

CS51 Final Project Write-up

James Bardin

4 May 2022

1 Extensions

The extension I chose to implement for my Final Project was an interpreter that manifests lexical scope instead of dynamic. The difference with this interpreter is that unlike dynamic scoping, in lexical scoping the functions are evaluated in the environment created when the function is defined.

The big difference between eval d and eval l are the fun and app matches. We will start with the evaluation of a function. The difference made in eval l from eval d is that the function returns a closure containing the function and the env when it is created. So, we have something along the lines of,

$$\text{Fun } (f, x) \longrightarrow \text{Closure } (\text{exp}, \text{env}) \quad (1)$$

And similarly when a function application is evaluated, the body is evaluated with the environment from the same closure.

I tested both of the evals using the recommended expressions given in the textbook assignment description and found that they both evaluated to their correct corresponding values.

So, the changes are not massive between the two scopes. If I had more time what could have been done is to make a single generic evaluation function, and include another input like a boolean or similar idea, if there are for instance 3 scopes, and have "if, then" statements to evaluate with different scopes. This, would have been a nice way to shorten the code, and clearly show the differences between the inner workings of the different scopes. But, I ran out of time and could not get this working in time.