Functional Programming: Assignment 1

Group 60

Lucas van der Laan
s1047485

1

1. (a) Lyrics first checks if argument is 0 or ‘n‘, if 0 show the 0 message, if ‘n‘ compute lyrics ‘n‘
   (b) Lyrics computes seats
   (c) Lyrics computes shows the message + lyrics (n-1)
   (d) Lyrics recursively goes down to lyrics 0
2. I think that the program will run indefinitely because it will never reach lyrics 0
3. The ‘-‘ operator is an infix function and lyrics requires an integer

2

f1: getSmaller
f2: strRepeat
f3: gcd

3

e1 = 1 + 125 * 8 ‘div‘ 10 - 59
{ definition of * }
=> 1 + 1000 ‘div‘ 10 - 59
{ definition of div }
=> 1 + 100 - 59
{ definition of - }
=> 1 + 41
{ definition of + }
=> 42

e2 = not True || True && False
{ definition of not }
=> False || True && False
{ definition of && }
=> False || False

{ definition of || }
=> False

e3 = 1 + 2 == 6 - 3
{ definition of + }
=> 3 == 6 - 3
{ definition of - }
=> 3 == 3
{ definition of == }
=> True

e4 = "1 + 2" == "6 - 3"
{ definition of == }
=> False

e5 = "1111 + 2222" == "1111" ++ " + " ++ "2222"
{ definition of ++ }
=> "1111 + 2222" == "1111 + " ++ "2222"
{ definition of ++ }
=> "1111 + 2222" == "1111 + 2222"
{ definition of == }
=> True

4

1. double 5 = incr (incr 0) where incr y = 5 + y
   { definition of incr }
   => incr (5 + 0)
   { definition of + }
   => incr (5)
   { definition of incr }
   => 5 + 5
   { definition of + }
   => 10
2. The evaluation order used is applicative order

5

See Database.hs

6

See Say.hs, SayTest.lhs, and Hello.hs

7

See Triangle.hs