

# Extensions Write Up

James Song

May 2020

## 1 Lexical Environment

I chose to extend the project by implementing an evaluator that can evaluate an expression using the substitution model, the dynamically scoped environment model, and the lexically scoped environment model. The third required, as the specs specify, two changes from the implementation of the dynamic model: in functions and function application.

I let the three models be types that can be passed in to the function "evaluator." This takes less code overall and it is practical when running tests. To run tests, all I did was design one series of tests for "evaluator" and pass in each of the three models to execute those tests.

## 2 Additional Types

Additionally, I implemented some values for type binop: "LessEq", "GreaterThan" and "GreaterEq" to enlarge the comparisons capabilities of miniML. I did this by adding categories to different lines in `miniml_lex.mll`. The "sym\_table" was expanded to include the three comparisons.

A corresponding update was made to the Parser in `miniml_parse.mly`. The grammar was expanded to accommodate the new comparisons and more tokens were added.