

COMP-335
Evan Chessman
Bryan Jensen
October 27, 2014

Homework #2 Written Answer

1. The worst case runtime for *get-fundef* and *lookup* is $O(n)$, as worst case the identifier is not in the list and you have to iterate every element once. To improve runtime speed, we could use something like a dictionary from Python or a map from C++ to provide a constant time lookup speed, $O(1)$, mapping from the identifier to the actual function definition.
2. The evaluation scheme for this interpreter is eager, as the function expects ExprC's in its argument slot, but immediately converts that expression into a number before storing it as a binding in the *env* variable.
3. Global pi
 - a. There are multiple ways to solve this dilemma. The best approach we could think of would be to modify the *interp* function to have an "else" option, under the *idC* branch of the type-case conditional. This "else" would be triggered if the lookup failed to find the identifier in the *env* variable, and perform another lookup trying to find it in a secondary list of bindings, possibly called *global*, where the *pi* bound instance would live. Only if both lookups failed would the error "name not found" be thrown.
 - b. With the approach we suggested in (a), there would be no issue with the function calls given in example, and all three would evaluate to the same value, roughly 8.1416.
 - c. The first print would output 10, and the second would print out 3.1416. This makes sense along with our understanding of scope defining two different *pi* variables, one inside the

function *mypi* and one outside. Changing the value of one by passing in 5 as the value of the inner *pi* doesn't change the outer one. However, this example does not demonstrate the concept of environment immutability as no attempt is made to change the contents of either environment after it has been created.