Topics in Programming Languages , CS6124D

Option 1: Learning OCaml and familiarizing with the available code.

Group Member: 1. Kundan Kumar M200285CS
2. Navin Kumar Das M200363CS

For this project we will be using Ocaml language , it has an advanced type system that helps catch your mistakes without getting in your way. It's used in environments where a single mistakes can cost millions and speed matters, Ocaml has a rich set of libraries and development tools. We covered the basics of Ocaml language including the data types, variables, control statement, looping, pattern matching, list, records, normal function, recursive function, exception handling, modules etc. We go through the already implemented code which was provided in the resource manual and found the code structure in already implemented interpreter as follows:

Main.ml : deals with the processing the command line, reading files, building lexers and parsers.

Lexer.mll : lexer.ml is automatically generated from lexer.mll and it contains lexical analysis(Tokenization).

Parser.mly : YACC grammar for parser, parser.ml and parser.mll is automatically generated from parser.mly.

Syntax.ml : syntax.mll and syntax.mli together generates syntax tree for the interpreter.

Core.ml: core.ml and core.mli containing all the rules available in the grammar.

Support.ml : collection of low level facility used by other modules.

File extension used:

.ml : ocaml files

.mli : headers files

.cmi : intermediate compiled file of header

.cmo : object file

References :

1. https://ocaml.org/docs/
2. https://www.cis.upenn.edu/~bcpierce/tapl/

3.https://youtube.com/playlist?list=PLea0WJq13cnCef-3KSU3qWFge9OGUlKx1