University of Copenhagen

Assignment 2: Counting parentheses
- resubmission

Jens Fredskov (chw752)

October 2, 2013

1 Introduction

The following report describes the implementation and testing of *let* and *lambda* in Haskell library for interpreting the APLisp dialect.

2 Implementation

2.1 Let

let is implemented in `eval`. If the scope is empty we evaluate tyhe body. If there actually are variables in the scope we evaluate each of their values and using `bindVar` with `local` binds it in the a local environment. Finally we again evaluate the body using the new environment.

2.2 Lambda

In `eval` *lambda* is merely returned. This means that we can store a *lambda* in a *let* and then apply it using `funcall`. This is done in `apply` where we allow either a single argument or a list of arguments. `apply` sends its arguments to `evalLambda` which binds every parameter to the respective argument of in a local scope and then evaluates the body. If there is a mismatch between arguments and parameters an error is returned.

3 Testing

Extra test cases have been added in `APLispTest.hs` to test *let* and *lambda*, all of which returns the expected results.

4 Conclusion

The implementation of *let* and *lambda* have been described. The tests have been documented and all return the expected result.