisher: MIT Press},
pages = {573--606},
file = {Full Text PDF:/Users/judah/Zotero/storage/VIIER4EP/Goodman - 1999 - Semiring Parsing.pdf:application/pdf},
}

@article{finkensiep_modeling_2021,
title = {{MODELING} {AND} {INFERRING} {PROTO}-{VOICE} {STRUCTURE} {IN} {FREE} {POLYPHONY}},
abstract = {Voice leading is considered to play an important role in the structure of Western tonal music. However, the expl
language = {en},
author = {Finkensiep, Christoph and Rohrmeier, Martin},
year = {2021},
file = {Finkensiep and Rohrmeier - 2021 - MODELING AND INFERRING PROTO-VOICE STRUCTURE IN FR.pdf:/Users/judah/Zotero/storage
}

@book{lerdahl_generative_2010,
address = {Cambridge, Mass.},
edition = {Repr.},
title = {A generative theory of tonal music},
isbn = {978-0-262-62107-6},
language = {eng},
publisher = {MIT Press},
author = {Lerdahl, Fred and Jackendoff, Ray},
year = {2010},
file = {Lerdahl and Jackendoff - 2010 - A generative theory of tonal music.pdf:/Users/judah/Zotero/storage/YX5TQ8JQ/Lerdahl
}

@misc{johannes_ms3_2021,
title = {ms3 - {Parsing} {MuseScore} 3},
shorttitle = {Ms3},
url = {https://github.com/johentsch/ms3},
abstract = {A parser for annotated MuseScore 3 files.},
author = {Johannes, Hentschel},
year = {2021},
}

@article{neuwirth_annotated_2018,
title = {The {Annotated} {Beethoven} {Corpus} ({ABC}): {A} {Dataset} of {Harmonic} {Analyses} of {All} {Beethoven} {String}
volume = {5},
issn = {2297-2668},
shorttitle = {The {Annotated} {Beethoven} {Corpus} ({ABC})},
url = {https://www.frontiersin.org/article/10.3389/fdigh.2018.00016/full},
doi = {10.3389/fdigh.2018.00016},
abstract = {STATISTICS The ABC contains expert harmonic annotations of all sixteen string quartets (70 movements) by Beethov
language = {en},
urldate = {2022-10-11},
journal = {Frontiers in Digital Humanities},
author = {Neuwirth, Markus and Harasim, Daniel and Moss, Fabian C. and Rohrmeier, Martin},
month = jul,
year = {2018},
pages = {16},
file = {Neuwirth et al. - 2018 - The Annotated Beethoven Corpus (ABC) A Dataset of.pdf:/Users/judah/Zotero/storage/CGR7ITXM/
}

@phdthesis{finkensiep_structure_2022,
type = {{PhD} thesis},
title = {The {Structure} of {Free} {Polyphony}},
author = {Finkensiep, Christoph},
year = {2022},
}

@article{marsden_schenkerian_2010,
title = {Schenkerian {Analysis} by {Computer}: {A} {Proof} of {Concept}},
volume = {39},
issn = {0929-8215, 1744-5027},
shorttitle = {Schenkerian {Analysis} by {Computer}},
url = {http://www.tandfonline.com/doi/abs/10.1080/09298215.2010.503898},
doi = {10.1080/09298215.2010.503898},
abstract = {A system for automatically deriving a Schenkerian reduction of an extract of tonal music is described. Schenkeri
language = {en},
number = {3},
urldate = {2022-11-14},
journal = {Journal of New Music Research},
author = {Marsden, Alan},
month = sep,
year = {2010},
pages = {269--289},
file = {Marsden - 2010 - Schenkerian Analysis by Computer A Proof of Conce.pdf:/Users/judah/Zotero/storage/2CJAF2LK/Marsden
}

@article{kidney_algebras_2021,
title = {Algebras for weighted search},
volume = {5},
issn = {2475-1421},
url = {https://dl.acm.org/doi/10.1145/3473577},
doi = {10.1145/3473577},
abstract = {DONNACHA OISÍN KIDNEY, Imperial College London, United Kingdom NICOLAS WU, Imperial College London, United Kingd
language = {en},
number = {ICFP},
urldate = {2022-11-27},
journal = {Proceedings of the ACM on Programming Languages},
author = {Kidney, Donnacha Oisín and Wu, Nicolas},
month = aug,
year = {2021},
pages = {1--30},
file = {Kidney and Wu - 2021 - Algebras for weighted search.pdf:/Users/judah/Zotero/storage/PNPGP9YX/Kidney and Wu - 2021 -
}

@book{russell_artificial_2010,
address = {Upper Saddle River},
edition = {3rd ed},
series = {Prentice {Hall} series in artificial intelligence},
title = {Artificial intelligence: a modern approach},
isbn = {978-0-13-604259-4},
shorttitle = {Artificial intelligence},
language = {en},
publisher = {Prentice Hall},
author = {Russell, Stuart J. and Norvig, Peter and Davis, Ernest},
year = {2010},
keywords = {Artificial intelligence},
file = {Russell et al. - 2010 - Artificial intelligence a modern approach.pdf:/Users/judah/Zotero/storage/38YISIQK/Russell e
}
```

Appendix C

Project Proposal

January 21, 2023

C.1 Abstract

A piece of music can be described using a sequence of chords, representing a higher level harmonic structure of a piece. There is a small, finite set of chord types, but each chord can be realised on the musical surface in a practically infinite number of ways. Given a score, we wish to infer the underlying chord types.

The paper *Modeling and Inferring Proto-voice Structure in Free Polyphony* describes a generative model that encodes the recursive and hierarchical dependencies between notes, giving rise to a grammar-like hierarchical system [?]. This proto-voice model can be used to reduce a piece into a hierarchical structure which encodes an understanding of the tonal/harmonic relations of a piece.

Christoph Finkensiep suggests in his paper that the proto-voice model may be an effective way to infer higher level latent entities, such as harmonies or voice leading schemata. Thus in this project I will ask the question: is this parsing model an effective way to annotate harmonies? By 'effective' we are referring to two things:

- Accuracy: can the model successfully emulate how experts annotate harmonic progressions in musical passages?
- Practicality: can the model be used to do this within a reasonable time frame?

While the original model could in theory be used to generate harmonic annotations, its exhaustive search strategy would be prohibitively time-consuming in practice for any but the shortest musical extracts; one half measure can have over 100,000 valid derivations [?]. My approach will be to explore the use of heuristic search algorithms to solve this problem.

C.2 Substance and Structure

C.2.1 Core: Search

The core of this project is essentially a search problem characterised as follows:

- The state space S is the set of all possible partial reductions of a piece along with each reduction step that has been done so far.
- We have an initial state $s_o \in S$, which is the empty reduction, corresponding to the unreduced surface of the piece. The score is represented as a sequence of slices grouping notes that sound simultaneously. We are also given the segmentation of the original chord labels that we wish to retrieve.
- We have a set of actions, A modelled by a function $action : A \times S \to S$. These actions correspond to a single reduction step.
  - The reduction steps are the inverses of the operations defined by the generative proto-voice model.
- Finally we have a goal test, $goal : S \to \{true, false\}$ which is true iff the partial reduction s has exactly one slice per segment of the input.

  - This means the partial reduction s contains a sequence of slices which start and end positions corresponding to the segmentation of the piece.

- At the first stage, this will be implemented using a random graph search algorithm, picking each action randomly, according to precomputed distributions.

C.2.2 Core: Evaluation

The second core task is to create an evaluation module that iterates over the test dataset, and evaluates the partial reduction computed by the search algorithm above. This will be done by comparing the outputs to ground truth annotations from the Annotated Beethoven Corpus.

In order to do this I will make use of the statistical harmony model from Finkensiep's thesis, *The Structure of Free Polyphony* [?]. This model provides a way of mapping between the slices that the algorithm generates and the chords in the ground truth. This can be used to empirically measure how closely the slices match the expert annotations.

C.2.3 Extension

Once the base search implementation and evaluation module have been completed, the search problem will be tackled by heuristic search methods, with different heuristics to be trialled and evaluated against each other. The heuristics will make use of the chord profiles from Finkensiep's statistical harmony model discussed above. These profiles relate note choices to the underlying harmony. Hence the heuristics may include:

- How the chord types relate to the pitches used.

- How the chord types relate which notes are used as ornamentation, and the degree of ornamentation.

- Contextual information about neighboring slices

C.2.4 Overview

The main work packages are as follows:

Preliminary Reading – Familiarise myself with the proto-voice model, and read up on similar models and their implementations. Study heuristic search algorithms.

Dataset Preparation – Pre-process the Annotated Beethoven Corpus into a suitable representation for my algorithm.

Basic Search – Implement a basic random search algorithm that takes in surface and segmentations, and outputting the sequence of slices matching the segmentations.

Evaluation Module – Implement an evaluation module to evaluate the output from the search algorithm.

End-to-end pipeline – Implement a full pipeline from the data to the evaluation that can be used to compare different reductions.

Heuristic Design – Extension – Trial different heuristics and evaluate their performance against each other.

Dissertation – I intend to work on the dissertation throughout the duration of the project. I will then focus on completing and polishing the project upon completion.

C.3 Starting Point

The following describes existing code and languages that will be used for this project:

Haskell – I will be using Haskell for this project as it is used in the proto-voice implementation. It must be noted that my experience with Haskell is limited, as I was first introduced to it via an internship this summer (July to August 2022).

Python – Python will be used for data handling. I have experience coding in Python.

Prior Research - Over the summer I have been reading the literature on computational models of music, as well as various parsing algorithms such as semi-ring parsing [?], and the CYK algorithm, which is used in the implementation of the proto-voice model.

Protovoices-Haskell – The paper *Modeling and Inferring Proto-Voice Structure in Free Polyphony* [?] includes an implementation of the proto-voice model in Haskell. A fork of this repository will form the basis of my project. This repository includes as parsing module which will be used to perform the actions in the search space of partial reductions. There is module that can exhaustively enumerate reductions of a piece, but this is infeasible in practice due to the blowup of the derivation forest.

MS3 – This is a library for parsing MuseScore Files and manipulating labels [?], which I will use as part of the data processing pipeline.

ABC – The *Annotated Beethoven Corpus* [?] contains analyses of all Beethoven string quartets composed between 1800 and 1826), encoded in a human and machine readable format. This will be used as a dataset for this project.

C.4 Success Criteria

This project will be deemed a success if I complete the following tasks:

- Develop a baseline search algorithm that uses the proto-voice model to output a partial reduction of a piece of music up to the chord labels.
- Create an evaluation module that can take the output of the search algorithm and quantitatively evaluate its accuracy against the ground truth annotations by providing a score based on a statistical harmony model.
- Extension: Develop one or more search algorithms that use additional heuristics to inform the search, and compare the accuracy with the baseline algorithm.

C.5 Timetable

Time frame	Work	Evidence
Michaelmas (Oct 4 to Dec 2)		
Oct 14 to Oct 24	*Oct 14*: Final proposal deadline. Preparation work: familiarise myself with the dataset and the proto-voice model implementation. Work on manipulating reductions using the proto-voice parser provided by the paper.	None
Oct 24 to Nov 7	Dataset preparation and handling.	Plot useful metrics about the dataset using Haskell
Nov 7 to Nov 21	Random Search implementation	None
Nov 21 to Dec 5	Evaluation Module. Continue with search implementation.	Evaluate a manually created derivation and plot results
Vacation (Dec 3 to Jan 16)		
Dec 5 to Dec 11	Evaluate performance of random search. Begin to work on extensions	Plot results
Dec 10 to Dec 21	Trial different heuristics. Implement an end-to-end pipeline from input to evaluation.	None
Dec 21 to Dec 27	None	None
Dec 27 to Jan 10	Continue trialing and evaluating heuristics	*Fulfill success criterion: At least one heuristic technique gives better performance than random search.*
Lent (Jan 17 to Mar 17)		
Jan 4 to Jan 20	Buffer Period to help keep on track	None
Jan 20 to Feb 3	*Feb 3*: Progress Report Deadline. Write progress report and prepare presentation. Write draft *Evaluation* chapter	Progress Report (approx. 1 page)
Feb 3 to Feb 17	Prepare presentation.	*Feb 8 – 15*: Progress Report presentation
Feb 17 to Mar 3	*Feb 17*: How to write a Dissertation briefing. Write draft Introduction and Preparation chapters. Incorporate feedback on Evaluation chap-	Send draft Introduction and Preparation chapter to supervisor

C.6 Resources

I plan to use my own laptop for development: MacBook Pro 16-inch, M1 Max, 32GB Ram, 1TB SSD, 24-core GPU.

All code will be stored on a GitHub repository, which will guarantee protection from data loss. I will easily be able to switch to using university provided computers upon hardware/software failure.

The project will be built upon work that has been done in the DCML (Digital cognitive musicology lab) based in EPFL. The files are in their Github repository, and I have been granted permission to access their in-house datasets of score annotations, as well as software packages which are used to handle the data.

C.7 Supervisor Information

Peter Harrison, head of Centre for Music and Science at Cambridge, has agreed to supervise me for this. We have agreed on a timetable for supervisions for this year. I am also working with Christoph Finkensiep, a PHD student at the DCML, and originator of the proto-voice model. Professor Larry Paulson has agreed to be the representative university teaching officer.