Monads

Samuel Mimram
samuel.mimram@lix.polytechnique.fr
http://lambdacat.mimram.fr

January 9, 2023

I The exception monad

Given an adjunction $F \dashv G$ between categories $\mathcal{C}$ and $\mathcal{D}$, the composite $T = G \circ F$ is always equipped with a structure of a monad, and the goal of this question is to study an instance of this situation.

We write $\mathbf{Set}_*$ for the category whose objects are *pointed sets*, i.e. pairs (A, a) where A is a set and $a \in A$, and morphisms $f : (A, a) \to (B, b)$ are functions such that $f(a) = b$. Here, the distinguished element of the pointed set will be seen as a particular value indicating an error or an exception.

1. Describe the *forgetful functor* $U : \mathbf{Set}_* \to \mathbf{Set}$.

   Solution. The functor U sends a pointed set (A, a) to the underlying set A and a pointed function to the function itself.

2. Construct a functor $F : \mathbf{Set} \to \mathbf{Set}_*$ which is such that the sets $\mathbf{Set}_*(FA, (B, b))$ and $\mathbf{Set}(A, U(B, b))$ are isomorphic. We will admit that F is left adjoint to U (what would remain to be shown?).

   Solution. We define the functor F as $FA = (A \sqcup \{\star\}, \star)$ and, given $f : A \to B$,

   $$\begin{aligned} Ff : FA &\to FB \\ A \ni a &\mapsto f(a) \\ \star &\mapsto \star \end{aligned}$$

   Let us construct the bijection:

   – given a pointed function $f : A \sqcup \{\star\} \to B$ we obtain a function $\phi(f) : A \to B$ by precomposing by the canonical inclusion $\iota : A \to A \sqcup \{\star\}$:

   $$\phi(f) = f \circ \iota$$

   – given a function $f : A \to B$, we obtain a pointed function $\psi(f) : A \sqcup \{\star\} \to (B, b)$ by

   $$\begin{aligned} \psi(f) : A \sqcup \{\star\} &\to B \\ A \ni a &\mapsto f(a) \\ \star &\mapsto b \end{aligned}$$

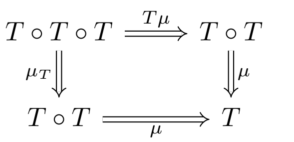
   The two are easily shown to be mutually inverse. Namely, given a pointed function $f : A \sqcup \{\star\} \to B$, we have for $a \in A$

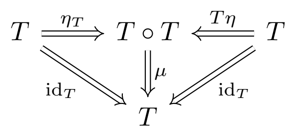
   $$\psi(\phi(f))(a) = \psi(f \circ \iota)(a) = f \circ \iota(a) = f(a) \qquad\qquad \psi(\phi(f))(\star) = b$$

   and thus $\psi(\phi(f)) = f$ because f is pointed. Conversely, given a function $f : A \to B$, we have for $a \in A$,

   $$\phi(\psi(f))(a) = \psi(f) \circ \iota(a) = f(a)$$

3. We recall that a *monad* consists of an endofunctor $T : \mathcal{C} \to \mathcal{C}$ together with two natural transformations $\mu : T \circ T \Rightarrow T$ and $\eta : \mathrm{id}_{\mathcal{C}} \Rightarrow T$ such that the following diagrams commute:





Describe a structure of monad on $T = U \circ F$.

Solution. We have $TA = A \sqcup \{\star\}$. We write $TTA = A \sqcup \{\star, \star'\}$ to distinguish between the two added fresh elements. We define the natural transformations

$$\eta_A : A \to TA \qquad\qquad \mu_A : TTA \to TA$$

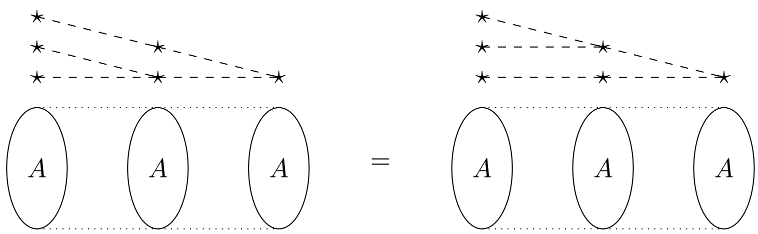
by η_A is the canonical inclusion and

$$\begin{aligned} \mu_A : A \sqcup \{\star, \star'\} &\to A \sqcup \{\star\} \\ A \ni a &\mapsto a \\ \star &\mapsto \star \\ \star' &\mapsto \star \end{aligned}$$

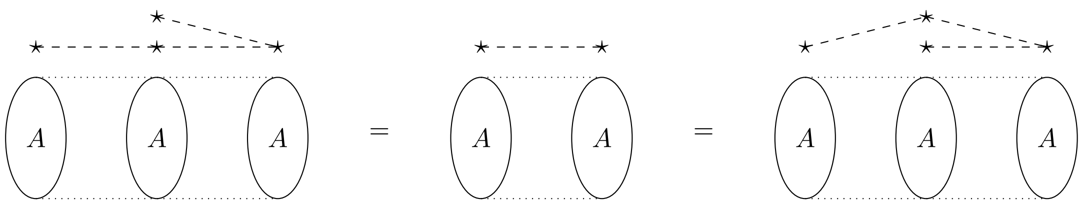
The family $(\eta_A)_{A \in \mathbf{Set}}$ is natural: given a function $f : A \to B$, we have

$$\begin{array}{ccc} A & \xrightarrow{\eta_A} & A \sqcup \{\star\} \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f \sqcup \{\star\}} \\ B & \xrightarrow[\eta_B]{} & B \sqcup \{\star\} \end{array}$$

since both morphisms send an element $a \in A$ to $f(a) \in B \sqcup \{\star\}$, and similarly for $(\mu_A)_{A \in \mathbf{Set}}$. Finally, we can check that the laws for monads are satisfied. Graphically, the associativity law is



and unit laws are



4. Explain how a function $A \to TB$ can be seen as "a function $A \to B$ which might raise an exception".

   Solution. A function $f : A \to B \sqcup \{\star\}$ can be seen as a function $f : A \to B$ which raises an exception when its image is $\star$.

5. Given $f : A \to B$ an OCaml function which might raise an unique exception e and $g : B \to C$ a function which might raise an unique exception e', construct a function corresponding to the composite of f and g which might raise a unique exception e''.

   Solution. We define the function

```
let comp f g x =
  try g (f x)
  with
  | E  -> raise E''
  | E' -> raise E''
```

   whose type is

```
('a -> 'b) -> ('b -> 'c) -> ('a -> 'c)
```

6. Given an arbitrary monad T on a category $\mathcal{C}$, we write $\mathcal{C}_T$ for the category whose objects are the objects of $\mathcal{C}$ and morphisms $f : A \to B$ in $\mathcal{C}_T$ are morphisms $f : A \to TB$ in $\mathcal{C}$, called the *Kleisli category* associated to T. Define composition and identities and show that the axioms of categories are satisfied.

   Solution. Given two morphism $f : A \to B$ and $g : B \to C$ in $\mathcal{C}_T$, i.e. morphisms $f : A \to TB$ and $g : B \to TC$ in $\mathcal{C}$, we define composition as

$$A \xrightarrow{f} TB \xrightarrow{Tg} TTC \xrightarrow{\mu_C} TC$$

   We define the identity $A \to TA$ to be η_A. Given $f : A \to B$ in $\mathcal{C}_T$, we can check that identity is a neutral element on the left ($f \circ \mathrm{id}_A = f$):

$$\begin{array}{ccccc} TA & \xrightarrow{Tf} & TTB & \xrightarrow{\mu_B} & TB \\ {\scriptstyle\eta_A}\uparrow & & {\scriptstyle\eta_{TB}}\uparrow & \nearrow_{\mathrm{id}_{TB}} & \\ A & \xrightarrow[f]{} & TB & & \end{array}$$

   and on the right ($\mathrm{id}_B \circ f = f$):

$$\begin{array}{ccccccc} & & & & TTB & & \\ & & & {\scriptstyle T\eta_B}\nearrow & & \searrow{\scriptstyle\mu_B} & \\ A & \xrightarrow{f} & TB & & \xrightarrow[\mathrm{id}_{TB}]{} & & TB \end{array}$$

   and that composition is associative ($h \circ (g \circ f) = (h \circ g) \circ f$): given $f : A \to TB$, $g : B \to TC$ and $h : C \to TD$, the composite $h \circ (g \circ f)$ is

$$A \xrightarrow{f} TB \xrightarrow{Tg} TTC \xrightarrow{\mu_C} TC \xrightarrow{Th} TTD \xrightarrow{\mu_D} TD$$

   On the other side, the composite is slightly more complicated: we first compute the composite $h \circ g$

$$B \xrightarrow{g} TC \xrightarrow{Th} TTD \xrightarrow{\mu_D} TD$$

   and thus the composite $(h \circ g) \circ f$ is

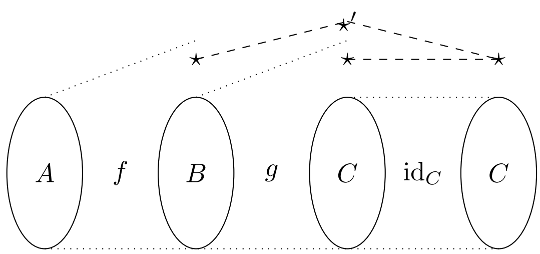
$$A \xrightarrow{f} TB \xrightarrow{Tg} TTC \xrightarrow{TTh} TTTD \xrightarrow{T\mu_D} TTD \xrightarrow{\mu_D} TD$$

   and we have

$$\begin{array}{ccccccccc} A & \xrightarrow{f} & TB & \xrightarrow{Tg} & TTC & \xrightarrow{TTh} & TTTD & \xrightarrow{T\mu_D} & TTD \\ & & & & \downarrow{\scriptstyle\mu_C} & & \downarrow{\scriptstyle\mu_{TD}} & & \downarrow{\scriptstyle\mu_D} \\ & & & & TC & \xrightarrow[Th]{} & TTD & \xrightarrow[\mu_D]{} & TD \end{array}$$

7. Give an explicit description of $\mathbf{Set}_T$ in the case of the above exception monad.

   Solution. Graphically the composition of $f : A \to B \sqcup \{\star\}$ and $g : B \to C \sqcup \{\star\}$ performs as follows:



   which is precisely the expected composition. The category $\mathbf{Set}_T$ can equivalently be described as the category of sets and partial functions.

II More monads

1. A *non-deterministic function* is a function that might return a set of values instead of a single value. How could we could we similarly define a category of non-deterministic functions by a Kleisli construction?

   Solution. For non-determinism, we want to take $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$ which to a set A associates the power set (= the set of subsets).

2. Recall the adjunctions defining a cartesian closed category. What is the associated monad?

   Solution. In a CCC $\mathcal{C}$, we have for every object B the following adjunction:

   $$\mathcal{C} \underset{B\Rightarrow -}{\overset{-\times B}{\underset{\bot}{\rightleftarrows}}} \mathcal{C}$$

   i.e. for every objects A and C, we have a natural bijection

   $$\mathcal{C}(A \times B, C) \simeq \mathcal{C}(A, B \Rightarrow C)$$

   Fixing an object S, the induced monad is $S \Rightarrow (S \times A)$ which is called the "state monad". Namely, TA can be seen as A which takes a state S as input and returns a modified state as output. A morphism $f : A \to B$ in the Kleisli category is a morphism in

   $$\mathcal{C}(A, S \Rightarrow (S \times B))$$

   which, by the adjunction is the same as a morphism in

   $$\mathcal{C}(S \times A, S \times B)$$

   and it can be checked that the composition is the expected one, which "passes on the state".

III Monads in Haskell

Here is an excerpt of `http://www.haskell.org/haskellwiki/Monad`:

```
Monads can be viewed as a standard programming interface to various data or control structures,
which is captured by the Monad class.  All common monads are members of it:

class Monad m where
  (>>=) :: m a -> (a -> m b) -> m b
  return :: a -> m a

In addition to implementing the class functions, all instances of Monad should obey the
following equations:

return a >>= k  =  k a
m >>= return  =  m
m >>= (\x -> k x >>= h)  =  (m >>= k) >>= h
```

1. What does the `Maybe` monad defined below do?

   ```
   data Maybe a = Nothing | Just a

   instance Monad Maybe where
       return         = Just
       Nothing  >>= f = Nothing
       (Just x) >>= f = f x
   ```

   Solution. This is the exception monad.

2. What does the `List` monad defined below do?

```
instance Monad [] where
    m >>= f  = concatMap f m
    return x = [x]
```

Solution. This is the non-determinism monad.

A *Kleisli triple* $(T, \eta, (-)^*)$ on a category $\mathcal{C}$ consists of

- a function $T : \mathrm{Ob}(\mathcal{C}) \to \mathrm{Ob}(\mathcal{C})$,
- a function $\eta_A : A \to TA$ for every object A of $\mathcal{C}$,
- a morphism $f^* : TA \to TB$ for every morphism $f : A \to TB$,

such that for every objects A, B, C and morphisms $f : A \to TB$ and $g : B \to TC$,

$$\eta_A^* = \mathrm{id}_{TA} \qquad f^* \circ \eta_A = f \qquad g^* \circ f^* = (g^* \circ f)^*$$

Our aim is to show that this data amounts to specify a monad on $\mathcal{C}$.

3. Construct the Kleisli category associated to a Kleisli triple.

   Solution. We construct the category $\mathcal{C}_T$ whose objects are the same as those of $\mathcal{C}$ and morphisms $f : A \to B$ in $\mathcal{C}_T$ are morphisms $f : A \to TB$ in $\mathcal{C}$. Identities are given by η. The composition of $f : A \to TB$ and $g : B \to TC$ is
   $$g^* \circ f$$
   We can check that composition is associative:
   $$(h^* \circ g)^* \circ f = h^* \circ g^* \circ f$$
   and admits identities as neutral elements:
   $$\eta_B^* \circ f = \mathrm{id}_{TB} \circ f = f \qquad f^* \circ \eta_A = f$$

4. Show that every Kleisli triple induces a monad.

   Solution. Suppose given a triple $(T, \eta, (-)^*)$, we extend T as a functor by defining, for every morphism $f : A \to B$,
   $$Tf = (\eta_B \circ f)^*$$
   This is indeed a functor since, given $g : B \to C$, we have
   $$Tg \circ Tf = (\eta_C \circ g)^* \circ (\eta_B \circ f)^* = ((\eta_C \circ g)^* \circ \eta_B \circ f)^* = (\eta_C \circ g \circ f)^* = T(g \circ f)$$
   and
   $$T\mathrm{id}_A = (\eta_A \circ \mathrm{id}_A)^* = \eta_A^* = \mathrm{id}_{TA}$$
   We take η as unit of the monad and define the multiplication by
   $$\mu_A = \mathrm{id}_{TA}^*$$
   The family $(\eta_A)_{A \in \mathcal{C}}$ is natural, i.e.
   $$\begin{array}{ccc} A & \xrightarrow{\eta_A} & TA \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle Tf} \\ B & \xrightarrow[\eta_B]{} & TB \end{array}$$
   since, for $f : A \to B$, we have
   $$Tf \circ \eta_A = (\eta_B \circ f)^* \circ \eta_A = \eta_B \circ f$$
   and similarly for $(\mu_A)_{A \in \mathcal{C}}$,
   $$\begin{array}{ccc} TTA & \xrightarrow{\mu_A} & TA \\ {\scriptstyle TTf}\downarrow & & \downarrow{\scriptstyle Tf} \\ TTB & \xrightarrow[\mu_B]{} & TB \end{array}$$

we have

$$\mu_B \circ TTf = \mathrm{id}_{TB}^* \circ (\eta_{TB} \circ (\eta_B \circ f)^*)^* = (\mathrm{id}_{TB}^* \circ \eta_{TB} \circ (\eta_B \circ f)^*)^* = (\mathrm{id}_{TB} \circ (\eta_B \circ f)^*)^* = (\eta_B \circ f)^{**}$$

and on the other side

$$Tf \circ \mu_A = (\eta_B \circ f)^* \circ \mathrm{id}_{TA}^* = ((\eta_B \circ f)^* \circ \mathrm{id}_{TA})^* = (\eta_B \circ f)^{**}$$

Finally, we can check that the laws for monads are satisfied: we have

$$\begin{array}{ccc} TTTA & \xrightarrow{T\mu_A} & TTA \\ {\scriptstyle \mu_{TA}}\downarrow & & \downarrow{\scriptstyle \mu_A} \\ TTA & \xrightarrow[\mu_A]{} & TA \end{array}$$

since

$$\mu_A \circ T\mu_A = \mathrm{id}_{TA}^* \circ (\eta_{TA} \circ \mathrm{id}_{TA}^*)^* = (\mathrm{id}_{TA}^* \circ \eta_{TA} \circ \mathrm{id}_{TA}^*)^* = (\mathrm{id}_{TA} \circ \mathrm{id}_{TA}^*)^* = \mathrm{id}_{TA}^{**}$$

and

$$\mu_A \circ \mu_{TA} = \mathrm{id}_{TA}^* \circ \mathrm{id}_{TTA}^* = (\mathrm{id}_{TA}^* \circ \mathrm{id}_{TTA})^* = \mathrm{id}_{TA}^{**}$$

as well as

$$\begin{array}{ccc} TA & \xrightarrow{\eta_{TA}} & TTA \\ & {\scriptstyle \mathrm{id}_{TA}} \searrow & \downarrow{\scriptstyle \mu_A} \\ & & TA \end{array}$$

since

$$\mu_A \circ \eta_{TA} = \mathrm{id}_{TA}^* \circ \eta_{TA} = \mathrm{id}_{TA}$$

and

$$\begin{array}{ccc} TTA & \xleftarrow{T\eta_A} & TA \\ {\scriptstyle \mu_A}\downarrow & \swarrow {\scriptstyle \mathrm{id}_{TA}} & \\ TA & & \end{array}$$

since

$$\mu_A \circ T\eta_A = \mathrm{id}_{TA}^* \circ (\eta_{TA} \circ \eta_A)^* = (\mathrm{id}_{TA}^* \circ \eta_{TA} \circ \eta_A)^* = (\mathrm{id}_{TA} \circ \eta_A)^* = \eta_A^* = \mathrm{id}_{TA}$$

5. Conversely show that every monad induces a Kleisli triple.

   Solution. Conversely, given a monad, we define for $f : A \to TB$

$$f^* = \mu_B \circ Tf$$

   and we check the laws:

$$\eta_A^* = \mu_A \circ T\eta_A = \mathrm{id}_{TA}$$

   and

$$f^* \circ \eta_A = \mu_B \circ Tf \circ \eta_A = \mu_B \circ \eta_{TB} \circ f = f$$

   and the last equality is similar to the associativity of the Kleisli category above.

We admit that the two transformations are mutually inverse.