

## Lecture 3

$\text{cata} :: \text{Bifunctor } f \Rightarrow (f \ a \ b \rightarrow b) \rightarrow \text{Mu } f \ a \rightarrow b$   
 $\text{cata } \phi (In \ x) = \phi (bimap \ id \ (\text{cata } \phi) \ x)$

data  $\text{Mu } f \ a = \text{In } (f \ a \ (\text{Mu } f \ a))$

$\text{para} :: \text{Bifunctor } f \Rightarrow (f \ a \ (b, \text{Mu } f \ a) \rightarrow b) \rightarrow \text{Mu } f \ a \rightarrow b$   
 $\text{para } \phi (In \ x) = \phi (bimap \ id \ (\text{cata } \phi \ \Delta \ id) \ x)$   
( $\text{para } \phi$ )  $\Delta$   $id$

dropWhile :: (a -> Bool) -> [a] -> [a]

dropWhile p [1] = [2]

dropWhile p (x:xs) = if p x then else x:xs

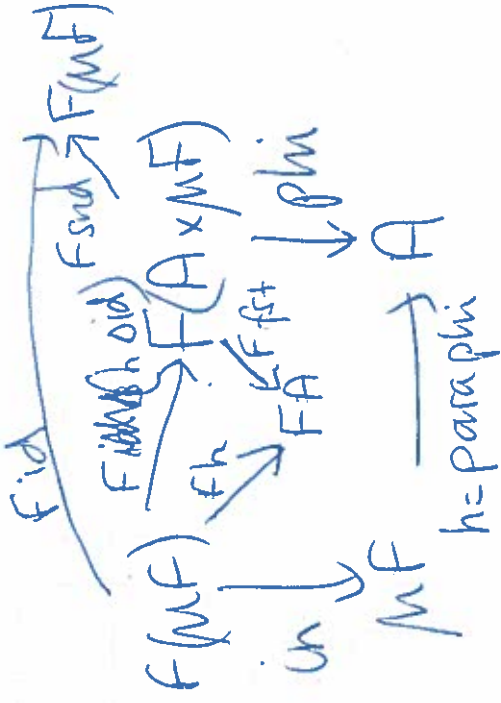
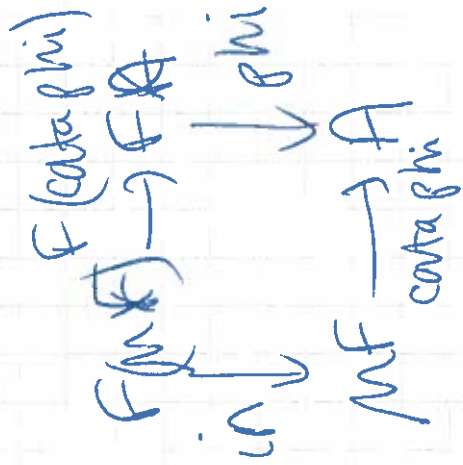
dropWhile even [2,4,5,6,8]

dropWhile p (x:xs) = ... x ... dropWhile p xs ... ?

dropWhile <sup>empty</sup> [5,6,8] = ... 5 ... dropWhile empty [8] ...

[ ]

h (x:xs) = ... x ... h xs ... [ ]



every cata is a para

cata  $\phi_i = \text{para } \psi_i$  where  $\psi_i = \text{cata } \phi_i$

every para is a cata with some post-processing

para  $\phi_i = \text{post} \circ \text{cata } \psi_i$  where  $\psi_i = \text{cata } \phi_i$   
 $\text{post} = \text{fst}$



Zygomorphism

for para,  $h = id$

$h(x \circ x) = x \circ x \circ x \circ x \circ x \circ x \circ x \circ x \circ x$

data Tree a = ~~Tip~~ Tip a | Bin (Tree a) (Tree a)

perfect :: Tree a -> Bool

perfect (Tip x) = True

perfect (Bin t u) = perfect t & perfect u & ht t == ht u

ph :: Tree a -> (Bool, Int)

ph (Tip x) = (True, 1)

ph (Bin t u) = let (b,m) = ph t; (c,n) = ph u in (b & c & m == n, 1 + max m n)

Zygo :: Bifunctor f => (f a (b,c) -> b) -> (f a c -> c) -> Mon f a -> b

Zygo phi psi = fst o zygo' phi psi

Zygo' :: Bifunctor f => (f a (b,c) -> b) -> (f a c -> c) -> Mon f a -> (b,c)

Zygo' phi psi (In x) = let x' = bimap id (zygo' phi psi) x in  
(phi x', psi (bimap id snd x'))

every Para is a Zygo ooo

every Zygo is a post-processed cata ooo

mutumorphisin

mutu :: Bifunctor  $f \Rightarrow (f a (b, c) \rightarrow b) \rightarrow (f a (b, c) \rightarrow c) \rightarrow$

$\text{Mu} f a \rightarrow b$

eg  $\text{minimax}$



isomorphism

(course-of-values iteration<sup>n</sup>)

$h(x:xs) = \dots x \dots \text{data} \leftarrow \text{cata}$

$ws :: [Int, Rat]$

$\text{Nat} \rightarrow \text{Rat}$   
knapsack :: Int  $\rightarrow$  Rat

knapsack c = maximum<sub>⊆</sub> [ v + knapsack (c-w) | (w,v) ∈ wvs, 0 ≤ w ≤ c ]

histo :: Bifunctor f ⇒ (f a (Cofree(f a) b) → b) →  
Mun f a → b

Cofree comonad on f a  
histo phi = root o cata (λ x → Node (phi x) x)

Cofree comonad

data Cofree f b = Node b (f (Cofree f b))

data Mu f ~~Node~~ = In (f (Mu f ~~Node~~))

root :: Cofree f b → b

root (Node x t) = x

subs :: Cofree f b → f (Cofree f b)

subs (Node x t) = t



data  $L a b = Nil \mid Cons a b$

what is  $Cons (L a) c$ ? effectively,

data  $Cons a c = Cons Nil c$   
 $\mid Cons c a (Cons a c)$

# Duality

para  $f_a(b, x, M, u, f, a)$

zygo

mito

histo

apo  $f_a(b + Nu, f, a)$

cozygo

comutu

futu