CS51 Final Project Extensions writeup May 6, 2020

The first extension that I implemented was to change the Miniml repl to display the evaluations using substitution semantics and dynamic semantics simultaneously. I first deleted the evaluate function in evaluation.ml because I knew that that abstraction would not be needed any longer, since I planned to call the eval_s and eval_d functions directly. Then, I edited the miniml repl code to run two evaluations, res_s and res_d for substitution and dynamic semantics, respectively. To print the results of the evaluations, I changed the match statement for res to a match statement matching the tuple, (res_s, res_d) and printed both results together.

Here is an example of the result:

For the user input, "let x = 2 in let f = fun y -> x * y in let x = 1 in f 21 ;;" the repl prints:

--> Let(x, Num(2), Let(f, Fun(y, Binop(Times, Var(x), Var(y))), Let(x, Num(1), App(Var(f), Num(21))))) s=> 42 d=> 21

I also tried to implement floats and float operators, but I had trouble with mli and cmi files and was unable to completely finish that extension and get it to compile. In attempts to implement floats, I'm confident that given more time, it would be pretty simple to add all additional atomic types and the corresponding literals and operators.