```
ecture 5.
         foldr :: (a>b>b) > b > [a] > b
        foldr f k [] = k
foldr f k (x:xs) = xf (foldr f k xs)
                        =fx (folder f kxs)
foldr(+) 0 [1, 2,3] =
  (+) 1 (follo (+) 0 [2,3])
  1 + (fold- (4) 0 (2,3])
   1 (x + (foldr (+) 0 [3]))
 1 + (2+(3+ folder (+) 0 []))
 1 + (2 + (3 + 0))
      sum xs = foldr (+) 0
```

tunction composition

$$f(x) = x^2 + 1$$

square :: Int -> Int square x= x x

succ :: Int -> Int succ x = x+1

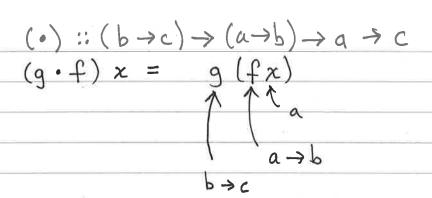
succesquere : lat -) lut succesquere = (XXX) +1

a

KS = [1, 2, 3]

Squarers 2 5 10]

map (succ · square) TS



h: c→d.

h.g.f

Monoids

$$x + 0 = x$$
 ? Unit laws $0 + x = x$

$$X+(y+2) = (X+y)+2$$
 } associative

mplus macro

mplus: a > a > a mzero : a

folder mplus maero

IPX

Type Classes

class Monoid me where mplus : a - a - a

mtero :: a nu

instance Monoid Int where mplus = (+)

mitero = 0

instance Monoid [a] where

mplus =

mzero =

mplus : [a] -> [a] -> [a]

ruplus = (++)

mtero :: [a]

mtero = []

principal type: everything has just 1 type.