

The biggest issue I had during this assignment has to be the typing function. Initially it was straight forward but when I got to the ITE, Abs, App, LetIn it got difficult. Due to the long period of time of not coding in haskell I forgot that I had to store the results of the function in variables with do. Due to that I wasted a bunch of time trying to find the bug to why the code wouldn't work. Overall I am not satisfied with this function due to the readability being super low. I wished I could of done this code in a way that it was more readable. Right now it seems like a huge blob of code.

vscode ended up giving me a recommendation to changing my typeCheck code from

```
typeCheck x = case typing map.empty x of
  Just x -> show x
  Nothing -> "Type error"
```

To

```
typeCheck :: Expr -> String
typeCheck x = maybe "Type Error" show (typing Map.empty x)
```

this made sense to me after I looked at it for a bit and decided to stick with it. I felt that this made my code more readable.

For the readExpr I ended up using the built in function/method read. Lastly for typingArith and typingEq I just did it as stated in the question and it worked right of the back.

Lastly due to the last assignment and how I received a low mark due to edge cases I am now somewhat insecure on if my code actually works as intended. Due to this I ended up using test cases posted in discord. So far I have around 100 test cases. Hopefully I have covered edge cases that could potentially give the wrong outputs. I have added my text cases/classmates testcases that are available on discord to the report.

Var "x1"

Plus (CInt 1) (CInt 2)

ITE (CBool True) (CInt 1) (CInt 2)

Abs "x" TInt (Var "x")

App (Abs "x" TInt (Var "x")) (CInt 1)

LetIn "x" TInt (CInt 1) (Var "x")

Var "x1"

CInt 1

CBool True

CBool False

Plus (CInt 1) (CInt 2)

Plus (CInt 1) (CBool True)

Minus (CInt 2) (CInt 1)

Minus (CInt 1) (CBool True)

Equal (CInt 1) (CInt 1)

Equal (CBool True) (CBool True)

Equal (CInt 1) (CBool True)

Equal (Abs "x" TInt (Var "x")) (Abs "y" TInt (Var "y"))

ITE (CBool True) (CInt 1) (CInt 2)

ITE (CBool True) (CBool True) (CBool False)

ITE (CInt 0) (CInt 10) (CInt 20)

Abs "x" TInt (Var "x")

Abs "x" TBool (Var "x")

Abs "x" TInt (CBool True)

Abs "x" TInt (App (Abs "y" TBool (Var "y")) (Var "x"))

App (Abs "x" TInt (Var "x")) (CInt 1)

App (Abs "x" TBool (Var "x")) (CBool True)

App (Abs "x" TInt (Var "x")) (CBool True)

LetIn "x" TInt (CInt 1) (Var "x")

LetIn "x" TBool (CBool True) (Var "x")

LetIn "x" TInt (CBool True) (Var "x")

LetIn "f" (TArr TInt TInt) (Abs "x" TInt (Var "x")) (App (Var "f") (CBool True))

Abs "x" TInt (Plus (Var "x") (CInt 1))

Abs "x" TInt (Abs "y" TInt (Plus (Var "x") (Var "y")))

LetIn "x" TInt (CInt 2) (Minus (Var "x") (CInt 1))

Var "x1"

CInt 0

CInt 1

CBool True

CBool False

Plus (CInt 1) (CInt 2)

Plus (CInt 1) (CBool True)

Plus (CInt 1) (CBool False)

Plus (CBool True) (CInt 2)

Plus (CBool False) (CInt 2)

ITE (CBool True) (CBool False) (CBool True)

ITE (CBool False) (CBool False) (CBool True)

ITE (CBool True) (CInt 1) (CInt 2)

ITE (CBool False) (CInt 1) (CInt 2)

ITE (CBool True) (CInt 1) (CBool False)

ITE (CBool False) (CInt 1) (CBool False)

ITE (CInt 1) (CBool False) (CBool True)

ITE (CInt 0) (CBool False) (CBool True)

ITE (CInt 1) (CInt 1) (CInt 2)

ITE (CInt 0) (CInt 1) (CInt 2)

ITE (CInt 1) (CInt 1) (CBool False)

ITE (CInt 0) (CInt 1) (CBool False)

Abs "x" TInt (Var "x")

Abs "x" TBool (Var "x")

Abs "y" TInt (Var "x")

Abs "y" TBool (Var "x")

Abs "x" TInt (CBool False)

Abs "x" TInt (CInt 0)

App (Abs "x" TInt (Var "x")) (CInt 1)

App (Abs "x" TBool (Var "x")) (CInt 1)

App (Abs "y" TInt (Var "x")) (CInt 1)

App (Abs "y" TBool (Var "x")) (CInt 1)
App (Abs "x" TInt (CBool False)) (CInt 1)
App (Abs "x" TInt (CInt 0)) (CBool False)
App (Abs "x" TInt (Var "x")) (CBool False)
App (Abs "x" TBool (Var "x")) (CBool False)
App (Abs "y" TInt (Var "x")) (CBool False)
App (Abs "y" TBool (Var "x")) (CBool False)
App (Abs "x" TInt (CBool False)) (CBool False)
App (Abs "x" TInt (CInt 0)) (CBool False)
LetIn "x" TInt (CInt 1) (Var "x")
LetIn "x" TInt (CBool True) (Var "x")
LetIn "x" TBool (CInt 1) (Var "x")
LetIn "y" TInt (CInt 1) (Var "x")
LetIn "y" TInt (CInt 1) (Abs "x" TInt (CBool False))
CInt 5
CBool True
Plus (CInt 1) (CInt 2)
Minus (CBool True) (CInt 1)
Equal (CBool False) (CBool True)
Equal (CInt 0) (CBool True)
Equal (CBool True) (CInt 0)
App (Abs "x" TInt (Plus (Var "x") (CInt 1))) (CInt 5)
LetIn "x" TInt (CInt 1) (Plus (Var "x") (CInt 2))
App (Abs "x" TInt (Var "x")) (CBool True)
Abs "x" TInt (Abs "y" TBool (Var "x"))
App (Abs "x" TInt (ITE (Equal (Var "x") (CInt 0)) (CInt 1) (App (Abs "y" TInt (Plus (Var "y") (CInt 1))) (CInt 2))))) (CInt 3)
LetIn "f" (TArr TInt TInt) (Abs "x" TInt (Minus (Var "x") (CInt 1))) (LetIn "g" (TArr TInt TInt) (Abs "y" TInt (App (Var "f") (Plus (Var "y") (CInt 2))))) (App (Var "g") (CInt 5)))

ITE (CBool True) (App (Abs "x" TInt (ITE (CBool False) (Minus (Var "x") (CInt 1)) (Plus (Var "x") (CInt 1)))) (CInt 10)) (CInt 0)

App (App (Abs "f" (TArr TInt TInt) (Abs "g" (TArr TInt TInt) (Abs "x" TInt (App (Var "f") (App (Var "g") (Var "x")))))) (Abs "y" TInt (Plus (Var "y") (CInt 1)))) (Abs "z" TInt (Minus (Var "z") (CInt 1)))

App (Abs "x" TInt (ITE (Equal (Var "x") (CInt 0)) (CBool True) (Plus (Var "x") (CInt 1)))) (CBool False)

Plus (App (Abs "x" TInt (Var "x")) (CInt 1)) (App (Abs "x" TInt (Var "x")) (CInt 1))

Minus (App (Abs "x" TInt (Var "x")) (CInt 1)) (App (Abs "x" TInt (Var "x")) (CInt 1))

Equal (App (Abs "x" TInt (Var "x")) (CInt 1)) (App (Abs "x" TInt (Var "x")) (CInt 1))

Plus (CInt 1) (Plus (CInt 1) (Plus (CInt 1) (Plus (CInt 1) (Plus (CInt 1) (Plus (CInt 1) (Plus (CInt 1) (Plus (CInt 1) (Plus (CInt 1) (CInt 2))))))))))

LetIn "x" TInt (CInt 1) (LetIn "y" TBool (CBool True) (Plus (Minus (Plus (Var "x") (CInt 2)) (CBool False)) (ITE (Equal (Var "y") (CBool True)) (CInt 10) (CInt 20))))