# Meeting Minutes

## Date: 10/10/14

## Previous Meeting: N/A

## Discussion:

Introduction to OCaml. Should look over the OCaml manual and begin to become aquianted with the language. Read around functional programming. Discussed the very basics of the project and what is to come.

## Date: 17/10/14

## Previous Meeting: 10/10/14

## Discussion:

More discussion about OCaml as a language. Brief discussion about Marco’s research paper *Fuzz.* Discussed potential direction of the project, maybe adapting *Fuzz* to do something different – could be quite complex. Should carry on working on OCaml, developing a knowledge of the language, and read the *Fuzz* research paper as best I can.

## Date: 24/10/14

## Previous Meeting: 17/10/14

## Discussion:

Short meeting about my progress with OCaml. I should carry on working through the examples in Real World Ocaml that I have found. Rather than completing the work in the OCaml shell I should try to compile files into byte code or native applications to be run from the command line.

## Date: 04/11/14

## Previous Meeting: 24/10/14

## Discussion:

Discussion mostly involving parsing and lexical analysis. I should read Chapter 16 of Real World OCaml to get an idea of how parsing works in OCaml now that I have an understanding of the language.

The goal for the next meeting is to create an arithmetic compiler that will be able to handle the set n | plus | minus | multiply | divide

## Date: 18/11/14

## Previous Meeting: 04/11/14

## Discussion:

Demonstration of my arithmetic compiler. Discussed issues with compilation of modules as well as reading in files into the compiler. As it stands the compiler allows the user to enter arithmetic expressions into the OCaml shell but does not read in a file, something that will be importand down the line.

Next steps involve dealing with the compilation errors and also the errors due to missing token combinations as the compiler throws errors when too many empty lines follow each other.

## Date: 20/11/14

## Previous Meeting: 18/11/14

## Discussion:

Looked at my arithmetic compiler again. I had been able to fix the errors surrounding the tokens, just required an extra rule in the parser to ignore extra empty lines. This will be the last meeting with Marco before he goes to the USA and I start exams etc.

The next step in the process is to start looking at lambda calculus and expanding the arithmetic compiler to parse and lex lambda expressions

## Date: 26/01/15

## Previous Meeting: 20/11/14

## Discussion:

Discussed project goal, editing *Fuzz* seems overly complex and would be much better to create my own compiler to handle arithmetics and lambda calculus with simple type checking. Time permitting it would be good to compare my work with Marco’s compiler mentioned in the previous weeks (<http://staff.computing.dundee.ac.uk/marcogaboardi/publication/Gaboardietal13submitted.pdf>)

## Date: 06/03/15

## Previous Meeting: 26/01/15

## Discussion:

Spoke about the project so far, progressing with the Lambda calculus compiler. Now that it reads from a text file it should begin to simpify lambda expressions. Look at beta simplification.

## Date: 26/03/15

## Previous Meeting: 06/03/15

## Discussion:

Discussed problems with beta simplification. I had been approaching the problem incorrectly, thinking of the lambda expressions as a list rather than applications of one another. Given information about the De Bruijn index which should help with the simplification process along with two examples of lambda calculus compilers written in ml. I need to make sure my lexer and parser are tranfering the data into the appropriate data types.

http://www.cs.dartmouth.edu/~mckeeman/cs118/lectures/14.html#anchor2

http://iml.univ-mrs.fr/~regnier/taylor/lambda.ml.html

## Date: 17/04/15

## Previous Meeting: 26/03/15

## Discussion:

Discussed the report and what should be submitted. Not really time to implement type checking anymore, just need to make sure the lambda compiler is properly tested and the report is well thought out. Sent a draft of the report to Marco for him to read through and check to make sure it is well structured and covers the correct material.