

Anik Roy

**A probabilistic programming
language in Ocaml**

Diploma in Computer Science

Christ's College

October 15, 2019

Proforma

Name: **Anik Roy**
College: **Christ's College**
Project Title: **A probabilistic programming language in Ocaml**
Examination: **Diploma in Computer Science, July 2020**
Word Count: **$-^1$ (well less than the 12000 limit)**
Project Originator: **Dr R. Mortier**
Supervisor: **Dr R. Mortier**

Original Aims of the Project

Work Completed

Special Difficulties

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

Declaration

I, Anik Roy of Christ's College, being a candidate for Part II of the Computer Science Tripos [or the Diploma in Computer Science], 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.

Signed [signature]

Date [date]

Contents

1	Introduction	1
1.1	Overview of the files	1
1.2	Building the document	1
1.2.1	The makefile	2
1.3	Counting words	2
2	Preparation	5
3	Implementation	7
3.1	Verbatim text	7
3.2	Tables	8
3.3	Simple diagrams	8
3.4	Adding more complicated graphics	8
4	Evaluation	13
4.1	Printing and binding	13
4.1.1	Things to note	13
4.2	Further information	14
5	Conclusion	15
	Bibliography	17
A	Latex source	19
A.1	diss.tex	19
A.2	proposal.tex	27
A.3	propbody.tex	27
B	Makefile	29
B.1	Makefile	29
B.2	refs.bib	29

List of Figures

3.1	A picture composed of boxes and vectors.	9
3.2	A diagram composed of circles, lines and boxes.	9
3.3	Example figure using encapsulated postscript	10
3.4	Example figure where a picture can be pasted in	10
3.5	Example diagram drawn using xfig	11

Acknowledgements

This document owes much to an earlier version written by Simon Moore [2]. His help, encouragement and advice was greatly appreciated.

Chapter 1

Introduction

1.1 Overview of the files

This document consists of the following files:

- `Makefile` — The Makefile for the dissertation and Project Proposal
- `diss.tex` — The dissertation
- `probody.tex` — Appendix C – the project proposal
- `proposal.tex` — A \LaTeX main file for the proposal
- `figs` – A directory containing diagrams and pictures
- `refs.bib` — The bibliography database

1.2 Building the document

This document was produced using $\text{\LaTeX} 2_{\epsilon}$ which is based upon \LaTeX [1]. To build the document you first need to generate `diss.aux` which, amongst other things, contains the references used. This is done by executing the command:

```
latex diss
```

Then the bibliography can be generated from `refs.bib` using:

```
bibtex diss
```

Finally, to ensure all the page numbering is correct run `latex` on `diss.tex` until the `.aux` files do not change. This usually takes 2 more runs.

1.2.1 The makefile

To simplify the calls to `latex` and `bibtex`, a makefile has been provided, see Appendix B.1. It provides the following facilities:

- `make`
Display help information.
- `make prop`
Run `latex proposal`; `xdvi proposal.dvi`.
- `make diss.ps`
Make the file `diss.ps`.
- `make gv`
View the dissertation using ghostview after performing `make diss.ps`, if necessary.
- `make gs`
View the dissertation using ghostscript after performing `make diss.ps`, if necessary.
- `make count`
Display an estimate of the word count.
- `make all`
Construct `proposal.dvi` and `diss.ps`.
- `make pub`
Make a `.tar` version of the `demodiss` directory and place it in my `public_html` directory.
- `make clean`
Delete all files except the source files of the dissertation. All these deleted files can be reconstructed by typing `make all`.
- `make pr`
Print the dissertation on your default printer.

1.3 Counting words

An approximate word count of the body of the dissertation may be obtained using:

```
wc diss.tex
```

Alternatively, try something like:

```
detex diss.tex | tr -cd '0-9A-Z a-z\n' | wc -w
```


Chapter 2

Preparation

This chapter is empty!

Chapter 3

Implementation

3.1 Verbatim text

Verbatim text can be included using `\begin{verbatim}` and `\end{verbatim}`. I normally use a slightly smaller font and often squeeze the lines a little closer together, as in:

```
GET "libhdr"

GLOBAL { count:200; all  }

LET try(ld, row, rd) BE TEST row=all
                        THEN count := count + 1
                        ELSE { LET poss = all & ~(ld | row | rd)
                              UNTIL poss=0 DO
                                { LET p = poss & -poss
                                  poss := poss - p
                                  try(ld+p << 1, row+p, rd+p >> 1)
                                }
                              }

LET start() = VALOF
{ all := 1
  FOR i = 1 TO 12 DO
  { count := 0
    try(0, 0, 0)
    writef("Number of solutions to %i2-queens is %i5*n", i, count)
    all := 2*all + 1
  }
  RESULTIS 0
}
```

3.2 Tables

Here is a simple example¹ of a table.

Left Justified	Centred	Right Justified
First	A	XXX
Second	AA	XX
Last	AAA	X

There is another example table in the proforma.

3.3 Simple diagrams

Simple diagrams can be written directly in \LaTeX . For example, see figure 3.1 on page 9 and see figure 3.2 on page 9.

3.4 Adding more complicated graphics

The use of \LaTeX format can be tedious and it is often better to use encapsulated postscript to represent complicated graphics. Figure 3.3 and 3.5 on page 11 are examples. The second figure was drawn using `xfig` and exported in `.eps` format. This is my recommended way of drawing all diagrams.

¹A footnote

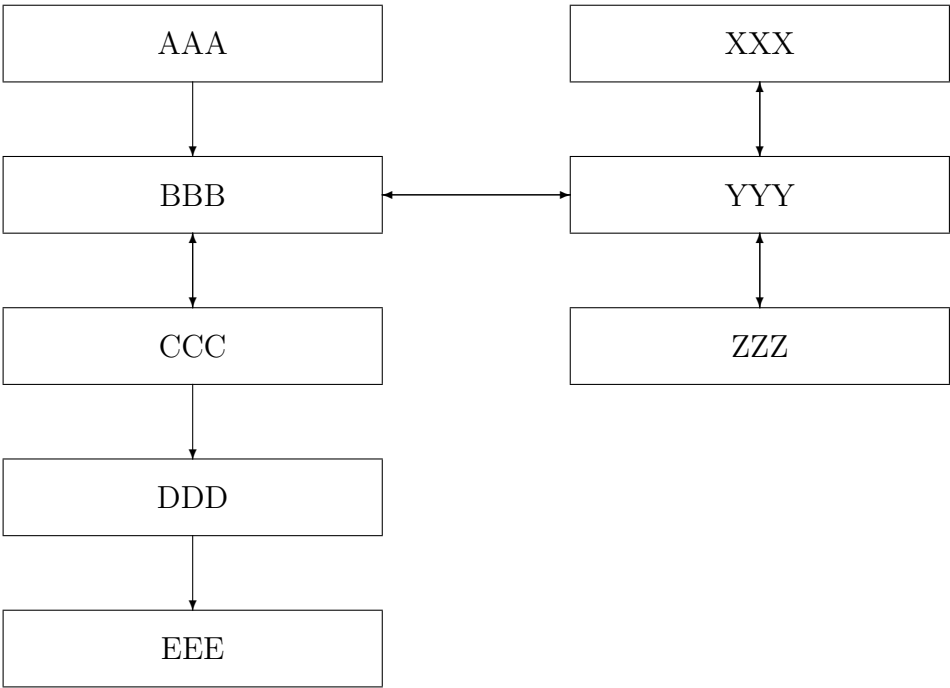


Figure 3.1: A picture composed of boxes and vectors.

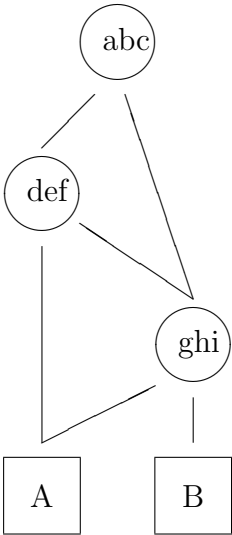


Figure 3.2: A diagram composed of circles, lines and boxes.

Figure 3.3: Example figure using encapsulated postscript

Figure 3.4: Example figure where a picture can be pasted in

Figure 3.5: Example diagram drawn using `xfig`

Chapter 4

Evaluation

4.1 Printing and binding

If you have access to a laser printer that can print on two sides, you can use it to print two copies of your dissertation and then get them bound by the Computer Laboratory Bookshop. Otherwise, print your dissertation single sided and get the Bookshop to copy and bind it double sided.

Better printing quality can sometimes be obtained by giving the Bookshop an MSDOS 1.44 Mbyte 3.5" floppy disc containing the Postscript form of your dissertation. If the file is too large a compressed version with **zip** but not **gnuzip** nor **compress** is acceptable. However they prefer the uncompressed form if possible. From my experience I do not recommend this method.

4.1.1 Things to note

- Ensure that there are the correct number of blank pages inserted so that each double sided page has a front and a back. So, for example, the title page must be followed by an absolutely blank page (not even a page number).
- Submitted postscript introduces more potential problems. Therefore you must either allow two iterations of the binding process (once in a digital form, falling back to a second, paper, submission if necessary) or submit both paper and electronic versions.
- There may be unexpected problems with fonts.

4.2 Further information

See the Computer Lab's world wide web pages at URL:

<http://www.cl.cam.ac.uk/TeXdoc/TeXdocs.html>

Chapter 5

Conclusion

I hope that this rough guide to writing a dissertation is \LaTeX has been helpful and saved you time.

Bibliography

- [1] L. Lamport. *LaTeX — a document preparation system — user's guide and reference manual*. Addison-Wesley, 1986.
- [2] S.W. Moore. How to prepare a dissertation in latex, 1995.

Appendix A

Latex source

A.1 diss.tex

```
% The master copy of this demo dissertation is held on my filespace
% on the cl file serve (/homes/mr/teaching/demodissert/)

% Last updated by MR on 2 August 2001

\documentclass[12pt,twoside,notitlepage]{report}

\usepackage{a4}
\usepackage{verbatim}

\input{epsf} % to allow postscript inclusions
% On thor and CUS read top of file:
% /opt/TeX/lib/texmf/tex/dvips/epsf.sty
% On CL machines read:
% /usr/lib/tex/macros/dvips/epsf.tex

\raggedbottom % try to avoid widows and orphans
\sloppy
\clubpenalty1000%
\widowpenalty1000%

\addtolength{\oddsidemargin}{6mm} % adjust margins
\addtolength{\evensidemargin}{-8mm}

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

\begin{document}

\bibliographystyle{plain}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title
```

```

\pagestyle{empty}

\hfill{\LARGE \bf Anik Roy}

\vspace*{60mm}
\begin{center}
\Huge
{\bf A probabilistic programming language in Ocaml} \\
\vspace*{5mm}
Diploma in Computer Science \\
\vspace*{5mm}
Christ's College \\
\vspace*{5mm}
\today % today's date
\end{center}

\cleardoublepage

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

\setcounter{page}{1}
\pagenumbering{roman}
\pagestyle{plain}

\chapter*{Proforma}

{\large
\begin{tabular}{ll}
Name: & \bf Anik Roy \\
College: & \bf Christ's College \\
Project Title: & \bf A probabilistic programming language in Ocaml \\
Examination: & \bf Diploma in Computer Science, July 2020 \\
Word Count: & \bf --\footnotemark[1] \\
(well less than the 12000 limit) \\
Project Originator: & Dr R.~Mortier \\
Supervisor: & Dr R.~Mortier \\
\end{tabular}
}
\footnotetext[1]{This word count was computed
by {\tt detex diss.tex | tr -cd '0-9A-Za-z $\tt\backslash$n' | wc -w}
}
\stepcounter{footnote}

\section*{Original Aims of the Project}

\section*{Work Completed}

\section*{Special Difficulties}

\newpage
\section*{Declaration}

```

I, Anik Roy of Christ's College, being a candidate for Part II of the Computer Science Tripos [or the Diploma in Computer Science], 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.

```
\bigskip
\leftline{Signed [signature]}
```

```
\medskip
\leftline{Date [date]}
```

```
\cleardoublepage
```

```
\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
```

```
\cleardoublepage      % just to make sure before the page numbering
                      % is changed
```

```
\setcounter{page}{1}
\pagenumbering{arabic}
\pagestyle{headings}
```

```
\chapter{Introduction}
```

```
\section{Overview of the files}
```

This document consists of the following files:

```
\begin{itemize}
\item {\tt Makefile} --- The Makefile for the dissertation and Project Proposal
\item {\tt diss.tex} --- The dissertation
\item {\tt propbody.tex} --- Appendix C -- the project proposal
\item {\tt proposal.tex} --- A \LaTeX\ main file for the proposal
\item {\tt figs} -- A directory containing diagrams and pictures
\item {\tt refs.bib} --- The bibliography database
\end{itemize}
```

```
\section{Building the document}
```

This document was produced using \LaTeXe which is based upon \LaTeX\cite{Lamport86}. To build the document you first need to generate {\tt diss.aux} which, amongst other things, contains the references used. This is done by executing the command:

```
{\tt latex diss}
```

`\noindent`

Then the bibliography can be generated from `{\tt refs.bib}` using:

`{\tt bibtex diss}`

`\noindent`

Finally, to ensure all the page numbering is correct run `{\tt latex}` on `{\tt diss.tex}` until the `{\tt .aux}` files do not change. This usually takes 2 more runs.

`\subsection{The makefile}`

To simplify the calls to `{\tt latex}` and `{\tt bibtex}`, a makefile has been provided, see Appendix~\ref{makefile}. It provides the following facilities:

`\begin{itemize}`

`\item{\tt make} \\\`

Display help information.

`\item{\tt make prop} \\\`

Run `{\tt latex proposal; xdvi proposal.dvi}`.

`\item{\tt make diss.ps} \\\`

Make the file `{\tt diss.ps}`.

`\item{\tt make gv} \\\`

View the dissertation using ghostview after performing `{\tt make diss.ps}`, if necessary.

`\item{\tt make gs} \\\`

View the dissertation using ghostscript after performing `{\tt make diss.ps}`, if necessary.

`\item{\tt make count} \\\`

Display an estimate of the word count.

`\item{\tt make all} \\\`

Construct `{\tt proposal.dvi}` and `{\tt diss.ps}`.

`\item{\tt make pub} \\\` Make a `{\tt .tar}` version of the `{\tt demodiss}` directory and place it in my `{\tt public_html}` directory.

`\item{\tt make clean} \\\` Delete all files except the source files of the dissertation. All these deleted files can be reconstructed by typing `{\tt make all}`.

`\item{\tt make pr} \\\`

Print the dissertation on your default printer.

`\end{itemize}`

`\section{Counting words}`

An approximate word count of the body of the dissertation may be obtained using:

```

{\tt wc diss.tex}

\noindent
Alternatively, try something like:

\verb/detex diss.tex | tr -cd '0-9A-Z a-z\n' | wc -w/

\cleardoublepage

\chapter{Preparation}

This chapter is empty!

\cleardoublepage
\chapter{Implementation}

\section{Verbatim text}

Verbatim text can be included using \verb|\begin{verbatim}| and
\verb|\end{verbatim}|. I normally use a slightly smaller font and
often squeeze the lines a little closer together, as in:

{\renewcommand{\baselinestretch}{0.8}\small\begin{verbatim}
GET "libhdr"

GLOBAL { count:200; all  }

LET try(ld, row, rd) BE TEST row=all
    THEN count := count + 1
    ELSE { LET poss = all & ~(ld | row | rd)
          UNTIL poss=0 DO
            { LET p = poss & -poss
              poss := poss - p
              try(ld+p << 1, row+p, rd+p >> 1)
            }
          }

LET start() = VALOF
{ all := 1
  FOR i = 1 TO 12 DO
    { count := 0
      try(0, 0, 0)
      writef("Number of solutions to %i2-queens is %i5*n", i, count)
      all := 2*all + 1
    }
  RESULTIS 0
}
\end{verbatim}
}

\section{Tables}

\begin{samepage}

```

Here is a simple example\footnote{A footnote} of a table.

```
\begin{center}
\begin{tabular}{l|c|r}
Left      & Centred & Right \\
Justified &         & Justified \\
%\hline\%[-2mm]
First     & A       & XXX \\
Second    & AA      & XX  \\
Last      & AAA     & X   \\
\end{tabular}
\end{center}
```

```
\noindent
There is another example table in the proforma.
\end{samepage}
```

```
\section{Simple diagrams}
```

Simple diagrams can be written directly in `\LaTeX`. For example, see figure~\ref{latexpic1} on page~\pageref{latexpic1} and see figure~\ref{latexpic2} on page~\pageref{latexpic2}.

```
\begin{figure}
\setlength{\unitlength}{1mm}
\begin{center}
\begin{picture}(125,100)
\put(0,80){\framebox(50,10){AAA}}
\put(0,60){\framebox(50,10){BBB}}
\put(0,40){\framebox(50,10){CCC}}
\put(0,20){\framebox(50,10){DDD}}
\put(0,00){\framebox(50,10){EEE}}

\put(75,80){\framebox(50,10){XXX}}
\put(75,60){\framebox(50,10){YYY}}
\put(75,40){\framebox(50,10){ZZZ}}

\put(25,80){\vector(0,-1){10}}
\put(25,60){\vector(0,-1){10}}
\put(25,50){\vector(0,1){10}}
\put(25,40){\vector(0,-1){10}}
\put(25,20){\vector(0,-1){10}}

\put(100,80){\vector(0,-1){10}}
\put(100,70){\vector(0,1){10}}
\put(100,60){\vector(0,-1){10}}
\put(100,50){\vector(0,1){10}}

\put(50,65){\vector(1,0){25}}
\put(75,65){\vector(-1,0){25}}
\end{picture}
\end{center}
\caption{\label{latexpic1}A picture composed of boxes and vectors.}
\end{figure}

\begin{figure}
\setlength{\unitlength}{1mm}
\begin{center}
```



```

\begin{picture}(100,70)
\put(47,65){\circle{10}}
\put(45,64){abc}

\put(37,45){\circle{10}}
\put(37,51){\line(1,1){7}}
\put(35,44){def}

\put(57,25){\circle{10}}
\put(57,31){\line(-1,3){9}}
\put(57,31){\line(-3,2){15}}
\put(55,24){ghi}

\put(32,0){\framebox(10,10){A}}
\put(52,0){\framebox(10,10){B}}
\put(37,12){\line(0,1){26}}
\put(37,12){\line(2,1){15}}
\put(57,12){\line(0,2){6}}
\end{picture}

\end{center}
\caption{\label{latexpic2}A diagram composed of circles, lines and boxes.}
\end{figure}

```

```
\section{Adding more complicated graphics}
```

The use of `\LaTeX` format can be tedious and it is often better to use encapsulated postscript to represent complicated graphics.

Figure~\ref{epsfig} and ~\ref{xfig} on page \pageref{xfig} are examples. The second figure was drawn using `{\tt xfig}` and exported in `{\tt eps}` format. This is my recommended way of drawing all diagrams.

```

\begin{figure}[tbh]
\centerline{\epsfbox{figs/cuarms.eps}}
\caption{\label{epsfig}Example figure using encapsulated postscript}
\end{figure}

```

```

\begin{figure}[tbh]
\vspace{4in}
\caption{\label{pastedfig}Example figure where a picture can be pasted in}
\end{figure}

```

```

\begin{figure}[tbh]

\centerline{\epsfbox{figs/diagram.eps}}
\caption{\label{xfig}Example diagram drawn using {\tt xfig}}
\end{figure}

```

```

\cleardoublepage
\chapter{Evaluation}

```

```
\section{Printing and binding}
```



```

\appendix

\chapter{Latex source}

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

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

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


\cleardoublepage

\chapter{Makefile}

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

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


\cleardoublepage

\chapter{Project Proposal}

\input{propbody}

\end{document}

```

A.2 proposal.tex

```

% This is a LaTeX driving document to produce a standalone copy
% of the project proposal held in propbody.tex. Notice that
% propbody can be used in this context as well as being incorporated
% in the dissertation (see diss.tex).

\documentclass[12pt,a4paper]{article}
\usepackage[parfill]{parskip}

\begin{document}

\include{propbody}

\end{document}

```

A.3 propbody.tex

```

% !TeX root = ./proposal.tex

\vfil

```

```

\begin{center}
  {\Large Computer Science Tripos -- Part II -- Project Proposal} \\
  \vspace{0.4in}
  {\huge \bf A probabilistic programming language in OCaml } \\
  \vspace{0.4in}
  {\large A. Roy, Christ's College} \\
  \vspace{0.1in}
  {\large \today} \\

\end{center}
\vspace{0.4in}

\vfil

{\bf Project Originator:} A. Roy
\vspace{0.1in}

{\bf Project Supervisor:} Dr R. Mortier
\vspace{0.1in}

{\bf Director of Studies:} Dr R. Mortier
\vspace{0.1in}

{\bf Project Overseers:} Dr~J.~A.~Crowcroft  \& Dr~T.~Sauerwald

\vfil

% Main document

\section*{Introduction}

A probabilistic programming language (PPL) is a framework in which one can create statistical models and have inference run

\section*{Starting Point}

There do exist PPLs for OCaml, such as IBAL \cite{Moore95} or HANSEI\footnote{HANSEI}

\section*{Resources Required}

I intend to use my personal laptop for the main development and subsequent write up (HP Pavilion 15, 8GB RAM, i5-8265U CPU)

The required software includes the ocaml compiler, with a build system (dune) and a package manager (opam). I will also use

For backups, I will use GitHub to host my git repository remotely, pushing frequently. I will also backup weekly to a USB

\section*{Substance and Structure of the Project}

\section*{Success Criteria}

\section*{Evaluation}

\section*{Extensions}

\section*{Schedule}

```

Appendix B

Makefile

B.1 Makefile

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"}  
  
@inproceedings{kiselyov2009embedded,  
  title={Embedded probabilistic programming},  
  author={Kiselyov, Oleg and Shan, Chung-Chieh},  
  booktitle={IFIP Working Conference on Domain-Specific Languages},  
  pages={360--384},  
  year={2009},  
  organization={Springer}  
}
```


Appendix C

Project Proposal

Computer Science Tripos – Part II – Project Proposal

A probabilistic programming language in OCaml

A. Roy, Christ's College

October 15, 2019

Project Originator: A. Roy

Project Supervisor: Dr R. Mortier

Director of Studies: Dr R. Mortier

Project Overseers: Dr J. A. Crowcroft & Dr T. Sauerwald

Introduction

A probabilistic programming language (PPL) is a framework in which one can create statistical models and have inference run on them automatically. A PPL can take the form of its own language (i.e. a DSL), or be embedded within an existing language (such as OCaml). Examples of

Starting Point

There do exist PPLs for OCaml, such as IBAL [2] or HANSEI¹

Resources Required

I intend to use my personal laptop for the main development and subsequent write up (HP Pavilion 15, 8GB RAM, i5-8265U CPU, running Ubuntu and Windows dual booted).

The required software includes the ocaml compiler, with a build system (dune) and a package manager (opam). I will also use the IDE VSCode with an OCaml extension, as well as git for version control and latex for the write up.

For backups, I will use GitHub to host my git repository remotely, pushing frequently. I will also backup weekly to a USB stick in case of failures. The software I require is available on MCS machines, so I'll be able to continue work in the event of a hardware failure.

Substance and Structure of the Project

Success Criteria

Evaluation

Extensions

Schedule

¹HANSEI