Reminder

Let C, D be categories. The product category $C \times D$ has objects $Obj_C \times Obj_D$, and morphisms $(f, f') :: (a, a') \to (b, b')$, $(f, f')\ (x, x') = (fx, f'x)$.

Composition is defined as

$$(f, f') \circ (g, g') = (f \circ g,\ f' \circ g')$$

A functor with signature

$$C \times D \to E$$

is called a bifunctor.

① Let

data Pair a b = Pair a b,

F a b = Pair a b

F f g = λx. Pair (f (fst x)) (g (snd x))

Show that F is a bifunctor.

F1 is obvious: a map $f :: (a, a') \to (b, b')$ gets mapped to

$$F f :: \underbrace{\text{Pair } a\ a'}_{= F\, a\, a'} \to \underbrace{\text{Pair } b\, b'}_{= F\, b\, b'} \quad \checkmark$$

F2:

$$F\, f\, f' \circ F\, g\, g' = \lambda x.\, \mathrm{Pair}\ (f\ (\mathrm{fst}\ x))\ \ (f'\ (\mathrm{snd}\ x))$$

$$\circ$$

$$\lambda x.\, \mathrm{Pair}\ (g\ (\mathrm{fst}\ x))\ \ (g'\ (\mathrm{snd}\ x))$$

$$= \lambda x.\ \mathrm{Pair} \Big(f\ (\mathrm{fst}\ (\mathrm{Pair}\ (g\ (\mathrm{fst}\ x))\ (g'\ (\mathrm{snd}\ x)))) \Big)$$

$$\Big(f'\ (\mathrm{snd}\ (\mathrm{Pair}\ (g\ (\mathrm{fst}\ x))\ (g'\ (\mathrm{snd}\ x)))) \Big)$$

$$= \lambda x.\ \mathrm{Pair}\ \big(f\ (g\ (\mathrm{fst}\ x)) \big)\ \big(f'\ (g\ (\mathrm{snd}\ x)) \big)$$

$$= \lambda x.\ \mathrm{Pair}\ \big((f \circ g)\ (\mathrm{fst}\, x) \big)\ \big((f' \circ g')\ \ (\mathrm{snd}\ x) \big)$$

$$= F\, \big((f, f') \circ (g, g') \big)$$

✓

F3:

$$F\ id_a\ \ id_b = \lambda x.\ \mathrm{Pair}\ \ (id_a\ (\mathrm{fst}\, x))\ \ \ (id_b\ (\mathrm{snd}\ x))$$

$$= \lambda x.\ \ \mathrm{Pair}\ \ (\mathrm{fst}\ x)\ \ (\mathrm{snd}\ x)$$

$$= id_{F a b}$$

✓

□

②

~~Let~~ Show that the following are isomorphic.

type Maybe a = Nothing | Just a ,

type Mabye' a = Either (Const () a) (Identity a)

Solution:

Let f :: Maybe a -> Maybe' a

f Nothing = Left (Const ())

f (Just x) = Right (Identity x)

g :: Maybe' a -> Maybe a

g (Left (Const () _)) = Nothing

g (Right (Identity x)) = Just x

now

$g \circ f$ Nothing = Nothing and

$g \circ f$ (Just x) = Just x $\Rightarrow g \circ f = id$

as desired $\square$

③

Show that

$$\texttt{data PreList a b = Nil | Cons a b}$$

is an instance of Bifunctor.

Solution

Let $f\ a\ b := \text{PreList}\ a\ b$, and define the bimap of type

$$(a \to b) \to (a' \to b') \to f\ a\ a' \to f\ b\ b'$$

as

$$\text{bimap}\ f\ g\ \text{Nil} = \text{Nil}$$

$$\text{bimap}\ f\ g\ (\text{Cons}\ x\ y) = \text{Cons}\ (f\ x)\ (g\ y)$$

F1: Trivial ✓

F2:

$$\text{bimap}\ f\ f' \circ \text{bimap}\ g\ g' = \lambda \begin{cases} \text{Nil}.\ \text{Nil} \\ \text{Cons}\ x\ y.\ \text{Cons}\ (f x)\ (f' y) \end{cases} \circ \lambda \begin{cases} \text{Nil}.\ \text{Nil} \\ \text{Cons}\ x\ y.\ \text{Cons}\ (g x)\ (g' y) \end{cases}$$

$$= \lambda \begin{cases} \text{Nil}.\ \text{Nil} \\ \text{Cons}\ x\ y.\ \text{Cons}\ (f\ (g x))\ (f'\ (g' y)) \end{cases}$$

$$= \lambda \begin{cases} \text{Nil}.\ \text{Nil} \\ \text{Cons}\ x\ y.\ \text{Cons}\ ((f \circ g)\ x)\ ((f' \circ g')\ y) \end{cases}$$

$$= \text{bimap}\left((f, f') \circ (g, g')\right) ✓$$

F3:

$$\text{bimap } id_a\ id_b = \lambda x. \begin{cases} Nil.\ Nil \\ Cons\ x\ y.\ \ Cons\ (id_a x)\ (id_b y) \end{cases}$$

$$= \lambda x. \begin{cases} Nil.\ Nil \\ Cons\ x\ y.\ \ Cons\ x\ y \end{cases}$$

$$= id_{\text{bimap } ab} \ \checkmark$$

$\square$

(4)

(i) Show that $F\ a\ b = K2\ a$

$F\ f\ g = \lambda x.\ K2\ (f\ x)$

is a bifunctor.

F1 is trivial ✓

F2:

$$F\ f\ f' \circ F\ g\ g' = \lambda x.\ K2\ (f\ x) \circ \lambda x.\ K2\ (g\ x)$$

$$= \lambda x.\ K2\ (f\ (K2\ (g\ x)))$$

$$= \lambda x.\ ^{K2}(f \circ g)\ x$$

$$= F\left((f, f') \circ (g, g')\right) \checkmark$$

F3: $F\ id_a\ id_b = \lambda x.\ K2\ (id_a\ x) = \lambda x.\ K2\ x = id_{F\ a\ b}$ ✓

(ii) Show that

$$F\,a\,b = a\,b$$

$$F\,f\,g = \lambda x.\ f\,x$$

is a bifunctor.

The solution is analogous to (4) (i)

(iii) Show that

$$F\,a\,b = a\,b$$

$$F\,f\,g = \lambda x.\ g\,x$$

is a bifunctor.

The solution is analogous to (4) (i).

$\square$