(CR15) CATEGORY THEORY FOR COMPUTER SCIENTISTS: HANDOUT FOR LECTURE 10

DECEMBER 21, 2023 — L. T. D. NGUYỄN

Context: there was a campus blockade at ÉNS Lyon on December 21, in protest against an immigration bill inspired by far-right ideology.[1] *This lecture could not be given in person, so here is a write-up of the material that was supposed to be covered – in particular, the first section on monads is necessary to do Exercise 1 of the homework.*

Monads (continued)

Last time, we defined *monads* from adjunctions: an adjunction $L \dashv R$ between $L \colon \mathcal{C} \to \mathcal{D}$ and $R \colon \mathcal{D} \to \mathcal{C}$ induces the monad (M, μ, η) on $\mathcal{C}$ where:

- $M = R \circ L$ is an endofunctor of $\mathcal{C}$
- $\mu = R(\varepsilon_L) : M \circ M \Rightarrow M$ where ε is the *counit* of the adjunction (indeed, $\varepsilon \colon L \circ R \Rightarrow \mathrm{Id}_{\mathcal{D}}$ therefore $R(\varepsilon_L) \colon R \circ L \circ R \circ L \Rightarrow R \circ \mathrm{Id}_{\mathcal{D}} \circ L$)
- $\eta \colon \mathrm{Id}_{\mathcal{C}} \Rightarrow M$ is the *unit* of the adjunction

We also saw examples such as the *list monad* induced by the adjunction $(-)^* \dashv U$ ("free/forgetful adjunction") between the category $\mathbf{Set}$ of sets and the category $\mathbf{Mon}$ of monoids, as well as the covariant powerset monad and the state monad.

These monads are all on $\mathbf{Set}$. A monad on $\mathbf{Set}$ admits a *bind* (`>>=`) operation

$$x \mathbin{\texttt{>>=}} f = \mu_B(M(f)(x)) \in M(B) \text{ for } x \in M(A) \text{ and } f \colon A \to M(B)$$

supporting the computer science point of view on monads as representing computational *effects*. For instance for the list monad, we had:

$$\begin{aligned} \mathrm{List}(A) &= A^* \qquad \text{(lists over } A\text{)} \\ \mathrm{List}(f)([a_1, \dots, a_n]) &= [f(a_1), \dots, f(a_n)] \qquad \text{("map")} \\ \eta_A(a) &= [a] \\ \mu_A([\ell_1, \dots, \ell_n]) &= \ell_1 \cdot \ldots \cdot \ell_n \qquad \text{("flatten")} \end{aligned}$$

therefore $[1, 2] \mathbin{\texttt{>>=}} (x \mapsto [0, 3x]) = [0, 3, 0, 6]$.

(*New material starts here.*) Given two functions $f \colon A \to M(B)$ and $g \colon B \to M(C)$, there is now an obvious way to "plug them together": define

$$f \mathbin{\texttt{>=>}} g = (x \mapsto f(x) \mathbin{\texttt{>>=}} g)$$

For example, $(y \mapsto [1 + y,\, 2]) \mathbin{\texttt{>=>}} (x \mapsto [0, 3x]) = (y \mapsto [0, 3 + 3y, 0, 6])$. Intuitively, this is a kind of "composition of functions with effects" – here, the effect is a sort of non-determinism.[2] This new operator generalizes to arbitrary categories:

Definition. Let $\mathcal{C}$ be a category, $f \in \mathcal{C}(A, M(B))$ and $g \in \mathcal{C}(B, M(C))$. Let (M, μ, η) be a monad on $\mathcal{C}$. We define the *Kleisli composition* of g and f as

$$g \mathbin{\texttt{<=<}} f = \mu_C \circ M(g) \circ f$$

[1] For more explanations in English, see this article: `https://www.theguardian.com/world/2023/dec/20/france-immigration-bill-passed-controversy-emmanuel-macron-marine-le-pen`

[2] Recall that when doing math, it is preferrable to use the powerset monad to model nondeterminism (this will be done in the coalgebra course) but when writing actual programs, the list monad may be more convenient way to keep track of a non-deterministic superposition of possibilities.

Proposition. *When $\mathcal{C} = \mathbf{Set}$, this general definition of Kleisli composition agrees with the set-specific one (using* `>>=`*): g* `<=<` *$f = f$* `>=>` *g.*

Proof idea. Just unfold the definitions. □

Since we have a composition operation, we'd like to use to build a *category*. But to do so, we'd need to check that, for example, Kleisli composition is associative, that is, $(h \mathbin{\texttt{<=<}} g) \mathbin{\texttt{<=<}} f = h \mathbin{\texttt{<=<}} (g \mathbin{\texttt{<=<}} f)$. The conditions that make this work are summed up in the following definition.

Definition. An *internal monoid in* $[\mathcal{C}, \mathcal{C}]$ is a 3-tuple (M, μ, η) where:

- M is an endofunctor of $\mathcal{C}$
- $\mu\colon M \circ M \Rightarrow M$ and $\eta\colon \mathrm{Id}_{\mathcal{C}} \Rightarrow M$ are natural transformations
- μ satisfies the *associativity law*: the diagram below commutes

$$\begin{tikzcd}
M \circ M \circ M \arrow[r, Rightarrow, "\mu_M"] \arrow[d, Rightarrow, "M(\mu)"'] & M \circ M \arrow[d, Rightarrow, "\mu"] \\
M \circ M \arrow[r, Rightarrow, "\mu"'] & M
\end{tikzcd}$$

- μ and η satisfy the *unit laws*: the diagram below commutes

$$\begin{tikzcd}
M \arrow[r, Rightarrow, "M(\eta)"] \arrow[dr, Rightarrow, "\mathrm{id}_M"'] & M \circ M \arrow[d, Rightarrow, "\mu"] & M \arrow[l, Rightarrow, "\eta_M"'] \arrow[dl, Rightarrow, "\mathrm{id}_M"] \\
& M &
\end{tikzcd}$$

  (this is equivalent to asking the left triangle and the right triangle to commute separately – each triangle corresponds to one of the two unit laws).

Remark. Recall that the notation $[\mathcal{C}, \mathcal{C}]$ stands for the category of functors from $\mathcal{C}$ to $\mathcal{C}$ (with natural transformations as morphisms). Let us justify the name "internal monoid" by analogy. By formally replacing $([\mathcal{C}, \mathcal{C}], \circ, \mathrm{Id}_{\mathcal{C}})$ with $(\mathbf{Set}, \times, \{*\})$, we may define an *internal monoid in* **Set** as a set M with $\mu\colon M \times M \to M$ (so μ is a binary operation on M) and $\hat{e}\colon \{*\} \to M$, where the "associativity law" becomes

$$\begin{tikzcd}
M \times M \times M \arrow[r, "\mu \times \mathrm{id}_M"] \arrow[d, "\mathrm{id}_M \times \mu"'] & M \times M \arrow[d, "\mu"] \\
M \times M \arrow[r, "\mu"'] & M
\end{tikzcd}$$

stating that μ is associative, while the unit laws become the fact that $\hat{e}(*)$ is a unit for μ. So an internal monoid in **Set** is a monoid in the usual sense!

This analogy will be made technically rigorous in Olivier Laurent's part of the course using the notion of internal monoid in a *monoidal category*.

Proposition. *Every monad on $\mathcal{C}$ is an internal monoid in $[\mathcal{C}, \mathcal{C}]$.*

Proof idea. The associativity and unit laws are consequences of the *triangle identities* relating the unit and the counit of the adjunction that induces our monad. □

Theorem. *Given an internal monoid (M, μ, η) in $[\mathcal{C}, \mathcal{C}]$, the following data indeed defines a category, called the* Kleisli category *$\mathcal{C}_M$ of M:*

- *the objects of $\mathcal{C}_M$ are the same as of $\mathcal{C}$*
- *$\mathcal{C}_M(A, B) = \mathcal{C}(A, M(B))$ for any two objects A and B*
- *composition is the Kleisli composition* `<=<`
- *the identity for A in $\mathcal{C}_M(A, A) = \mathcal{C}(A, M(A))$ is η_A*

Proof idea. The associativity law on μ is used to show that `<=<` is associative, while f `<=<` $\eta_A = \eta_B$ `<=<` $f = f$ for $f \in \mathcal{C}_M(A,B) = \mathcal{C}(A, M(B))$ thanks to the unit laws concerning μ and η. $\square$

If we see a morphism $f \in \mathcal{C}_M(A,B) = \mathcal{C}(A, M(B))$ as an "effectful morphism" for the effect represented by the monad M – as in our previous examples over **Set** – this means we have a general well-behaved notion of "effectful composition". Now, let us see another remarkable property of Kleisli categories.

Theorem. *Let (M, μ, η) be an internal monoid in $[\mathcal{C}, \mathcal{C}]$. Then we have an adjunction $L \dashv R$ inducing the monad (M, μ, η), where*

$$\begin{aligned} L\colon \mathcal{C} &\to \mathcal{C}_M & R\colon \mathcal{C}_M &\to \mathcal{C} \\ A &\mapsto A & A &\mapsto M(A) \\ f \in \mathcal{C}(A,B) &\mapsto \eta_B \circ f & h \in \mathcal{C}_M(A,B) &\mapsto \mu_B \circ M(f) \end{aligned}$$

Proof idea. Some tedious verifications using the associativity and unit laws show that L and R are indeed functors. For the adjunction, note that

$$\mathcal{C}_M(L(A), B) = \mathcal{C}(A, M(B)) = \mathcal{C}(A, R(B))$$

and, for $f \in \mathcal{C}(A', A)$ and $g \in \mathcal{C}_M(B, B')$, we have

$$\mathcal{C}_M(L(f), g) = (h \in \mathcal{C}(A, M(B)) \mapsto g \texttt{ <=< } h \circ f) = \mathcal{C}(f, R(g))$$

so we have an equality of functors (i.e. equality on both objects and morphisms)

$$\mathcal{C}_M(L(-), -) = \mathcal{C}(-, R(-))$$

and two equal functors are naturally isomorphic. For the unit of the adjunction, it's the image of the identity for $L(A)$ in $\mathcal{C}_M$ via the isomorphism... but this identity in $\mathcal{C}_M$ is η_A and the isomorphism sends it to itself.

Finally, the counit is $\varepsilon_A = \mathrm{id}_{M(A)} \in \mathcal{C}(R(A), R(A)) = \mathcal{C}_M(L(R(A)), A)$, so $R(\varepsilon_{L(A)}) = R(\varepsilon_A) = \mu_A \circ M(\mathrm{id}_{M(A)}) = \mu_A$: the induced monad is (M, μ, η). $\square$

What does this mean? We saw previously that every monad on $\mathcal{C}$ is an internal monoid in $[\mathcal{C}, \mathcal{C}]$, but the above theorem tells us that the converse is also true. Hence **a monad on $\mathcal{C}$ is the same thing as an internal monoid in** $[\mathcal{C}, \mathcal{C}]$. In fact, monads are often *defined* as such internal monoids.[3]

This is useful to check that a monad is a monad without having to guess an adjunction: it suffices to verify that μ and η satisfy the associativity and unit laws. For instance, we may turn the "option data type" into a monad by defining directly μ and η as follows:

$$\begin{aligned} \mu_A\colon \mathrm{Option}(\mathrm{Option}(A)) &\to \mathrm{Option}(A) & \eta_A\colon A &\to \mathrm{Option}(A) \\ \mathsf{Some}(\mathsf{Some}(a)) &\mapsto \mathsf{Some}(a) & a &\mapsto \mathsf{Some}(a) \\ \mathsf{Some}(\mathsf{None}) &\mapsto \mathsf{None} \\ \mathsf{None} &\mapsto \mathsf{None} \end{aligned}$$

The option monad models the computational effect of *partiality*, in other words, of the possibility of *failure*. The Kleisli category $\mathbf{Set}_{\mathrm{Option}}$ is indeed equivalent (and even isomorphic) to the category **PSet** of *partial functions* (cf. Homework 3).

Remark. The Kleisli category of the powerset monad is isomorphic to **Rel**.

[3]Now you can understand this joke: `https://stackoverflow.com/questions/3870088/a-monad-is-just-a-monoid-in-the-category-of-endofunctors-whats-the-problem`

Preservation of (co)products

A key notion of this course has been that of adjoint functor:

- they correspond to global solutions to problems specified by universal properties, which are central to the categorical point of view;
- they induce monads, which are useful for computer science.

As we will see now, another advantage of adjoint functors is that they "interact nicely" with (co)products. This is also the case of representable functors.

Definition. A functor $F\colon \mathcal{C} \to \mathcal{D}$ *preserves products* if, whenever $(B, (\pi_i)_{i\in I})$ is a product of $(A_i)_{i\in I}$ in $\mathcal{C}$, then $(F(B), (F(\pi_i))_{i\in I})$ is a product of $(F(A_i))_{i\in I}$.

Let's start with an intuitive proposition (the proof is left as an exercise).

Proposition. *Suppose that we have a product $(B, (\pi_i)_{i\in I})$ of a family of objects $(A_i)_{i\in I}$, and an isomorphism with another object $f \in \mathrm{Iso}_{\mathcal{C}}(C, B)$. Then $(C, (\pi_i \circ f)_{i\in I})$ is also a product of $(A_i)_{i\in I}$.*

Theorem. *Let $X \in \mathrm{ob}(\mathcal{C})$. The Hom-functor $\mathcal{C}(X, -)$ preserves products.*

Note that in **Set**, this corresponds to $(A \times B)^X \cong A^X \times B^X$, while in $\mathbf{Set}^{\mathrm{op}}$ we get $X^{A+B} \cong X^A \times X^B$.

Proof. Let $(B, (\pi_i)_{i\in I})$ be a product of $(A_i)_{i\in I}$. Using the correspondence between universal morphisms and functor representations, we have a natural isomorphism φ from $\mathcal{C}^{\mathrm{op}}(B, -) = \mathcal{C}(-, B)$ to $\prod_{i\in I} \mathcal{C}(-, A_i)$ such that $\varphi_B(\mathrm{id}_B) = (\pi_i)_{i\in I}$.

In particular, we have an isomorphism

$$\varphi_X\colon \mathcal{C}(X, B) \to \prod_{i\in I} \mathcal{C}(X, A_i)$$

Let $F = \mathcal{C}(X, -)$ be the Hom-functor we are interested in. Then $\prod_{i\in I} \mathcal{C}(X, A_i)$, the set-theoretic product of the sets $F(A_i)$, is a categorical product of this family of objects in **Set** with the usual projections $p_i =$ "take the i-th coordinate". By applying the previous proposition to this product and φ_X, we get another product $(\mathcal{C}(X, B), (p_i \circ \varphi_X)_{i\in I})$ of $(F(A_i))_{i\in I}$ in **Set**.

The theorem that we want to prove is that F preserves products, so our goal is to show that $(F(B), (F(\pi_i))_{i\in I})$ is a product of $(F(A_i))_{i\in I}$. To do so, we just need to show that it coincides with the product we obtained above, that is:

$$(F(B), (F(\pi_i))_{i\in I}) = (\mathcal{C}(X, B), (p_i \circ \varphi_X)_{i\in I})$$

By definition, $F(B) = \mathcal{C}(X, B)$. The equalities between projections require a bit more work. Let $f \in \mathcal{C}(X, B)$. We must check that

$$\forall i \in I,\ \mathcal{C}(X, \pi_i)(f) = p_i(\varphi_X(f))$$

Observe that

$$(\pi_i \circ f)_{i\in I} = (\mathcal{C}(f, A_i)(\pi_i))_{i\in I} = \left(\prod_{i\in I} \mathcal{C}(f, A_i)\right)((\pi_i)_{i\in I})$$

and we knew from the beginning that $\varphi(\mathrm{id}_B) = (\pi_i)_{i\in I}$. By naturality of φ,

$$\left(\prod_{i\in I} \mathcal{C}(f, A_i)\right)(\varphi_B(\mathrm{id}_B)) = (\varphi_X \circ \mathcal{C}(f, B))(\mathrm{id}_B) = \varphi_X(\mathrm{id}_B \circ f) = \varphi_X(f)$$

In conclusion, $(\pi_i \circ f)_{i\in I} = \varphi_X(f)$. Since $\mathcal{C}(X, \pi_i)(f) = \pi_i \circ f$ and p_i takes the i-th coordinate, this is equivalent to what we wanted to check. □

If we combine this theorem with the following property:

Proposition. *If $F \cong G$ and F preserves products, then so does G.*

(proof left as exercise) to derive the following consequence:

Corollary. *Representable functors preserve products.*

Now that we have treated the case of representable functors, let's do the case of adjoints, which is arguably the more important one.

Theorem. *If a functor F is a right adjoint, then it preserves products.*
Dually, if a functor F is a left adjoint, then it preserves coproducts.

(Beware: to *be* a right adjoint is to *have* a left adjoint.) There are many examples, let's talk about them before proving the theorem:

- The forgetful functor $U\colon \mathbf{Mon} \to \mathbf{Set}$ is right adjoint to the "free monoid" functor $(-)^*$. Therefore, it preserves products. Indeed, $\times$ on monoids is built using $\times$ on the underlying sets!
- However, one can find monoids A and B with a coproduct (C, ι_1, ι_2) such that $U(C) \not\cong U(A) + U(B)$ (for example, if neither A nor B is a singleton, then C is infinite even if A and B are finite). Thanks to the theorem, and to the fact that coproducts are unique up to isomorphism, this is a sufficient proof that U is not a left adjoint.
- But since $(-)^*$ is a left adjoint, $(A + B)^*$ is a coproduct of A^* and B^* in $\mathbf{Mon}$. Taking $A = B = \{1\}$ we get (up to isomorphism) the example that $\{a, b\}^*$ is a coproduct of the monoid $\mathbb{N}$ with itself (cf. Homework 2).
- The forgetful preorder-to-set functor *has* both a left adjoint (discrete order) and a right adjoint (full preorder), so it *is* both a right adjoint and a left adjoint. This is consistent with the fact that products of preorders can be built using products of sets, and coproducts of preorders can also be built using coproducts of sets.
- From the adjunction $(-) \times A \dashv \mathbf{Set}(A, -)$, we get
  - $\mathbf{Set}(A, B \times C) \cong \mathbf{Set}(A, B) \times \mathbf{Set}(A, C)$ (also a consequence of the fact that Hom-functors preserve products)
  - $(B+C)\times A \cong (B\times A)+(C\times A)$ – so even though this *distributivity* law of products over coproducts does not work in arbitrary categories, it has a categorical explanation in $\mathbf{Set}$! (And in other *cartesian closed categories*, see Olivier Laurent's part of the course)

Proof that right adjoints preserve products (idea). Let $L \dashv R$ with $R\colon \mathcal{C} \to \mathcal{D}$. Let $(B, (\pi_i)_{i\in I})$ be a product of $(A_i)_{i\in I}$ in $\mathcal{C}$. For any $Y \in \mathrm{ob}(D)$,

$$\mathcal{D}(Y, R(B)) \cong \mathcal{C}(L(Y), B) \cong \prod_{i\in I} \mathcal{C}(L(Y), A_i) \cong \prod_{i\in I} \mathcal{D}(Y, R(A_i))$$

Each $\cong$ comes from a component of some natural isomorphism, so with some care we can get a natural isomorphism between $\mathcal{D}(-, R(B))$ and $\prod_i \mathcal{D}(-, R(A_i))$. This shows that $R(B)$ with a certain choice of projections is a product of $(R(A_i))_{i\in I}$, and with a bit more work one can check that these projections are equal to $R(\pi_i)$. $\square$

A few superficial words about limits and colimits

We can generalize the preservation theorems above to a much larger class of stuff defined by universal properties. This is a good pretext to present an important notion of category theory, mostly for cultural interest (no technical manipulation).

Let $\mathcal{J}$ and $\mathcal{C}$ be two categories. The diagonal functor $\Delta\colon \mathcal{C} \to [\mathcal{J}, \mathcal{C}]$ maps every object $X \in \mathrm{ob}(\mathcal{C})$ to the constant functor $\mathcal{J} \to \mathcal{C}$ (i.e. object of $[\mathcal{J}, \mathcal{C}]$) equal to X – $\Delta(X)(Y) = X$ and $\Delta(X)(f) = \mathrm{id}_X$ – and every morphism to a constant natural transformation – $\Delta(g)_Y = g$ for $g \in \mathcal{C}(X, X')$.

Definition. A *limit* in $\mathcal{C}$ is a universal morphism from the above diagonal functor $\Delta\colon \mathcal{C} \to [\mathcal{J}, \mathcal{C}]$ to some object $F \in [\mathcal{J}, \mathcal{C}]$, for some category $\mathcal{J}$.

What does this mean? Let's take $\mathcal{J}$ to be a very simple category, with only 3 objects and 2 non-identity morphisms, drawn below (left). A functor $F\colon \mathcal{J} \to \mathcal{C}$ corresponds to choosing 3 objects and 2 morphisms in $\mathcal{C}$, as follows (right):

$$\begin{tikzcd}
 & 1 \arrow[d, "*"] \\
2 \arrow[r, "*"'] & 0
\end{tikzcd}
\qquad
\begin{tikzcd}
 & A_1 \arrow[d, "f_1"] \\
A_2 \arrow[r, "f_2"'] & B
\end{tikzcd}$$

$\Delta(X)$ corresponds to the case where the 3 chosen objects are equal to X, and the 2 chosen morphisms are id_X. So a natural transformation $\eta\colon \Delta(X) \Rightarrow F$ is a family $(\eta_X)_{X\in\{0,1,2\}}$ making the diagram below commute:

$$\begin{tikzcd}
 & X \arrow[d, "\mathrm{id}_X"] \arrow[dr, "\eta_1"] & \\
X \arrow[r, "\mathrm{id}_X"'] \arrow[dr, "\eta_2"'] & X \arrow[dr, "\eta_0"] & A_1 \arrow[d, "f_1"] \\
 & A_2 \arrow[r, "f_2"'] & B
\end{tikzcd}
\quad \text{or equivalently} \quad
\begin{tikzcd}
X \arrow[r, "\eta_1"] \arrow[d, "\eta_2"'] \arrow[dr, "\eta_0"] & A_1 \arrow[d, "f_1"] \\
A_2 \arrow[r, "f_2"'] & B
\end{tikzcd}$$

The morphism η_0 is kind of redundant, it's determined by $\eta_0 = f_1 \circ \eta_1 = f_2 \circ \eta_2$, so we can omit it in diagrams. If (X, η) is a universal morphism from Δ to F, its universal property is given by the commutative diagram below; the quantifiers in the statement are "for every (Y, g), there exists a unique h":

$$\begin{tikzcd}
Y \arrow[dr, dashed, "\exists! h"] \arrow[drr, bend left, "g_1"] \arrow[ddr, bend right, "g_2"'] & & \\
 & X \arrow[r, "\eta_1"] \arrow[d, "\eta_2"'] & A_1 \arrow[d, "f_1"] \\
 & A_2 \arrow[r, "f_2"'] & B
\end{tikzcd}$$

This special case of limit is called a *pullback*: (X, η_1, η_2) is the pullback of (f_1, f_2). Two examples:

- In **Set**, if A_1 and A_2 are two subsets of B, and f_1, f_2 are their inclusion maps, then the intersection $A_1 \cap A_2$ with two inclusion maps is a pullback of f_1 and f_2.
- If B is a terminal object and f_i is the unique morphism from A_i to the terminal object, then a pullback is the same as a product of A_1 and A_2.

Pullbacks are the limits whose "shape" $\mathcal{J}$ is a specific category with 3 objects. Other kinds of limits can be obtained by varying $\mathcal{J}$. Dually, *colimits* in $\mathcal{C}$ are limits in $\mathcal{C}^{\mathrm{op}}$, i.e. universal morphisms from some F to $\Delta\colon \mathcal{C} \to [\mathcal{C}, \mathcal{J}]$.

Theorem (not proved here)**.** *Representable functors and right adjoints preserve limits. Left adjoints preserve colimits. (This indeed generalizes what we saw before, since a product is the same as a limit whose shape $\mathcal{J}$ is a category where the only morphisms are identities.)*

Remark. Some textbooks derive the preservation of limits by adjoint functors from the preservation of limits by representable functors (whereas, for products, we gave independent proofs of the two cases). This argument is elegant but requires the *Yoneda lemma*, one of the many important things in category theory that we did not have the time to cover.