CS51 - Final Project

Ray Chen

May 2020

1. Extension: Lexicon Semantics Scope:

   As my extension, I wrote an additional evaluator for lexicon environment semantics. As described it the textbook, it uses similar features as the dynamic environment semantics where the interpreter refers to an environment / table of stored values to determine the evaluation of a variable. Unlike dynamic environments however, lexicon environments take advantage of closures when defining and adding functions to the environment, taking a snapshot of the environment / values defined at the time of recording the new function. Consequently, this makes lexicon environments ”more correct” than the dynamic environment.

   This feature was tested out in expr11 where the expression:
   ”$let\ x = 5\ \ in\ \ let\ f = fun\,y- > x + y\ \ in\ \ let\ x = 10\ \ in\ \ f0$ ”

   In the lexicon environment, the interpreter spits out a 5 as expected while dynamic environment's interpreter returns a 10.

   When implementing the lexicon extension, the main 2 features were 1) creating closures for function definitions and 2) creating temp environments for defining recursive functions and overriding said recursive functions.

   For addressing closures, I had to make a separate copy of the environment to preserve the original values at hand (this took place in Env.close).

   For addressing let rec, I had to modify the extend function such that it knows how to update existing environments such that it gives the variable name the proper definition.