Advanced Programming
Home Assignment 2

Carlo Rosso, Chinar Shah

Contents

1 Introduction

This week both of us have little time to work on the assignment, therefore we did not manage to complete all the tasks. Particularly, we managed to implement the first and the second tasks. We are confident in the correctness of our solutions, because we have tested them with the provided examples, so we got a test coverage over 80% (almost 90%).
Following, I describe how to run the tests:

1. unzipping `a1-handout.zip`;
2. entering the `a1-handout` directory through the terminal;
3. running `cabal test` in the terminal.

Note that you can run `cabal --enable-coverage test` to see the coverage of the codebase and you can also see what lines are not covered by the tests.

2 Task - Printing

We implement the `Print` constructor in the `Exp` data type, in the most flexible way we can think of, considering the second task.

For example changing `envExtend` to `mapExtend` we are able to define a function that extends both the environment and the state.

Therefore define a function `combineStates` that takes in the `State` type which is defined as a tuple of a list of strings and a list of pairs of `Val` and `Val`. The function makes it easier to propagate the state and the console through the computation.

```
mapExtend :: key -> val -> [(key, val)] -> [(key, val)]
mapExtend v val env = (v, val) : env

combineStates :: State -> State -> State
combineStates (console1, store1) (console2, store2) =
  (console1 ++ console2, foldr expand store1 store2)
  where
    expand (key, val) = mapExtend key val
```

3 Task - Key-value store

We think this implementation was hard, indeed it took us a lot of time to complete it and we are not sure it is correct. We have tested it with the provided examples and they work just fine.

We also experimented with a more compact sytax instead of the `do` notation:

```
eval (KvGet k) = evaleval (KvPut k v) =
  eval v >>= \val ->
    (\() -> val) <$> (eval k >>= \key -> evalKvPut key val)
eval (KvGet k) = eval k >>= \key -> evalKvGet key
```

4 Task - Type checking

We have not implemented this task, because we did not have enough time to complete it.

5 Answers to Questions

We have not answered the questions, because we did not have enough time to complete it.