Problem Set 2

IAP 2020 18.S097: Programming with Categories

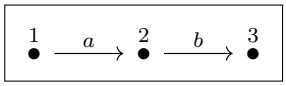
Due Friday January 24

Good academic practice is expected. In particular, cooperation is encouraged, but assignments must be written up alone, and collaborators and resources consulted must be acknowledged.

We suggest that you attempt all problems, but we do not expect all problems to be solved. We expect some to be easier if you have more experience with mathematics, and others if you have more experience with programming. A guideline is that five problems is a good number to write up and submit as homework.

Question 1. *Functors out of* **Set**.

Consider the category **3**, which has three objects and six morphisms, and is depicted as follows:



How many functors are there from **Set** to **3**? Write them down.

Question 2. *Constant functors.*

Let $\mathcal{C}$ and $\mathcal{D}$ be categories. Given any object d in $\mathcal{D}$, we can define the *constant functor* $K_d \colon \mathcal{C} \to \mathcal{D}$ on d. This functor sends *every* of object $\mathcal{C}$ to $d \in \operatorname{Ob} \mathcal{D}$, and *every* morphism of $\mathcal{C}$ to the identity morphism on d.

(a) Take the set $B = \{T, F\}$. Show that the constant functor $K_B \colon \mathbf{Set} \to \mathbf{Set}$ obeys the two functor laws: preservation of composition and preservation of identities.

(b) Implement in Haskell the constant functor on the type `Bool`.

Question 3. *The naturality of the diagonal.*

Let $\mathcal{C}$ and $\mathcal{D}$ be categories, and let $F, G \colon \mathcal{C} \to \mathcal{D}$ be functors. Recall that a natural transformation $\alpha \colon F \Rightarrow G$ consists of a morphism $\alpha_c \colon F(c) \to G(c)$ for each $c \in \operatorname{Ob}(\mathcal{C})$, such that for each $f \colon c_1 \to c_2$ in $\mathcal{C}$ the square:

$$\begin{array}{ccc} F(c_1) & \xrightarrow{F(f)} & F(c_2) \\ {\scriptstyle \alpha_{c_1}}\downarrow & & \downarrow{\scriptstyle \alpha_{c_2}} \\ G(c_1) & \xrightarrow[G(f)]{} & G(c_2) \end{array}$$

commutes in $\mathcal{D}$.

Write $\mathrm{id}_{\mathbf{Set}} \colon \mathbf{Set} \to \mathbf{Set}$ for the identity functor on **Set**, and $\mathrm{Double} \colon \mathbf{Set} \to \mathbf{Set}$ for the functor that sends a set X to the set $X \times X$, and a function $f \colon X \to Y$ to the function $\mathrm{Double}(f) \colon X \times X \to Y \times Y$ that maps (x_1, x_2) to $(f(x_1), f(x_2))$.

(a) For each set X, define the *diagonal function* $\delta_X : X \to X \times X$ to map x to the pair (x, x). Prove that $\delta : \mathrm{id}_{\mathbf{Set}} \Rightarrow \mathrm{Double}$ defines a natural transformation.
(b) Using the universal property of the product, write a polymorphic Haskell function `diag :: a -> (a,a)` implementing this natural transformation.

Question 4. *Uniqueness of universal objects.*

Recall the definitions of terminal object and product from Chapter 3 in the notes.

(a) Show that if t and t' are both terminal objects in a category, then t and t' are isomorphic.
(b) Let a and b be objects of a category. Show that p and p' are both products of a and b, then they are isomorphic.
(c) Discuss the similarities between your two proofs. Could the same idea be used to show that any two initial objects are isomorphic?

Question 5. *Products in preorders.*

Given a set X, a binary operation on X is a function $* : X \times X \to X$ – that is, a way of taking two elements of X and returning a third. This question explores how the product unifies many seemingly quite different binary operations commonly used in math.

(a) Consider the category where the objects are natural numbers and where there is a unique morphism from m to n if m divides n. Given two numbers, for example 42 and 27, what is their product? What is the name of this binary operation?
(b) Consider the category where the objects are subsets of the set $\{a, b, c, d\}$, and where there is a unique morphism from X to Y if X is a subset of Y. Given two subsets, for example $\{a, b, c\}$ and $\{b, d, c\}$, what is their product? What is the name of this binary operation?
(c) Consider the category where the objects are `true` and `false`, and where there is a unique morphism a to b if a implies b. Given two objects, for example `true` and `false`, what is their product? What is the name of this binary operation?

Question 6. *Products in Hask.*

Recall that Haskell has a built-in product (pair) type with constructors written:

```
data (a,b) = (a,b)
```

Implement isomorphisms of the following type signatures by drawing diagrams and translating them into code. Explain why the functions you have constructed are isomorphisms.

(a) `swap :: (a,b) -> (b,a)`
(b) `unit :: a -> ((),a)`
(c) `assoc :: (a,(b,c)) -> ((a,b),c)`

Question 7. *The product of categories.*

Given two categories $\mathcal{C}$ and $\mathcal{D}$, we may construct a new category $\mathcal{C} \times \mathcal{D}$ by taking pairs of objects and morphisms. More precisely:

- The objects of $\mathcal{C} \times \mathcal{D}$ are pairs (c, d), where $c \in \operatorname{Ob} \mathcal{C}$ and $d \in \operatorname{Ob} \mathcal{D}$.
- The morphisms $(c_1, d_1) \to (c_2, d_2)$ are pairs (f, g) where $f \colon c_1 \to c_2$ in $\mathcal{C}$ and $g \colon d_1 \to d_2$ in $\mathcal{D}$.
- Composition is given pointwise: given $(f, g) \colon (c_1, d_1) \to (c_2, d_2)$ and $(h, k) \colon (c_2, d_2) \to (c_3, d_3)$, their composite is $(h \circ f, k \circ g) \colon (c_1, d_1) \to (c_3, d_3)$.
- Similarly, the identity morphisms are given by $(\mathrm{id}_c, \mathrm{id}_d) \colon (c, d) \to (c, d)$.

Recall the category **Cat** whose objects are categories and morphisms are functors. Show that $\mathcal{C} \times \mathcal{D}$ is the product of $\mathcal{C}$ and $\mathcal{D}$ in **Cat**.

Question 8. *Bifunctors.*

A bifunctor is a polymorphic type constructor `F` that takes two variables together with an implementation of a function

```
bimap :: (a -> b) -> (c -> d) -> F a c -> F b d
```

Using the universal property of the coproduct, provide an implementation of `bimap` for the sum type constructor `Either a b`.

Question 9. *Programming with categories.*

Tell a story about what's going on in this course. Is there an example you find interesting or enlightening, or just fun? Have you had any *a-ha!* moments? Share one.

Question 10. *Grade the pset.*

Give a grade to this problem set, taking into account how much you learned, how interesting or fun it was, and how much time you spent on it. Explain your grade.