Haskell Assignment 1

Philip Rajani Lassen (vgh804)
Nicolas Ringsmose Larsen (vgn209)

September 2018

1 evalExpr

For evalExpr in particular, we had issues with passing the environment through the monad. We felt that this was necessary, due to the way the SubsM type was defined. We weren't sure how to "retrieve" the value from the monad, such that we could use it recursively. This made us unable to evaluate expression of ArrayVal type and gave us a lot of trouble when assigning values as well.

2 Process

In order to implement the code we spent a large portion of the time reading about Monads. We hoped that this would help speed our ability to produce the code, however we still lacked some understanding. When we wrote the the code we had difficulties getting our implementations to pass the type-checker. This was due to our lack of understanding of why functions and types were chosen to be what they were.

3 Assessment

We struggled to understand the newtype declaration of SubsM. We similarly struggled to grasp the purpose of the type SubsM for implementing the interpreter. This made the implementation very difficult as we had little to no intuition about how to work with the monad. It is quite possible that at some points we would have been able to reduce the number of lines code, as well as simplify some of the implementations of the function if we had a better grasp for the utility of type SubsM. At some stages we wrote code that worked, even when we weren't completely sure what they were doing.

4 Testing

Since we had trouble with the monads and how to pass the environments along, we also were not sure how to test it. Thus, the only tests we did, came from

using the onlineTA. We were not able to pass all the tests of the onlineTA, due to our failure to implement evalExpr on the types for Array [Value] as well as our inability to implement it for the function names.