プログラミング言語基礎論第 2 回レポート

Kwame Ackah Bohulu, 1631133

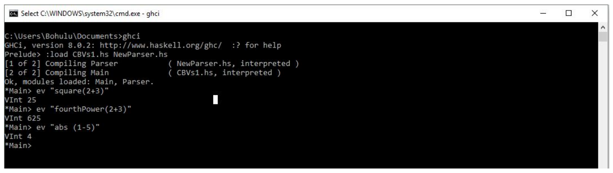
21-08-2017

[1-a] –Built-inFunctions
builtinFns :: [([Char], [Char])]
builtinFns = [(”square”, ”lambda x . x * x”),
(”fourthPower”, ”lambda x . square (square x)”),
(”abs”,”lambda x . if x < 0 then (0 - x) else x”)]

[1-b] ToEnv :: [([Char], [Char])] ->[(Variable, Val)]
ToEnv [] =[]
ToEnv ((k,v):ps) = (k,vv) : ToEnv ps
where vv = ev v

globalEnv :: Env
globalEnv = ToEnv builtinFns

[1-c] ev :: [Char] -> Val
ev s = expval (parseProg s) globalEnv



```
C:\Users\Bohulu\Documents>ghci
GHCi, version 8.0.2: http://www.haskell.org/ghc/  :? for help
Prelude> :load CBVs1.hs NewParser.hs
[1 of 2] Compiling Parser           ( NewParser.hs, interpreted )
[2 of 2] Compiling Main             ( CBVs1.hs, interpreted )
Ok, modules loaded: Main, Parser.
*Main> ev "square(2+3)"
VInt 25
*Main> ev "fourthPower(2+3)"
VInt 625
*Main> ev "abs (1-5)"
VInt 4
*Main>
```

[2-1]　5 行目から 14 行目の部分を次のように変更

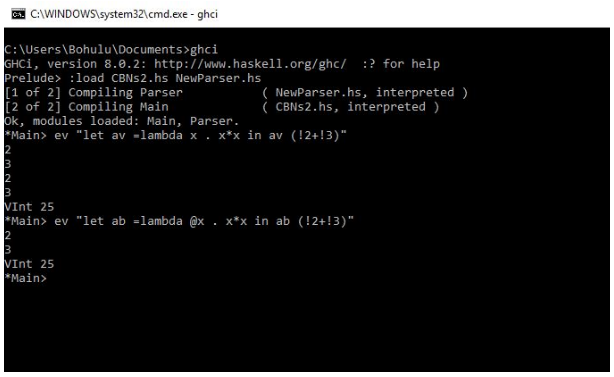
data Val = VInt Int
| VBool Bool
| VClosure Variable Expr Env Bool
deriving (Eq, Show)

showVal :: Val -> [Char]
showVal (VInt m) = show m
showVal (VBool b) = show b
showVal (VClosure v e r b) =
"Closure [lambda " ++(if b then "@" else "")++ show v ++ " . " ++ show e
++ "]"

72 行目から 78 行目の部分を次のように変更
expval (Fun x e) env = VClosure x e env True

expval (SFun x e) env = VClosure x e env False

expval (Apply e1 e2) env = v ‘seq‘ g ‘seq‘ expval body newenv
where g = expval e1 env
v = (if t then delay e2 env else Evaled (expval e2 env))
VClosure x body env’ t = g
newenv = updateEnv x v env’



```
C:\WINDOWS\system32\cmd.exe - ghci

C:\Users\Bohulu\Documents>ghci
GHCi, version 8.0.2: http://www.haskell.org/ghc/  :? for help
Prelude> :load CBNs2.hs NewParser.hs
[1 of 2] Compiling Parser           ( NewParser.hs, interpreted )
[2 of 2] Compiling Main             ( CBNs2.hs, interpreted )
Ok, modules loaded: Main, Parser.
*Main> ev "let av =lambda x . x*x in av (!2+!3)"
2
3
2
3
VInt 25
*Main> ev "let ab =lambda @x . x*x in ab (!2+!3)"
2
3
VInt 25
*Main>
```