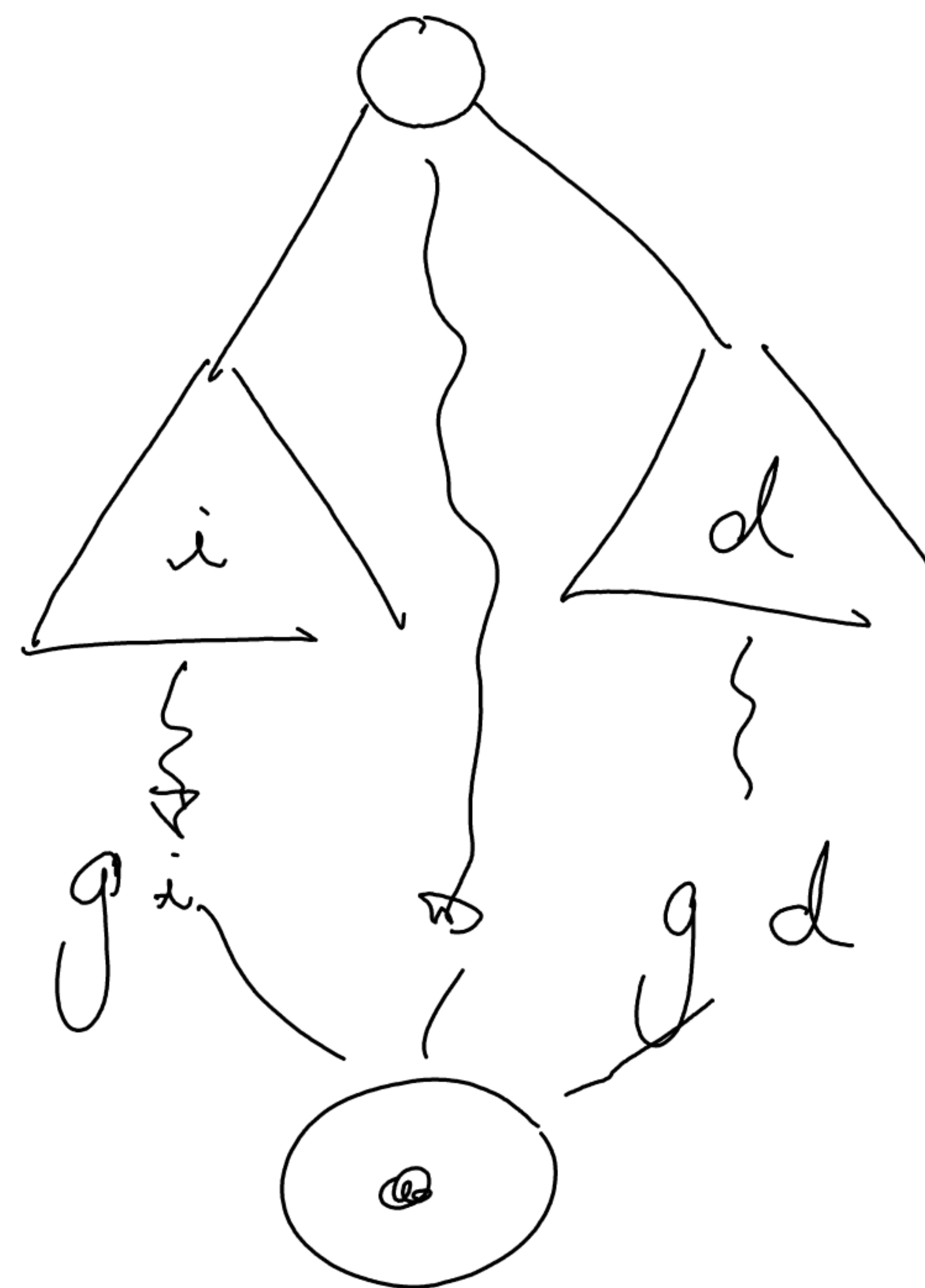
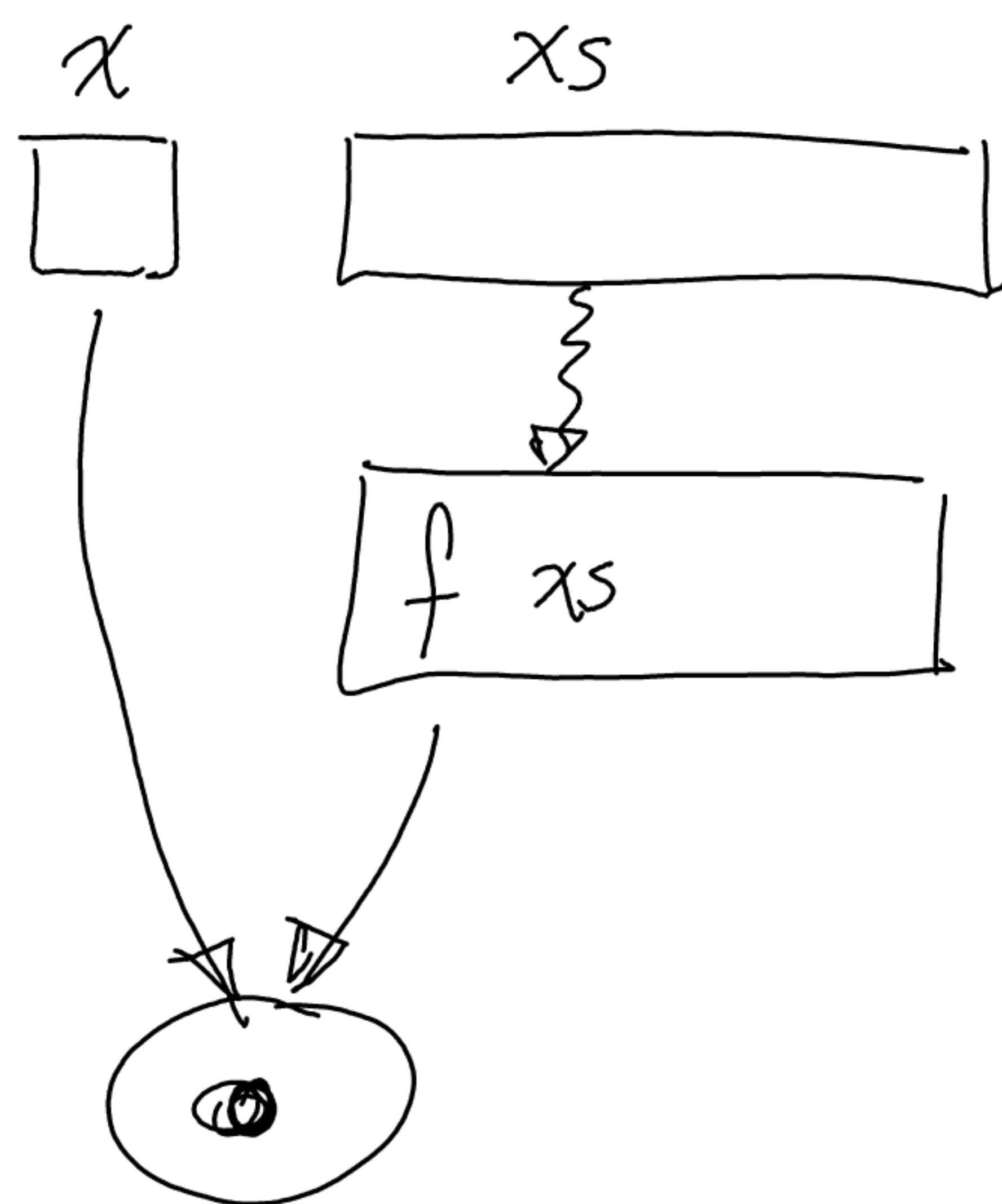


$\text{data } [a] = [] \mid \text{Cons } \underline{a} \underline{[a]}$

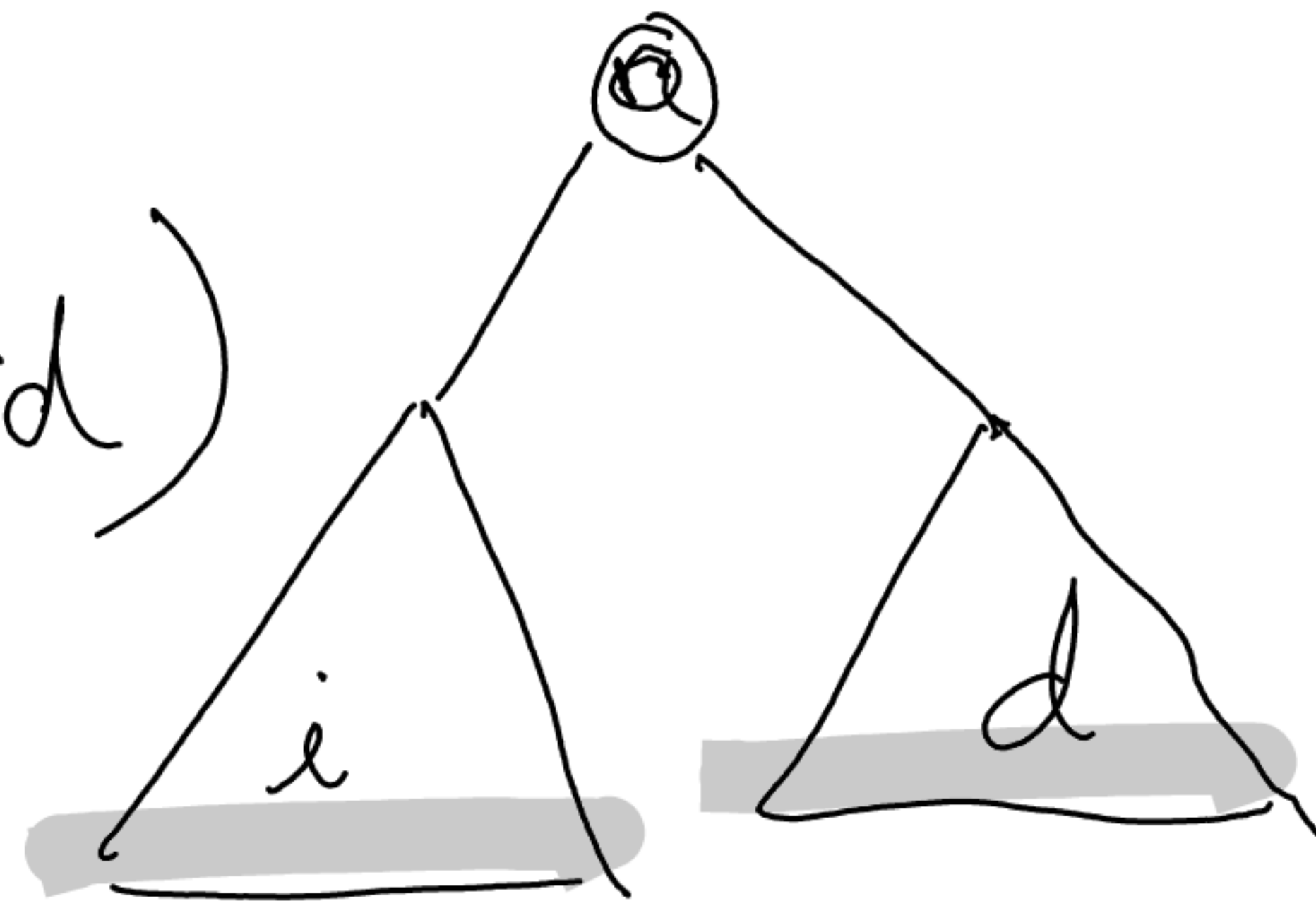


$\text{cantNodos} :: \text{AEB } a \rightarrow \text{Int}$
 $\text{cantNodos } t = \text{foldAEB}$

$\text{cantNodos}(\text{Hoja } 2) \rightsquigarrow 1$

$(\text{const } 1)$ \nearrow $(\lambda a \rightarrow 1)$
 $(\lambda r i a r d \rightarrow 1 + r i + r d)$

t
 \nearrow
 Se pueden sacar.



$1 + \text{cantNodos } i + \text{cantNodos } d$

$\text{cantNodos} = \text{foldAEB } (\text{const } 1) \ (\lambda i - d \rightarrow 1 + i + d)$

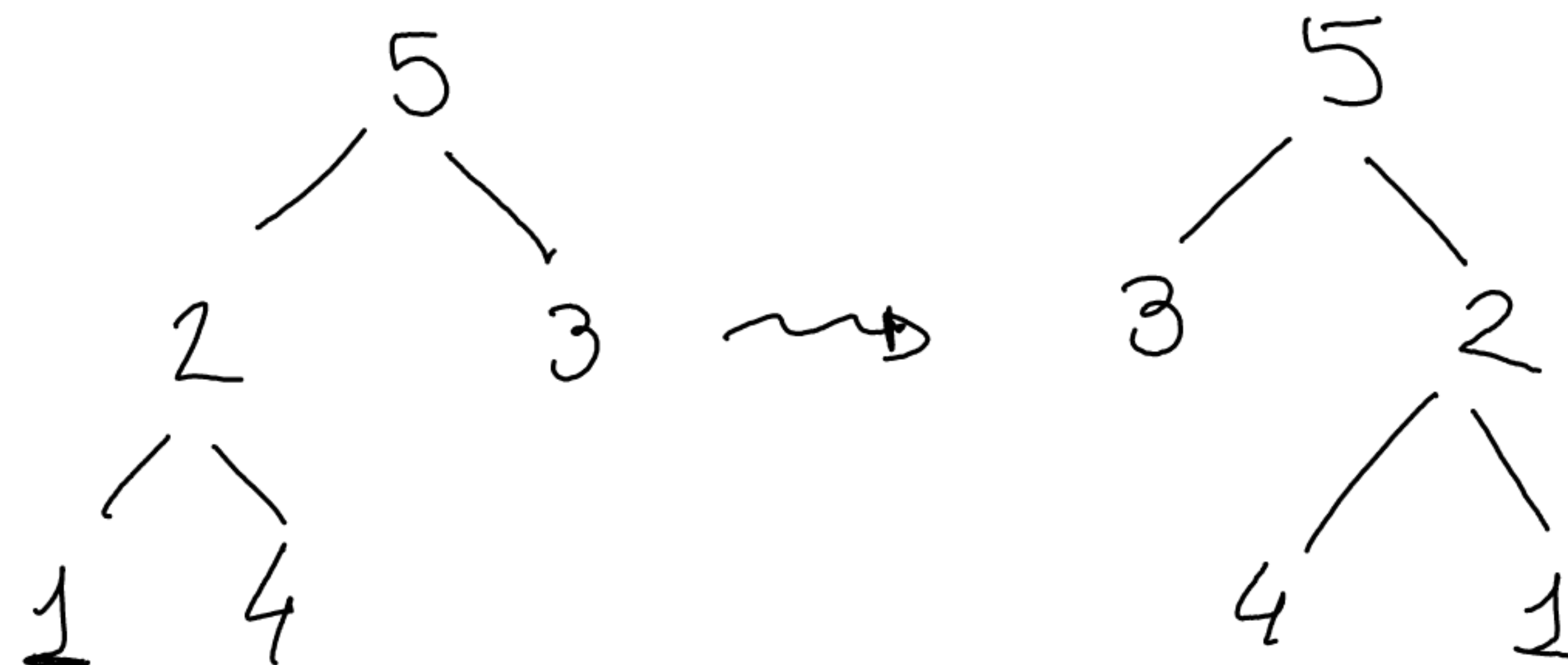
espejo :: AEB a \rightarrow AEB a

espejo = fold AEB Hoja

(\l r d \rightarrow

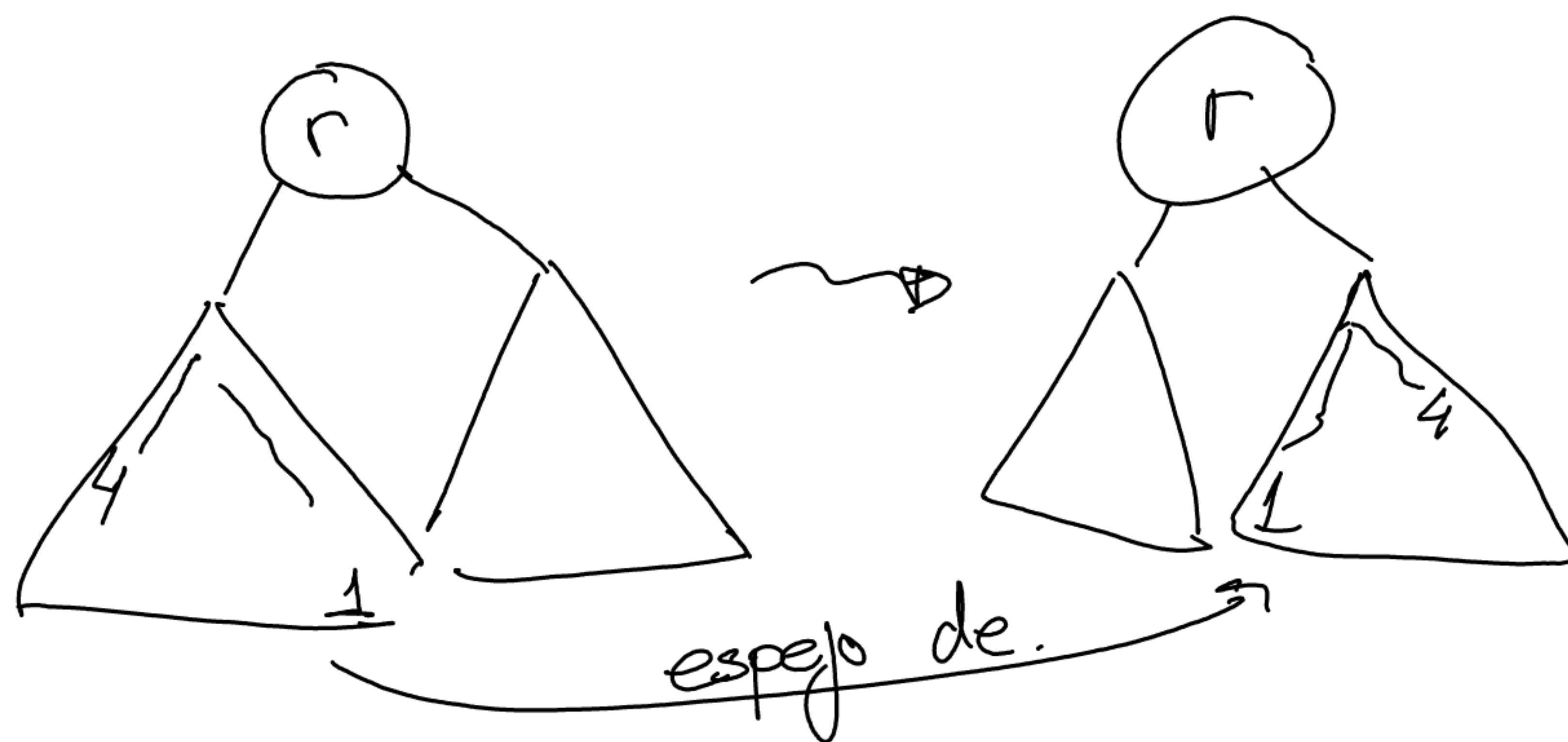
Bin d r i)

! ya es el espejo
o del árbol izquierdo
original.



Hoja :: a \rightarrow AEB a

~~$\lambda n \rightarrow$~~ Hoja ~~n~~
 $a \rightarrow$ AEB a



data AB a = Nil | Bin (AB a) a (AB a)

foldAB :: b → (b → a → b → b) → AB a → b

recrAB :: b → (AB a → b → a → AB a → b → b) → AB a → b

recrAB fNil fBin t =

case t of

Nil → fNil

Bin i r d → fBin i (rec i) r d (rec d)

where

rec = recrAB fNil fBin

data Poli a = X | Cte a | Suma (Poli a) (Poli a) | Prod (Poli a) (Poli a)

evaluar :: (Num a) => a -> Poli a -> a

evaluar x X = x

evaluar x (Cte n) = n

evaluar x (Suma p q) = (evaluar x p) + (evaluar x q)

evaluar x (Prod p q) = (evaluar x p) * (evaluar x q)

$X^2 + 2X + 1 \equiv \text{Suma} (\text{Suma} (\text{Prod } X \ X) (\text{Prod} (\text{Cte } 2) \ X)) (\text{Cte } 1)$

foldPoli :: b -> (a -> b) -> (b -> b -> b) -> (b -> b -> b) -> Poli a -> b

evaluar' x = foldPoli x id (+) (*)

data RoseTree a = Rose a [RoseTree a]

foldRT :: (a → [b] → b) → RoseTree a → b

foldRT f (Rose r ts) = f r (map (foldRT f) ts)

ramas :: RoseTree a → [[a]]

ramas = foldRT (λ r res →
 if null res then [[r]]
 else
 map (r:) (concat res))

(++) :: [c] → [c] → [c]

concat :: [[c]] → [c]

concat res ≈ [b₁, b₂ ... b_n, c₁ ... c_k, d₁ ... d_m]

map :: (a → b) → [a] → [b]

