Brief notes on category theory

Prakash Panangaden

28th May 2012

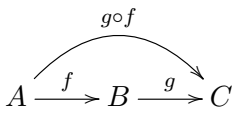
Abstract

These are brief informal notes based on lectures I gave at McGill. There is nothing original about them.

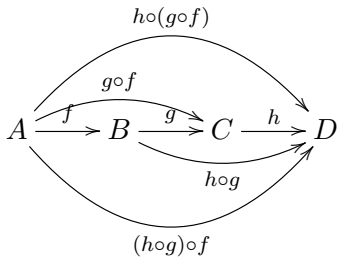
1 Basic definitions

A category consists of two "collections" of things called *objects* and *morphisms* or *arrows* or *maps*. We write $\mathcal{C}$ for a category, $\mathcal{C}_0$ for the objects and $\mathcal{C}_1$ for the morphisms. They satisfy the following conditions:

1. Every morphism f is associated with two objects (which may be the same) called the *domain* and *codomain* of f. One can view a morphism as an arrow from one object to another thus forming a directed graph. We sometimes write $\mathsf{cod}(f)$ and $\mathsf{dom}(f)$ to denote these objects, more often we give them names like A and B. We write $f : A \longrightarrow B$ or $A \xrightarrow{f} B$.
2. Given $A \xrightarrow{f} B$ and $B \xrightarrow{g} C$ we say f and g are *composable* (i.e. $\mathsf{cod}(f) = \mathsf{dom}(g)$). We define a binary operation, written $\circ$, on composable maps so that $g \circ f$ is a morphism: $\mathsf{dom}(g \circ f) = \mathsf{dom}(f)$ and $\mathsf{cod}(g \circ f) = \mathsf{cod}(g)$. In pictures



3. Composition is associative: $(h \circ g) \circ f = h \circ (g \circ f)$ as shown below:



4. For every object $A \in \mathcal{C}_0$ there is a unique morphism $1_A : A \longrightarrow A$ such that for every $f : A \longrightarrow B$, $f \circ 1_A = f$ and for every $g : C \longrightarrow A$, $1_A \circ g = g$.

5. The collection of all the morphisms between A and B is denoted $\mathcal{C}(A, B)$ or $\mathbf{Hom}_{\mathcal{C}}(A, B)$ or just $\mathbf{Hom}(A, B)$.

We usually control the "logical size" of categories by demanding that for all objects A and B the collection $\mathbf{Hom}(A, B)$ is a set; such a category is said to be *locally small*. If the collection of all the objects forms a set we say that the category is *small*. We will almost always be talking about locally small categories. In such categories we call $\mathbf{Hom}(A, B)$ a homset.

Example 1. The collection of *all* sets with functions as the morphisms and ordinary function composition as the operation. This is clearly not small but is locally small. This category is written **Set**.

Example 2. The collection of all sets with *binary relations* as the morphisms and relational composition as the operation. This shows that one can have the same objects in two completely different categories. The morphisms are crucial, more than the objects.

Example 3. The collection of all vector spaces over a given field (say the complex numbers) with linear maps as the morphisms. This category is written **Vect**. This is prototypical of categories of sets with algebraic structure. One can have categories of groups, monoids, rings, semi-rings, boolean algebras, lattices, posets, modules *etc.* in the same vein. One can also define categories where the objects have other kinds of structures, for example, the category **Top** where the objects are topological spaces and the morphisms are *continuous functions*.

Example 4. Any poset (or indeed any preorder) can be viewed as a category where the homsets have at most one element.

Example 5. The objects are the types of a type theory *function types*. A morphism from A to B is a term of type $A \Rightarrow B$. These are very special

categories.
Example 6. The objects are formulas of some logic and the morphisms are *proofs*. Thus, a proof of B under the assumption A is an arrow from A to B. Arrows need not be functions at all! Clearly the proof rules have to have some reasonable properties.

1.1 Some special morphisms

A lot can just be done with just arrows and composition without talking about sets and membership. We define some special types of morphisms: monics or monomorphisms, epics or epimorphisms, isomorphisms, sections and retractions.
Definition 7. A morphism $m : A \longrightarrow B$ is a **monic** if for every pair of morphisms $g, h : C \longrightarrow A$ we have $m \circ h = m \circ g$ implies $h = g$.

In the category **Set** a function is monic iff it is an injection. By flipping the arrows we get a dual concept.
Definition 8. A morphism $m : A \longrightarrow B$ is an **epic** if for every pair of morphisms $g, h : B \longrightarrow C$ we have $h \circ m = g \circ m$ implies $h = g$.

In the category **Set** a function is epic iff it is a surjection.
Definition 9. A morphism $f : A \longrightarrow B$ is called a **section** if it has a *left-inverse*, i.e. there is a morphism $g : B \longrightarrow A$ such that $g \circ f = 1_A$. In this case g has a *right-inverse* and is called a **retraction**.

In any category a section is a monic and a retraction is an epic, but the converses are false.
Definition 10. A morphism is that is both a section and a retraction is called an **isomorphism**. If there is an isomorphism between two objects we say that the objects are **isomorphic**.

In category theory we will usually only care about uniqueness up to isomorphism.

Exercise prove all the statements made without proof so far. Give an example where a morphism is an epic and a monic but is not an isomorphism. [Hint: think about topology.]

2 Duality

In category theory one gets two concepts for the price of one: from one concept one can get another by flipping arrows. We saw this above with epics and monics for example. If we take a category $\mathcal{C}$ we get a new category just by reversing all the arrows; we call this $\mathcal{C}^{op}$. The category $\mathbf{Set}^{op}$ has as objects sets; a morphism from A to B is a function from B to A! This is called the *dual category*. In some cases the dual category is the "same" as the original category, this happens with **Rel** for example. In some cases the dual category is utterly different as happens with **Set**. For example, consider the category of *finite* sets and functions. The opposite of this category is finite *boolean algebras*.

3 Universal and couniversal properties

An absolutely central concept is that of a *universal property* and dually a *couniversal property*. A precise statement will have to wait until we know about functors. Roughly speaking a universal property states that a particular construction subsumes all other like constructions.

3.1 Initial and terminal objects

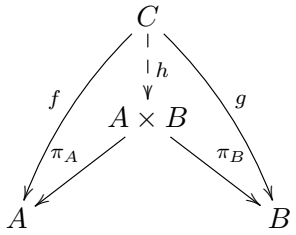
We say that an object I in a category is **initial** if for *every* object A there is a unique morphism from I to A. In **Set** the initial object is the empty set, the unique morphism is the empty function. In the category of groups the initial object is the one-element group.

We say that an object T is **terminal** or **final** if for every object in the category there is a *unique morphism* to T. In **Set** the final object is any one-element set.

There may be many initial or terminal objects in a category but all the initial objects will have to be isomorphic to each other and all terminal objects will be isomorphic to each other.

3.2 Products and coproducts

A (binary) **product** of two objects A, B in a category $\mathcal{C}$ is another object, traditionally written $A \times B$, *together with two morphisms* $\pi_A : A \times B \longrightarrow A, \pi_B : A \times B \longrightarrow B$ such that for *any other object* C and morphisms $f : C \longrightarrow A, g : C \longrightarrow B$ there exists a *unique* morphism, written $\langle f,\ g\rangle : C \longrightarrow A \times B$ such that the diagram below commutes, I have written h for $\langle f,\ g\rangle$:



This contains the crucial ingredient: universality. A product is an object together with morphisms of the appropriate kind. *The crucial property is that if there is any other object claiming to be the product, as C does in the above picture, we can factor C through the real product.* Universality implies that the product, when defined, is unique *up to isomorphism.*

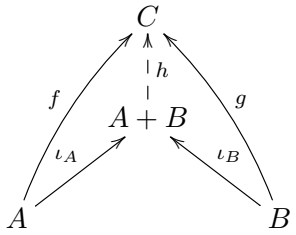
In **Set** this is just the cartesian product. The morphisms π_A and π_B are just the projections. The map $\langle f,\ g\rangle$ is given by $\langle f,\ g\rangle(c \in C) = \langle f(c),\ g(c)\rangle$. Clearly this makes the diagram commute. Is it the only map that makes the diagram commute? Let us call the dashed map h to signify that, for the moment we do not know exactly what it is. Now $h(c)$ has to produce an element of $A \times B$. Because the diagram must commute the first element of this pair has to be $f(c)$ and the second has to be $g(c)$. Thus, the only possible way for h to be defined so that the diagram commutes is for it to be defined as we have done. In a poset viewed as a category the product is the greatest lower bound.

Note that just naming something does not mean that it exists. Some categories have products, some do not. Some have products of some objects but not for all pairs of objects. When we say a category *has products* we mean that every pair of objects has a product. It then follows, by induction, that every *finite* set of objects has a product. A category might or might not have products of infinite collections of objects.

Exercise: Show that **Rel** has products.

Now we reap the benefits of duality. We get a new and interesting concept

by flipping the arrows. A **coproduct** for A and B is an object, $A + B$ and two morphisms $\iota_A : A \to A + B$ and $\iota_B : B \to A + B$ such that if C is any object with morphisms $f : A \to C$ and $g : B \to C$ then there is a *unique* morphism, written $[f, g] : A + B \to C$ such that the diagram below commutes, I have written h for $[f, g]$:



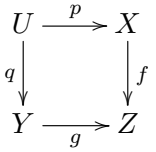
Here the analogue of the universal property is called the couniversal property. In a poset the coproduct corresponds to a least upper bound. In **Set** the coproduct of two sets is the disjoint union. The unique morphism is defined as follows: $[f, g](\iota_A(a)) = f(a)$ and $[f, g](\iota_B(b)) = g(b)$; this looks like a "case" statement and indeed it is.

Exercise Show that **Rel** has coproducts. What observation that we have made before makes this exercise trivial?

3.3 Pullbacks and pushouts

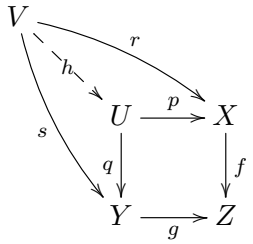
Two very important constructions are pullbacks and pushouts. The pullback is sometimes called the "fibred product" in topology.

The pullback is essentially a restricted version of the product. Given morphisms $f : X \to Z$ and $g : Y \to Z$, a **pullback** is an object U with morphisms $p : U \to X$ and $q : U \to Y$ such that the diagram below commutes, i.e. $f \circ p = g \circ q$



and such that (here comes the universal property), if V is any object with maps $r : V \to X$ and $s : V \to Y$ such that $f \circ r = g \circ s$ (V is "masquerading

as the pullback") then there exists a *unique* morphism $h : V \longrightarrow U$ such that the diagram below commmutes:

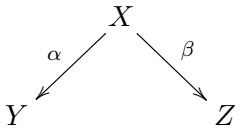


In **Set** pulbacks are constructed as follows. The set U is $\{(x, y) \in X \times Y \mid f(x) = g(y)\}$ and the maps are $p(x, y) = x$ and $q(x, y) = y$. Given another set V and maps r, s as shown in the diagram above we have $h(v) = (r(v), s(v))$. The sceptical reader should check that everything commutes and that this is indeed the only way to define h.

A pair of maps $f : X \longrightarrow Y$ and $g : X \longrightarrow Z$ is called a *span*. Spans are a way of defining relations in a category without talking about elements and sets of ordered pairs. If one has "relations" of this kind how can one compose them? The reader should be able to see that if the category has pullbacks then there is a natural way to compose spans.

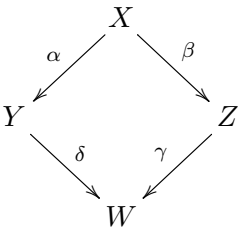
Now for a related concept; this is the *pushout*. It is the colimit of a span, so it is not exactly the dual of the pullback which was the limit of a cospan.

Definition 11. Suppose that we have the following span



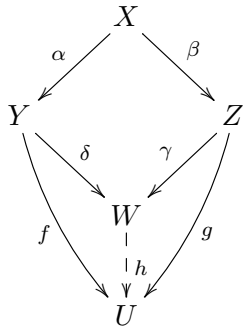
(1)

Then we say that the category has **pushouts** if there is an object W and maps $\delta : Y \longrightarrow W$ and $\gamma : Z \longrightarrow W$ such that the following diagram commutes



(2)

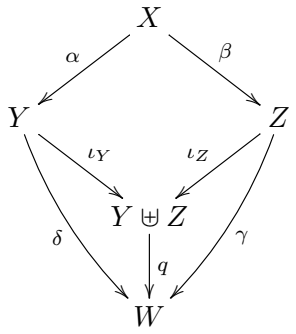
and, furthermore, (here comes *couniversality*) if there is any other commuting square formed by an object U and maps $f : Y \to U$ and $g : Z \to U$ then there is a unique map $h : W \to U$ such that the following diagram commutes:



(3)

Theorem 12. The category **Set** has pushouts.

Proof . We construct W as follows. We form the *disjoint union* $Y \uplus Z$; I will denote elements of this set by y and z without showing the (implicit) canonical injections from Y and Z to $Y \uplus Z$. We define R a binary relation on $Y \uplus Z$ by yRz if $\exists x \in X$ st $y = \alpha(x) \wedge z = \beta(x)$. Now I define an equivalence relation, $\sim$, on $Y \uplus Z$ as the least equivalence relation generated by R. W is obtained as the quotient $(Y \uplus Z)/\sim$. The canonical surjection from $Y \uplus Z$ to W is denoted by q. We define the maps γ and δ by composing the canonical injections with q as shown in the diagram below:



(4)

Clearly the square of Diagram 2 commutes by construction. I need to show that the co-universality property of Diagram 3 holds.

Lemma 13. If we have the span of Diagram 1 and we have *any* commuting square formed with this span and with U, f, g as shown in Diagram 3 then

1. if $y \sim y'$ then $f(y) = f(y')$ and

2. if $z \sim z'$ then $g(z) = g(z')$.

Proof . We proceed by induction on the number of steps required in the transitive closure. The base case is when $y \sim y'$ and there exist $x, x' \in X$ such that $\alpha(x) = y$, $\alpha(x') = y'$ and $\beta(x) = \beta(x') = z$. Then we have

$$f(y) = f(\alpha(x)) = g(\beta(x)) = g(z) = g(\beta(x')) = f(\alpha(x')) = f(y').$$

The same argument applies *mutatis mutandis* to the base case of part 2.

Suppose that $y \sim y'$ because $\exists x, x' \in X$ such that $\alpha(x) = y, \alpha(x') = y', \beta(x) = z, \beta(x') = z'$ and $z \sim z'$. By the inductive hypothesis we assume that $g(z) = g(z')$. Now we have

$$f(y) = f(\alpha(x)) = g(\beta(x)) = g(z) = g(z') = g(\beta(x')) = f(\alpha(x')) = f(y').$$

Again the same argument establishes the inductive case of part 2. ∎

I will now complete the proof of couniversality by describing how to construct the h of Diagram 3. Note that α and β are not necessarily surjections. If there is a $y \in Y$ that is not in the image of α it will not be equivalent to anything except itself; the same statement applies to Z. The elements of W (which are equivalence classes of $\sim$) break into three disjoint subsets: those of the form $\{y\}$ where $y \in Y \setminus \alpha(X)$, those of the form $\{z\}$ where $z \in Z \setminus \beta(X)$ and those of the form $[y] = [z]$ where $y \in Y$ and $z \in Z$. Note that equivalence classes of the third type have to have representatives from both Y and Z. On elements of the first two types we define $h([y]) = f(y)$ and $h([z]) = g(z)$; clearly this is the only way to have the triangles commute. On an element of W of the third type we define $h([y]) = h([z]) = f(y) = g(z)$. We need to verify that this last definition makes sense. We can always choose representatives y, z such that $\exists x \in X$ with $\alpha(x) = y$ and $\beta(x) = z$ so we will have $f(y) = g(z)$ for these representatives. Suppose we had picked any other representative y', then $y \sim y'$ and, by Lemma 13 we have that $f(y) = f(y')$, similarly if we had chosen any other representative from Z. Thus h is uniquely determined and by construction Diagram 3 commutes. ∎

3.4 Limits and colimits

All the constructions shown so far are instances of **limits** (products, pullbacks) or **colimits** (coproducts, pushouts).

Definition 14. Given an arbitrary commuting diagram $\mathcal{D}$ consisting of objects $(X_i \mid i \in I)$ and morphisms $(\alpha_{ij} : X_i \longrightarrow X_j \mid (i,j) \in J \subset I \times I)$[1] in a category $\mathcal{C}$, a **cone over** $\mathcal{D}$ is an object C and a family of maps $\phi_i : C \longrightarrow X_i$ such that for every $(i,j) \in J$ we have $\alpha_{ij} \circ \phi_i = \phi_j$.

In order to define a limit we need to state a universal proeprty.
Definition 15. A **limit** for a diagram $\mathcal{D}$ is a *universal* cone over $\mathcal{D}$. More explicitly, it is a cone (X, ψ_i) such that for any other cone (C, ϕ_i) there is a *unique* map $\gamma : C \longrightarrow X$ such that for every i we have $\phi_i = \psi_i \circ \gamma$.

Dually we can define a cocone and a colimit.

Exercise Define cocone and colimit. Explain what is the diagram for which the initial object is a limit and the terminal object is a colimit.

I will show how to construct two very important examples in **Set**: these are sometimes called *inverse limits* (or projective limits, they are limits in the sense above) and *direct limits* (or inductive limits, they are just colimits in the sense above).

We start by defining some auxiliary concepts.
Definition 16. A **directed set** S is a poset with the order relation $\leq$ such that for every $x, y \in S$ there exists $z \in S$ with $x \leq z$ and $y \leq z$.
Definition 17. A **projective system**, also called an **inverse system**, is a family of objects $\{X_i \mid i \in I\}$, indexed by a directed poset I such that whenever $i \leq j \in I$ there is a morphism $\phi_{ji} : X_j \longrightarrow X_i$ such that

1. $\phi_{ii} = 1_{X_i}$ and
2. if $i \leq j \leq k$ then $\phi_{ki} = \phi ji \circ \phi_{kj}$.

Now we are ready to define a projective limit.
Definition 18. A **projective limit** is a limit for a projective system.
Theorem 19. Countable projective limits exist in **Set**.

Proof . This generalizes the product construction. Assume that we have a projective system as defined above. We construct the set $X = \prod_{i \in I} X_i$. We write a typical member of X as $(x_i)_I$. We choose a subset U as follows

$$U := \{(x_i)_I \mid \forall i \leq j \in I . \phi_{ji}(x_j) = x_i\}.$$

Now we define maps $\pi_j : U \longrightarrow X_j$ by $\pi_j((x_i)_I) = x_j$. It is easy to see that this gives a cone and all the morphisms commute. If V is another set and

[1] There is not necessarily a morphism between every pair of objects.

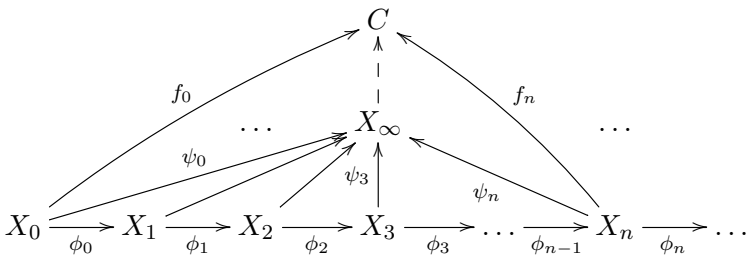
$\psi_i : V \longrightarrow X_i$ are maps such that we have another cone then we construct a map $\beta : V \longrightarrow U$ by

$$\beta(v) = (\psi_i(v))_I.$$

It is easy to see that β is the only map making the entire diagram commute. ∎

Exercise The dual concept is a direct limit or **filtered colimit**. Define it and show that for countable filtered diagrams **Set** has filtered colimits. [Hint: Mimic the construction used for pushouts.]

Here is a diagram for a particular example of a colimit



4 Biproducts in Vect and Rel

In the category **Vect** it turns out that products and co-products are the same object: we call them **biproducts**. The same thing happens in **Rel** as well. In retrospect, this is not a surprise as these are self-dual categories.

First we construct products in **Vect** as follows. Given vector spaces U, V we define the product *as a set* to be $U \times V = \{(u, v) \mid u \in U, v \in V\}$. We make it a vector space by defining addition as

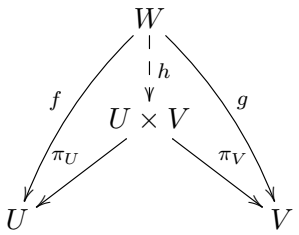
$$(u, v) + (u', v') = (u + u', v + v')$$

and scalar multiplication by

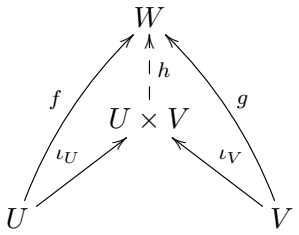
$$r \cdot (u, v) = (r \cdot u, r \cdot v).$$

In the diagram below we define all the maps just as in **Set**; one has to check

that all the maps are linear, but this is obvious in this case.



The vector space $U \times V$ is usually written $U \oplus V$ and called the *direct sum* in linear algebra. We will now check that it is also the *co-product*. In the diagram below



the maps ι_U, ι_V are defined as $\iota_U(u) = (u, 0)$ and $\iota_V(v) = (0, v)$. In **Set** we cannot do this as we do not have a preferred element like 0. Now we define h as follows

$$h(u, v) = f(u) + g(v).$$

It is easy to see that all the maps are linear, the diagram commutes and that h is the only map that makes it commute. We thus have something that is both a product and a co-product, we call it a *biproduct*.

Note that the *tensor product* is *not* a categorical product. There is no canonical map that one can define from $U \otimes V$ to U or to V. This is exactly what leads to the notion of *entanglement*. Tensor products are axiomatized in a completely different way.

5 Functors

Category theory was originally invented to make sense of the notion of natural transformation. First we define functors: they are ways of moving from one catgory to another. One can, in fact, make a category of categories with functors as the morphisms. This category is called **Cat**.

Definition 20. Given categories $\mathcal{C}$ and $\mathcal{D}$ a (covariant) **functor** F from $\mathcal{C}$ to $\mathcal{D}$ is a pair of maps $F : \mathcal{C}_0 \to \mathcal{D}_0$ and $F : \mathcal{C}_1 \to \mathcal{D}_1$ such that

1. Given a morphism $f : A \to B$ in $\mathcal{C}$ there is a morphism $F(f) : F(A) \to F(B)$ in $\mathcal{D}$.
2. $F(1_A) = 1_{F(A)}$.
3. Given $A \xrightarrow{f} B$ and $B \xrightarrow{g} C$ in $\mathcal{C}$ we have

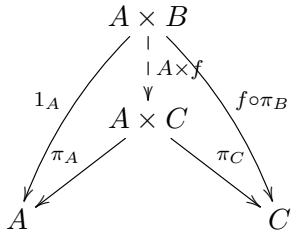
$$F(g) \circ F(f) = F(g \circ f).$$

Now for some examples.

Example 21. Suppose that $\mathcal{C}$ is a category with products. For each object A we define a functor $A \times (\cdot)$ as follows:

$$A \times ((\cdot))(B) = A \times B;$$

this is, of course, not the complete definition. *We must specify what happens to arrows.* Given $f : B \to C$ we have to define a morphism written $A \times f : A \times B \to A \times C$. How do we do it without talking about elements? We use universality as shown below.



We need to check that the conditions for a functor are met of course, but it should be clear that this will turn out to be the case after some simple diagram chasing.

A functor defines a function from homests to homsets; these functions may or may not enjoy standard set-theoretic properties.

Definition 22. We say that a functor $F : \mathcal{C} \to \mathcal{D}$ is *faithful* if the map from $\mathcal{C}(A, B)$ to $\mathcal{D}(F(A), F(B))$ is injective. We say that the functor is *full* if the function is surjective.

The next example is fundamental.

Example 23. Given any category $\mathcal{C}$ there are functors from $\mathcal{C}$ to **Set** for every object A in $\mathcal{C}$. We define $\mathbf{Hom}(A, \cdot) : \mathcal{C} \longrightarrow \mathbf{Set}$ by

$$\mathbf{Hom}(A, \cdot)(B) = \mathbf{Hom}(A, B) \text{ and } \mathbf{Hom}(A, \cdot)(B \xrightarrow{f} C) = \lambda g \in \mathbf{Hom}(A, B).f \circ g.$$

Note that type of $\mathbf{Hom}(A, \cdot) : \mathbf{Hom}(A, B) \longrightarrow \mathbf{Hom}(A, C)$. This is a **Set**-valued functor. As usual there are a number of things to verify, which I have omitted.

What if we looked at arrows *into* an object A? Let us try it. On objects we could still say $\mathbf{Hom}(\cdot, A)(B) = \mathbf{Hom}(B, A)$ but we have a problem with the arrow part of the functor. If we have $f : B \longrightarrow C$ there is no way we can use this to construct a function from $\mathbf{Hom}(B, A)$ to $\mathbf{Hom}(C, A)$; we cannot compose an arrow from B to C with an arrow from B to A. We can, however, compose an arrow from B to C with an arrow from C to A! This leads to a concept dual to that of functor; we call them *contravariant* functors.

Definition 24. Given categories $\mathcal{C}$ and $\mathcal{D}$ a **contravariant functor** F from $\mathcal{C}$ to $\mathcal{D}$ is a pair of maps $F : \mathcal{C}_0 \longrightarrow \mathcal{D}_0$ and $F : \mathcal{C}_1 \longrightarrow \mathcal{D}_1$ such that

1. Given a morphism $f : A \longrightarrow B$ in $\mathcal{C}$ there is a morphism $F(f) : F(B) \longrightarrow F(A)$ in $\mathcal{D}$.
2. $F(1_A) = 1_{F(A)}$.
3. Given $A \xrightarrow{f} B$ and $B \xrightarrow{g} C$ in $\mathcal{C}$ we have
$$F(f) \circ F(g) = F(g \circ f).$$

We could have simply said that a contravariant functor from $\mathcal{C}$ to $\mathcal{D}$ is an ordinary functor from $\mathcal{C}^{op}$ to $\mathcal{D}$.

Example 25. The *set*-functor $\mathbf{Hom}(\cdot, A)$ is a contravariant functor from $\mathcal{C}$ or a functor from $\mathcal{C}^{op}$ to **Set**.

Example 26. We define an interesting contravariant functor from **Vect** to itself. We fix some arbitrary field of scalars k. Let V be a vector space. We define V^* to be the *dual space* of V, i.e. the space of linear maps from V to k. We define a contravariant functor from **Vect** to itself by mapping V to V^*. What is the arrow part of this functor? Given a linear map $\lambda : U \longrightarrow V$ we need to define a linear map $\lambda^* : V^* \longrightarrow U^*$; note the reversal of direction. Let ϕ be an element of V^*, we want $\lambda^*(\phi)$ to be an element of U^*, i.e. it should assign an element of k to any $u \in U$. The types force our hand: $\lambda^*(\phi)(u) = \phi(\lambda(u))$. It is easy to verify all the conditions for a functor.

Example 27. Here is an interesting functor from **Set** to itself. Given a set X we define $\mathcal{P}(X)$ to be the power set of X. On arrows $f : X \to Y$ we define $\mathcal{P}(f) : \mathcal{P}(X) \to \mathcal{P}(Y)$ by $\mathcal{P}(A \subseteq X) := \{f(x) \mid x \in A\}$. We can also define a contravariant power set functor. We call it $\mathcal{Q}$. It is the same as $\mathcal{P}$ on objects but on arrows we define it by

$$\mathcal{Q}(f : X \to Y)(B \subseteq Y) := f^{-1}(B).$$

Example 28. The next example is a little long but it provides one of the fundamental examples of a functor: the fundamental group of a topological space.

For convenience we work with *pointed* topological spaces, that is a topological space X with a distinguished point x_0. A morphism between two pointed spaces $f : (X, x_0) \to (Y, y_0)$ is a *continuous* function f from X to Y such that $f(x_0) = y_0$. Pointed spaces and maps form a category that we call $\mathbf{Top}_\bullet$. We will define a functor π_1 from $\mathbf{Top}_\bullet$ to **Grp**, the category of groups. This functor will measure the "obstructions" to deforming loops in a space.

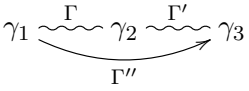
A *path* in X is simple a continuous map $\gamma : [0, 1] \to X$. Note that according to this definition the parametrization matters; a path is not just the image of $[0, 1]$ under γ. A *loop based at* x_0 is a path γ such that $\gamma(0) = x_0 = \gamma(1)$.

Fundamental group detect "holes" by deforming the loop. If there are no holes any loop based at x_0 can be deformed into any other loop based at x_0. In fact the x_0 is not important if we assume that the space is path connected; i.e. every two points can be joined by a path.

We need to define what is meant by deforming a loop. We will define an equivalence relation called *homotopy* to formalize this. We assume that we have two loops based at x_0 i.e. $\gamma_1(0) = \gamma_2(0) = \gamma_1(1) = \gamma_2(1) = x_0$. We write $\gamma_1 \sim \gamma_2$ if $\exists \Gamma : [0, 1] \times [0, 1] \to X$ s.t. Γ is continuous and

$$\Gamma(s, 0) = \gamma_1(s) \qquad \Gamma(s, 1) = \gamma_2(s) \qquad \forall t.\Gamma(0, t) = x_0 = \Gamma(1, t).$$

We refer to the Γ as a homotopy from γ_1 to γ_2 or a deformation from γ_1 to γ_2.

It is clear that this relation is reflexive and symmetric. To show transitivity we assume that we have homotopies, Γ and Γ' as shown: 

and we have to construct a homotopy Γ''. We proceed as follows:

$$\Gamma'' = \begin{cases} \Gamma(s, 2t) & \text{if } 0 \leq t \leq \frac{1}{2} \\ \Gamma'(s, 2t-1) & \text{if } \frac{1}{2} < t \leq 1 \end{cases}$$

Now that we have an equivalence relation we can work with the equivalence classes; of course we will generally work with representatives and check[2] that the result is well defined and independent of the choice of representatives. We are going to make the set of equivalence classes – clearly a much smaller set than the set of loops – into a group. We define a product on equivalence classes

$$[\gamma_1] \times [\gamma_2] = [\gamma_1 \times \gamma_2]$$

by

$$\gamma_1 \times \gamma_2 = \begin{cases} \gamma_1(2t) & \text{if } 0 \leq t \leq \frac{1}{2} \\ \gamma_2(2t) & \text{if } \frac{1}{2} \leq t \leq 1. \end{cases}$$

We define the identity as the loop that just stays at x_0. Inverse just reverses the direction of the loop: $\gamma^{-1}(t) = \gamma(1-t)$. It is straightforward to check that this definition does not depend on the choice of representatives and that the group axioms are satisfied. We have thus defined a group $\pi_1(X, x_0)$. It is not a functor unless we figure out a suitable transformation on morphisms $f : (X, x_0) \longrightarrow (Y, y_0)$. This is just

$$\pi_1(f)(\gamma) = f \circ \gamma.$$

We claim that $(f \circ \gamma_1) \sim (f \circ \gamma_2)$ so this is a well-defined map on the groups. It is easy but annoying to check that $\pi_1(f)$ is a group homomorphism. A crucial consequence of functoriality is that isomorphisms of objects in $\mathbf{Top}_\bullet$ get mapped to isomorphisms of groups.

If the topological space is path connected then for any x_1 and x_2 the groups $\pi_1(X, x_1)$ and $\pi_1(X, x_2)$ are isomorphic. The idea is to take a path σ from x_1 to x_2 and define a map on loops based at x_2 to loops based at x_1 by $\gamma \mapsto \gamma'$ where

$$\gamma'(t) \begin{cases} \sigma(3t) & \text{if } \frac{1}{3} \geq t \geq 0, \\ \gamma(3t-1) & \text{if } \frac{1}{3} \leq t \leq \frac{2}{3} \\ \sigma^{-1}(3t-2) & \text{if } \frac{2}{3} \leq t \leq 1. \end{cases}$$

[2] Or just claim!

This actually defines an isomorphism of groups. Thus we can forget about the point and talk about the fundamental group as a functor from **Top** to **Grp**.

If two topological spaces are homomorphic, their fundamental groups are isomorphic. One non-trivial consequence is Brouwer's fixed-point theorem. Here is a simple instance of it. Consider the closed unit disk in $\mathbb{R}^2$ and its boundary circle S^1. Now π_1 of the disk is the trivial group 0 since every loop can be shrunk to a point. The fundamental group of the boundary is $\mathbb{Z}$; to see this just note that a loop that it wound around the circle n times cannot be unwound, it can be deformed to any other loop that is also wound around n times. The number of times a loop winds around a circle is counted as negative or postive depending on the sense of the winding. This is called the winding number of a map into the circle. Clearly the fundamental groups of the disc and its boundary are not isomorphic. This means that there cannot be a continuous map of the disc onto its boundary. If there were, the two fundamental groups would be the same.

This implies the fixed point theorem. Suppose that f is a continuous function from D to itself and $\forall x : D.f(x) \neq x$, then we can draw a straight line from x, through $f(x)$ and extend it until it hits the circle. However, if that is the case, we have a continuous deformation of the circle onto its boundary and hence their fundamental groups must be the same. This contradiction shows that there is no such f, i.e. every f has a fixed point.

6 Natural transformations

Category theory was invented to formalize the intuitive notion of "naturality." In linear algebra we say that the finite-dimensional space V is isomorphic to its dual space V^* but not in a natural way: the isomorphism depends on a choice of basis. On the other hand there is a "natural" isomorphism from $V \longrightarrow V^{**}$. It is $v \longmapsto \Lambda_v \in V^{**}$ where $\Lambda_v(\sigma) = \sigma(v)$ with $\sigma \in V^*$.

Definition 29. Given two functors $F, G : \mathcal{C} \longrightarrow \mathcal{D}$, a **natural transformation** η from F to G is a correspondence between objects A of $\mathcal{C}$ and morphisms η_A of $\mathcal{D}$ such that if $f : A \longrightarrow B$ is a morphism in $\mathcal{C}$ the following

diagram commutes:

$$\begin{array}{ccc} F(A) & \xrightarrow{\eta_A} & G(A) \\ {\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle G(g)} \\ F(B) & \xrightarrow[\eta_B]{} & G(B) \end{array}$$

Example 30. We return to the example with which we began. Suppose that V is a finite-dimensional vector space over some field k and V^* is the dual space and V^{**} is the double dual. If $\{e_i\}$ is a basis for V we can define $\{\sigma_j\}$ by $\sigma_j(e_i) = \delta_{ij}$, the dual basis of V^*. If σ is any element of V^* we can write $\sigma = \sum_j \sigma(e_j)\sigma_j$ so this is indeed a basis for V^*. Now we can define the iso $e_i \mapsto \sigma_i$, but this is basis dependant. We have already given the natural isomorphism from V to V^{**}. We now show how it fits the definition of a natural transformation between the identity functor and the double-dual functor.

Recall the contravariant functor $(\cdot)^*$ from **Vect** to itself: on arrows it takes $\lambda : U \longrightarrow V$ to $\lambda^* : V^* \longrightarrow U^*$ according to the rule $\lambda^*(\sigma \in V^*) = ((u \in U) \mapsto \sigma(\lambda(u))$. If $\Lambda \in U^{**}$ then $\lambda^{**}(\Lambda)(\sigma \in V^*) = \Lambda(u \mapsto \sigma(\lambda(u))$.

We define the natural transformation $\eta : I \longrightarrow (\cdot)^{**}$ by $\eta_U(u) = \Lambda_u$, where $\Lambda_u = \sigma \mapsto \sigma(u)$. Given a linear map between finite-dimensional vector spaces: $U \xrightarrow{\lambda} V$ we need to show that the following diagram commutes.

$$\begin{array}{ccc} U & \xrightarrow{\eta_U} & U^{**} \\ {\scriptstyle \lambda}\downarrow & & \downarrow{\scriptstyle \lambda^{**}} \\ V & \xrightarrow[\eta_V]{} & V^{**} \end{array}$$

We have $\eta_U(u \in U) = \Lambda_u$ and then

$$\lambda^{**}(\Lambda_u)(\sigma \in V^*) = \Lambda_u(\lambda^*(\sigma)) = \Lambda_u(u' \mapsto \sigma(\lambda(u'))) = \sigma(\lambda(u)).$$

Going the other way we have $\eta_V(\lambda(u))(\sigma) = \sigma(\lambda(u))$ so the diagram commutes. Thus we have a natural transformation.

Example 31. We will construct two functors $F : set \longrightarrow Mon$ and $U : Mon \longrightarrow Set$ and exhibit two common natural transformations. The functor $U : Mon \longrightarrow Set$ is defined by simply taking the underlying set of a monoid and forgetting that there was an operation defined on it. The "U" is supposed to make you think of "underlying." On arrows it takes monoid homomorphisms and views them as plain set theoretic functions.

The functor F is defined by starting with a set, say Σ, and constructing the set of all finite words Σ^*. The binary operation is concatenation of words and the identity element is the empty word ε. In this way we have manufactured a monoid "freely" from Σ. Given a function $f : \Sigma \longrightarrow \Gamma$ in **Set** we define a monoid homomorphism $f^! : \Sigma^* \longrightarrow \Gamma^*$ by $f^!(a_1 \dots a_n) = f(a_1) \dots f(a_n)$ and $f^!(\varepsilon) = \varepsilon$.

Now we have functors $U \circ F$: **Set** $\longrightarrow$ **Set** and $F \circ U$: **Mon** $\longrightarrow$ **Mon** obtained by composing these functors. We will define two natural transformations:

$$FU \xrightarrow{\epsilon} I_{mon} \text{ and } I_{set} \xrightarrow{\eta} UF \ .$$

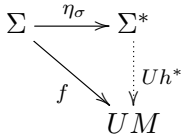
The map η_Σ is defined by $\eta_\Sigma(a) = a$. Naturality is completely obvious in this case. Though obvious, this is a very important example and this natural transformation is often called "inclusion of generators."

To define ϵ_M where $(M, \times, \varepsilon_M)$ is a monoid, we need a monoid homomorphism from M^* to M. The monoid M^* is obtained by first forgetting that M is a monoid and then forming words out of the elements of M. What could be more "natural" than to define

$$\epsilon_M(m_1 m_2 ... m_n) = m_1 \times m_2 \times ... \times m_n.$$

Once again it is completely clear that this is a natural transformation.

In order to prepare for adjunctions and to give another example of a universal property we note the following. The map $\eta_\Sigma : \Sigma \longrightarrow \Sigma^*$ enjoys the following universal property: given any $f : \Sigma \longrightarrow UM$, there exists a unique monoid homomorphism $\Sigma^* \xrightarrow{h^*} M$ such that the diagram



commutes. This homomorphism is given by

$$h^*(x_1 x_2 ... x_n) = f(x_1) \times f(x_2) \times ... \times f(x_n),$$

in other words it is just $\epsilon_M \circ F(f)$. There is a similar couniversal property going the other way[3].

[3]What is it?

Functors and natural transformations form categories in their own right.

Proposition 32. Given categories $\mathcal{C}$ and $\mathcal{D}$, the collection of all functors from $\mathcal{C}$ to $\mathcal{D}$ forms a category, the morphisms are the natural transformations.

This kind of category is called a *functor category* and is written $[\mathcal{C}, \mathcal{D}]$ or, more commonly, $\mathcal{D}^{\mathcal{C}}$. We write $\mathbf{Nat}(F, G)$ for the homsets in this category.

7 Yoneda'a Lemma

A vital theorem about natural transformations is given by what is called Yoneda's Lemma.

Theorem 33 (Yoneda)**.** Let $F : \mathcal{C} \to \mathbf{Set}$ be a functor and let $\mathbf{Hom}(A, \cdot) : \mathcal{C} \to \mathbf{Set}$ be the hom-functor which maps an object B of $\mathcal{C}$ to the homset $\mathbf{Hom}(A, B)$. There is a natural bijection between $\mathbf{Nat}(\mathbf{Hom}(A, \cdot), F)$, the set of natural transformations between $\mathbf{Hom}(A, \cdot)$ and F, and $F(A)$.

Proof . I will just exhibit the bijection. First, given an element a of $F(A)$ we need to associate to it a natural transformation $\eta^a : \mathbf{Hom}(A, \cdot) \to F$. This means to every X in $\mathcal{C}$ we need a function $\eta^a_X : \mathbf{Hom}(A, X) \to F(X)$ and given any morphism $f : X \to Y$ in $\mathcal{C}$ we need the following diagram to commute:

$$\begin{array}{ccc} \mathbf{Hom}(A, X) & \xrightarrow{\eta^a_X} & F(X) \\ {\scriptstyle \mathbf{Hom}(A,f)}\downarrow & & \downarrow{\scriptstyle F(f)} \\ \mathbf{Hom}(A, Y) & \xrightarrow[\eta^a_Y]{} & F(Y) \end{array}$$

The action of the function $\mathbf{Hom}(A, f)$ is given by $\mathbf{Hom}(A, f)(h : A \to X) = f \circ h$ as per the definition of the **Hom**-functor. The actions of the components of the natural transformation are

$$\eta^a_X(h : A \to X) = F(h)(a) \text{ and } \eta^a_Y(g : A \to Y) = F(g)(a).$$

We check commutativity:

$$\eta^a_Y(\mathbf{Hom}(A, f)(h)) = \eta^a_Y(f \circ h) = F(f \circ h)(a) = F(f)(F(h)(a)) = F(f)(\eta^a_X(h)).$$

Thus for every element a of $F(A)$ we have a natural transformation η^a.

Given a natural transformation $\eta : \mathbf{Hom}(A, \cdot) \longrightarrow F$ we get an element $\eta_A(id_A)$ of $F(A)$. Using naturality we have that the following square commutes for any $f : A \longrightarrow X$:

$$\begin{array}{ccc} \mathbf{Hom}(A,A) & \xrightarrow{\eta_A} & F(A) \\ {\scriptstyle \mathbf{Hom}(A,f)}\downarrow & & \downarrow{\scriptstyle F(f)} \\ \mathbf{Hom}(A,X) & \xrightarrow[\eta_X]{} & F(X) \end{array}$$

which says that

$$\eta_X(f) = \eta_X(f \circ id_A) = \eta_X(\mathbf{Hom}(A, f)(id_A)) = F(f)(\eta_A(id_A)).$$

In other words the action of the natural transformation η is completely determined by $\eta_A(id_A)$. It is easy to see that these correspondences are inverses of each other. ∎

Recall the definition of full and faithful functors. The Yoneda Lemma gives as a corollary the following embedding of any category into a particular functor category.

Corollary 34. The functor $\mathcal{Y} : \mathcal{C} \longrightarrow \mathbf{Set}^{\mathcal{C}^{op}}$ given by $\mathcal{Y}(A) = \mathbf{Hom}(\cdot, A)$ is a full and faithful embedding.

Proof . The functor $\mathcal{Y}$ defines a map, call it Y, from $\mathcal{C}(A, A')$ to $\mathbf{Nat}(\mathbf{Hom}(\cdot, A), \mathbf{Hom}(\cdot, A'))$. Let $f \in \mathbf{Hom}(A, A')$, then $Y(f)$ is a natural transformation from $\mathbf{Hom}(\cdot, A)$ to $\mathbf{Hom}(\cdot, A')$. Explicitly, given B, an object of $\mathcal{C}$, we have the function $Y(f)_B : \mathbf{Hom}(B, A) \longrightarrow \mathbf{Hom}(B, A')$ given by $g \mapsto f \circ g$. Clearly if $f = f'$ then $Y(f) = Y(f')$. If $Y(f) = Y(f')$ we can use A for B and the identity for g to conclude that $f = f'$. Hence Y is injective. The proof of the Yoneda Lemma makes it clear that it is surjective. ∎

8 Adjoint Functors

We warm up with an example from posets. Recall that a poset is a category in which every homset is either empty or has one element. A number of ideas can be illustrated by first looking at the poset version. A functor corresponds to a monotone function.

Definition 35. A **Galois connection** between two posets $(P, \leq_P)$ and $(Q, \leq_Q)$ is a pair of monotone functions $f : P \to Q$ and $g : Q \to P$ such that for every $x \in P$ and $y \in Q$ we have $f(x) \leq_Q y$ if and only if $x \leq_P g(y)$.

This says one can carry out the comparisons in either P or in Q. It follows from this definition that $x \leq g(f(x))$. To see this, note that $f(x) \leq f(x)$, from the definition this immediately gives $x \leq g(f(x))$. Similarly, $f(g(y)) \leq y$. Taking $y = f(x)$ in this last inequality we get $f(g(f(x))) \leq f(x)$, applying g to both sides we get $g(f(g(f(x)))) \leq g(f(x))$, but we also have $g(f(x)) \leq g(f(g(f(x))))$, hence we get $g \circ f \circ g \circ f = g \circ f$. Thus $g \circ f$ is monotone, increasing and idempotent; such a function is called a closure operator. Similarly $f \circ g$ is montone *decreasing* and idempotent, such a function is called a *kernel operator*.

When we boost all this to the categorical level we will need to compare functors; we do this with natural transformations.

Definition 36. Let $\mathcal{C}$ and $\mathcal{D}$ be categories. An **adjunction** between $\mathcal{C}$ and $\mathcal{D}$ is a pair of functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$ such that for every object X of $\mathcal{C}$ and Y of $\mathcal{D}$ there is a natural isomorphism between the homsets $\mathcal{C}(X, GY)$ and $\mathcal{D}(FX, Y)$.

It is important to visualize this properly. The following picture is useful.

$$X \longrightarrow GY \qquad \text{of } \mathcal{C}$$

$$FX \longrightarrow Y \qquad \text{of } \mathcal{D}$$

It shows the homsets that are isomorphic. In order to show that one has an adjoint pair one has to construct a bijection between these two homsets. The F appears on the left, it is called a *left adjoint* and G is called a *right adjoint*. We say that F *is* a left adjoint and that it *has* a right adjoint.

The naturality condition can be made more explicit, but in order to do so we need to introduce some notation for composing natural transformations with functors.

Suppose that we have the following diagram of categories and functors

$$\mathcal{A} \xrightarrow{U} \mathcal{C} \underset{G}{\overset{F}{\Downarrow \eta}} \mathcal{D} \xrightarrow[V]{} \mathcal{B}$$

with a natural transformation η from F to G. One has composed functors $U \circ G$ and $U \circ F$ with a natural transformation written ηU from $U \circ F$ to $U \circ G$. Given an object A of $\mathcal{A}$ we have UA as object of $\mathcal{C}$ and a morphism $\eta_{UA} : FUA \longrightarrow GUA$ in $\mathcal{D}$; we define this morphism to be ηU_A. Similarly we have a natural transformation, written $V\eta$, from $V \circ F$ to $V \circ G$. Given an object $C \in \mathcal{C}$ we have a morphism $\eta_C : FC \longrightarrow GC$. Applying the functor V to this morphism gives a morphism $V\eta_C : V(FC) \longrightarrow V(GC)$ in $\mathcal{B}$; we define this to be $(V\eta)_C$.

We now give an equivalent definition of an adjunction with the natural transformations shown explicitly.

Definition 37. An **adjunction** between categories $\mathcal{C}$ and $\mathcal{D}$ is a pair of functors $F : \mathcal{C} \longrightarrow \mathcal{D}$ and $G : \mathcal{D} \longrightarrow \mathcal{C}$ together with a pair of natural transformations $\eta : \mathbf{1}_{\mathcal{C}} \longrightarrow GF$ and $\varepsilon : FG \longrightarrow \mathbf{1}_{\mathcal{D}}$ such that the following equations hold

$$(G\varepsilon) \circ (\eta G) = 1_G \text{ and } (\varepsilon F) \circ (F\eta) = 1_F.$$

Here 1_G and 1_F are the identity natural transformations.

Finally, there is a third way of seeing what an adjunction is in terms of universal properties. Suppose that we have a functor G from $\mathcal{D}$ to $\mathcal{C}$. Then for every object A of $\mathcal{C}$ there is a universal arrow from A to an object of the form GX, this universal arrow is precisely the arrow $\eta_A : A \longrightarrow GFA$. What is the universal property? Given any other object B of $\mathcal{D}$ and an arrow $f : A \longrightarrow GB$ there is a unique arrow $\hat{f}$ from $F(A)toB$ such that f can be written as $f = G(\hat{f}) \circ \eta_A$. The correspondence between f and $\hat{f}$ is precisely the natural correspondence between homsets that defines the adjunction.

We have already seen examples of this concept. The functors F : **Set** $\longrightarrow$ **Mon** and U : **Mon** $\longrightarrow$ **Set** are an adjoint pair with the ε and η defined there being exactly the natural transformations referred to in the definition of an adjunction. Here F is the left adjoint and U is the right adjoint.

There are a number of important facts about adjunctions. First of all we can compose them in the obvious way to get adjunctions. Second, if there is an adjunction then one partner of the pair determines the other. Thirdly and most important: right adjoints preserve limits, left adjoints preserve colimits. The adjoint functor theorem gives conditions under which left and right adjoints exist. Ignoring technical size conditions: a functor that preserves limits has a left adjoint and a functor that preserves colimits has a right adjoint.

9 Cartesian Closed Categories

10 The Curry-Howard-Lambek Isomorphism

11 Induction and Coinduction

The slogan is "initiality is induction." In order to make sense of this we need the categorical analogue of basic fixed point theory from posets.

Definition 38. Given a functor $F : \mathcal{C} \longrightarrow \mathcal{C}$ we define a category of F-algebras by taking the objects to be morphisms $\alpha : FA \longrightarrow A$. A morphism from $\alpha : FA \longrightarrow B$ to $\beta : FB \longrightarrow B$ is a $\mathcal{C}$ morphism f such that the following diagram commutes:

$$\begin{array}{ccc} FA & \xrightarrow{\alpha} & A \\ {\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle f} \\ FB & \xrightarrow[\beta]{} & B. \end{array}$$

Why are these things called "algebras?" If, for example, $FX) = X \times X$ then an F-algebra gives a binary operation on X so F-algebras generalize the notion of set equipped with operations.

Theorem 39 (Lambek's Lemma)**.** An *initial* F-algebra $\iota : FI \longrightarrow I$, if it exists, defines an *isomorphism* between I and FI.

Proof .

$$\begin{array}{ccc} FI & \xrightarrow{\iota} & I \\ {\scriptstyle F(\theta)}\downarrow & & \downarrow{\scriptstyle \theta} \\ F^2I & \xrightarrow{F(\iota)} & FI \\ {\scriptstyle F(\iota)}\downarrow & & \downarrow{\scriptstyle \iota} \\ FI & \xrightarrow[\iota]{} & I \end{array}$$

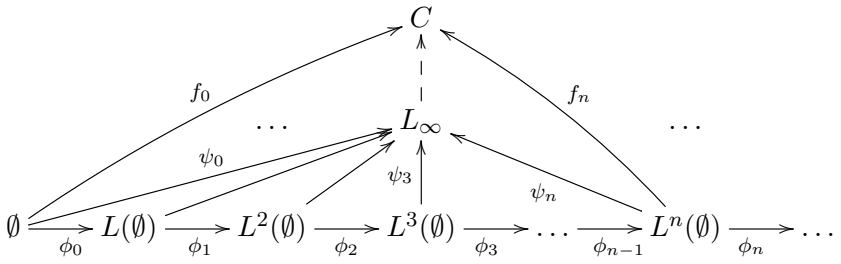
In the diagram above we need to define a map θ as shown and prove that ι and θ are inverses. Since F is a functor we can apply it to ι to obtain the second row of the picture, which is an F-algebra. Since ι is an *initial* F-algebra there is a unique arrow θ from I to FI such that the upper square of the diagram commutes. The lower square commutes by inspection. Composing the vertical maps we obtain that $\iota \circ \theta$ is a morphism in the category of F-algebras from ι to itself, but since ι is initial there can be

only one morphism from ι to itself, namely the identity morphism. Thus we have $\iota \circ \theta = id_I$. From functoriality and the fact that the upper square commutes we have $F(\iota \circ \theta) = F(\iota) \circ F(\theta) = \theta \circ \iota$. But $\iota \circ \theta$ is id_I so $F(\iota \circ \theta) = F(id_I) = id_{FI}$; thus $\theta \circ \iota = id_{FI}$. We have shown that ι and θ are inverses. ∎

What does this buy us? First of all let us explore the connection between having an initial F-algebra and inductive definitions. I claim that the initiality condition *automatically* gives a scheme for inductive definitions on the initial F-algebra. We will consider functors from **Set** to **Set**. We define a functor called L (for List)as follows; we fix a set At of atoms:

$$L(X) = At \times X \uplus \{NIL\}, \quad L(f : X \to Y)(NIL) = NIL,\ L(f)(a, x) = (a, f(x)).$$

First we will construct the initial L-algebra by forming a suitable colimit, one that we have seen before.

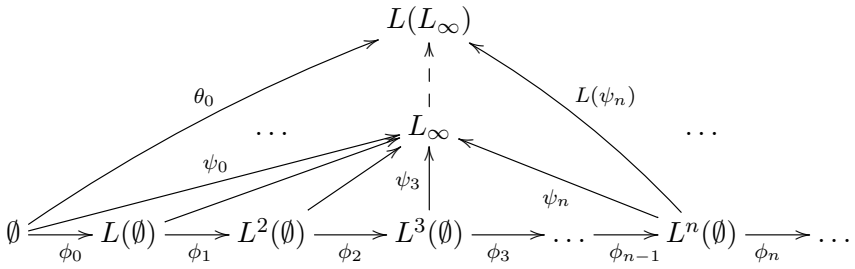


The set $L(\emptyset)$ contains just the single element NIL, which we assume to be distinct from any of the atoms. If we take atoms to be the set $\{a, b\}$, then the set $L^2(\emptyset) = \{NIL, (a, NIL), (b, NIL)\}$ and the map from $L(\emptyset)$ to $L^2(\emptyset)$ is just the inclusion map. Similarly the set

$$L^3(\emptyset) = \{NIL, (a, NIL), (b, NIL), (a, (a, NIL)), (a, (b, NIL)), (b, (a, NIL)), (b, (b, NIL))\}.$$

We usually write these in the more pleasant display forms $\{NIL, a, b, aa, ab, ba, bb\}$ but mathematically (and in the implementation of lists!) they are nested pairs as I have shown. The maps in the infinite horizontal line are inclusions and, given the definition of colimit, the colimit object is then just the union of all these sets. Thus L_∞ is the collection of finite lists build up inductively.

Now consider the object $L(L_\infty)$. By applying the functor L to the diagram formed by the above picture, excluding C and the arrows to C, we get



Thus we get a cocone over the same diagram and by couniversality we get a map from $L_\infty \longrightarrow L(L_\infty)$. This is not the arrow we need to make L_∞ into the carrier of an L-algebra.

Suppose that L has a special property: it commutes with colimits. This means that if one constructs a colimiting diagram and X is the colimit object then $L(X)$ is the colimit of the diagram obtained by applying L to every member of the original diagram. Our functor L has this property, more on this later. For now, just assume that it does.

Definition 40. a functor F is said to be **ω-cocontinuous** if it commutes with the formation of any countable colimit.

One needs general theorems about when functors have this property; we will not prove them here.

Now the diagram obtained by applying L to the colimit diagram

$$\emptyset \xrightarrow[\phi_0]{} L(\emptyset) \xrightarrow[\phi_1]{} L^2(\emptyset) \xrightarrow[\phi_2]{} L^3(\emptyset) \xrightarrow[\phi_3]{} \ldots \xrightarrow[\phi_{n-1}]{} L^n(\emptyset) \xrightarrow[\phi_n]{} \ldots$$

is just

$$L(\emptyset) \xrightarrow[\phi_1]{} L^2(\emptyset) \xrightarrow[\phi_2]{} L^3(\emptyset) \xrightarrow[\phi_3]{} \ldots \xrightarrow[\phi_{n-1}]{} L^n(\emptyset) \xrightarrow[\phi_n]{} \ldots$$

but we can always tack on the initial object in front, as the morphism from it is unique, to get exactly the same picture as we had before. Since the formation of colimits commutes with applying L we have that $L(L_\infty)$ is also the colimit, hence by uniqueness of colimits we have $L(L_\infty) \simeq L_\infty$. This isomorphism gives us the map λ we want from $L(L_\infty)$ to L_∞.

There was nothing special about L in this example; any ω-cocontinuous functor F would have been the same. It would have given us an object

F_∞ with an isomorphism between it and $F(F_\infty)$. So let us switch back to speaking about a general such functor. We have to show that the F-algebra we have just constructed is initial. Let $\alpha : FA \longrightarrow A$ be any F-algebra. We can construct a cocone of arrows pointing into A as follows. We have the unique arrow $!_A : 0 \longrightarrow A$ where 0 is the initial object of the category. We apply F and get $F(!_A) : F(0) \longrightarrow F(A)$. Compose this with α to get $\alpha \circ F(!_A) : F(0) \longrightarrow A$. We can continue in this way to get the cocone we want. Now couniversality gives us a unique morphism f from F_∞ to A. Since $F(F_\infty)$ is also a colimit we get a unique morphism γ, from $F(F_\infty)$ to A. Now we have the following picture

$$\begin{array}{ccc} F(F_\infty) & \xrightarrow{\iota} & F_\infty \\ {\scriptstyle F(f)}\downarrow & \overset{\gamma}{\searrow} & \downarrow{\scriptstyle f} \\ F(A) & \xrightarrow[\alpha]{} & A \end{array}$$

and by uniqueness of γ we have

$$f \circ \iota = \gamma = \alpha \circ F(f).$$

In short we have constructed a unique morphism from ι to α.

Finally we want to see how to define functions from F_∞ to other types by induction. For this let us return to our list example. Suppose we have an L-algebra $\alpha : LA \longrightarrow A$. For definiteness we take A to be the natural numbers $\mathbb{N}$ and $\alpha(NIL) = 0$, $\alpha(a, l) = 1 + \alpha(l)$. We then have a unique function, we will call it $\mathsf{len} : L_\infty \longrightarrow \mathbb{N}$ given as follows:

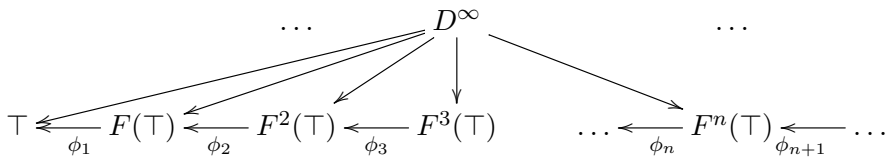
$$\mathsf{len}(NIL) = \alpha(NIL) = 0, \quad \mathsf{len}(a.l) = \alpha(a, l) = 1 + \alpha(l).$$

This is precisely the inductive definition of the length function.

In general the map ι tells you how to put together elements of the colimit (it encodes the constructors), the function f is defined inductively by matching on the image of elements of $F(F_\infty)$ (pattern matching), the map $F(f)$ followed by α gives the inductive definition and initiality promises that this yields a well defined function f.

The amazing thing is that all this can be dualized to give coinduction. We define a *coalgebra* for a functor F to be a morphism $\delta : D \longrightarrow FD$. A morphism of coalgebras is defined in the evident way. Lambek's lemma works with the obvious modification. In other words if we have a final

(terminal) coalgebra we get an isomorphism. In order to build the terminal coalgebra we use the following diagram



where $\top$ is the terminal object and the maps are constructed by using finality to get a unique map from $F(\top)$ to $\top$ and then using functoriality to get the morphisms in the infinite horizontal row. Now we require functors to preserve countable *limits*, such functors are called *continuous*.

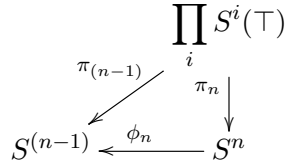
Let us consider an example: we take the cartesian product functor with a fixed set At, we will call this functor S. The terminal object is any one-element set $\{\bullet\}$. $S(\top) = \{a\bullet, b\bullet\}$ and

$$S^2(\top) = \{aa, ab, ba, bb\}$$

where I have stopped writing $\bullet$. The map from $S^(\top) \longrightarrow S^(\top)$ is the following

$$aa \mapsto a,\ ab \mapsto a,\ ba \mapsto b,\ bb \mapsto b.$$

Thus each of the left pointing maps chops off the last element. How is the limit constructed? We take sequences $\sigma \in \prod_i S^i(\top)$ that satisfy the consistency condition



which means that such a σ is an infinite sequence of finite sequences *each of which is an extension of the previous one*. Such a family of finite sequences can be regarded as a single infinite sequence. In short we have constructed the infinite lists! It is fascinating that the colimit gives the finite lists and the limit gives the infinite streams.

The induction principle for finite lists is essentially initiality. What is the dual coinduction concept? If we call the streams that we have just constructed St we have the map $\delta : St \longrightarrow At \times St$. What δ does is tell us how to *split* a stream into a head and a tail; for lists we had constructors for

streams we have destructors. A better notation for δ is $\langle \mathsf{hd}, \mathsf{tl} \rangle$; it takes a stream apart and gives back the head and the tail.

Suppose that we built streams out of another alphabet Σ; this would come with its own similar destructors; let us call them hd' and tl'. Now we can define an S-coalgebra as follows, $\sigma \in St$ and $\sigma' \in \Sigma^\infty$:

$$t : \Sigma^\infty \to At \times \Sigma^\infty,\ t(\sigma') = (f(\mathsf{hd}'(\sigma')), \mathsf{tl}'(\sigma')),$$

where f is some function from Σ to At. The finality property now induces a function from Σ^∞ to St and this will be precisely the stream transducer induced by f.

12 Stream programming examples

This section is based on a meeting with Ryan Kavanagh in my office and was not given as a lecture. Ryan typeset the notes.

We will denote throughout the set of infinite streams of elements of a set S by $S^{\mathbb{N}}$.

We recall that, for a set of atomic elements At and for an endofunctor F given by $F(X) = At \times X$ and $F(f)(a, x) = (a, f(x))$, we have a terminal coalgebra $\iota : At^{\mathbb{N}} \to F(At^{\mathbb{N}})$ given by $\iota(\sigma) = (\mathrm{hd}(\sigma), \mathrm{tl}(\sigma))$, where hd and tl are respectively the head and tail of the stream σ. Then co-inductively defining a stream generator f consists of finding a F-coalgebra $\alpha : X \to F(X)$ such that the following diagram commutes.

$$\begin{array}{ccc} At^{\mathbb{N}} & \xrightarrow[\cong]{\iota} & At \times At^{\mathbb{N}} = F(At^{\mathbb{N}}) \\ {\scriptstyle f}\uparrow & & \uparrow{\scriptstyle F(f)} \\ X & \xrightarrow[\alpha]{} & At \times X = F(X) \end{array}$$

Basic type-checking tells us that α is of the form $x \mapsto (a, x')$ for $x \in X$. Commutativity implies that $(\mathrm{hd}(f(x)), \mathrm{tl}(f(x)) = \iota(f(x)) = F(f)(\alpha(x)) = F(f)(a, x') = (a, f(x'))$, giving us the following pair of equalities:

$$\begin{cases} \mathrm{hd}(f(x)) & = \pi_1(\alpha(x)) \\ \mathrm{tl}(f(x)) & = f(\pi_2(\alpha(x))) \end{cases} \tag{5}$$

We thus see that the stream generated by $f(x)$, $(a_0, a_1, a_2, \dots)$, is co-inductively defined by α, and we can give an inductive formula for the elements: with $x_0 = x \in X$, we let $x_{n+1} = \pi_2(\alpha(x_n))$ and $a_n = \pi_1(\alpha(x_n))$.

Throughout, we take $\wedge$ to be lazy cons.

Integers starting at n We are interested in generating the stream $(n, n+1, n+2, \dots)$ for a given integer n. Let us define this stream using the morphism $\text{nums\_from} : \mathbb{N} \to \mathbb{N}^{\mathbb{N}}$ given by $\text{nums\_from}(n) = n \wedge \text{nums\_from}(n+1)$. How do we co-inductively define this stream? That's to say, what is the morphism μ such that the following diagram commutes?

$$\begin{array}{ccc} \mathbb{N}^{\mathbb{N}} & \xrightarrow{\iota} & \mathbb{N} \times \mathbb{N}^{\mathbb{N}} = F(\mathbb{N}^{\mathbb{N}}) \\ {\scriptstyle \text{nums\_from}}\uparrow & & \uparrow{\scriptstyle F(\text{nums\_from})} \\ \mathbb{N} & \xrightarrow[\mu]{} & \mathbb{N} \times \mathbb{N} = F(\mathbb{N}) \end{array}$$

We need to have $\iota(\text{nums\_from}(n)) = F(\text{nums\_from})(\mu(n))$, where $\iota(\sigma)$ is as above. Letting $\mu(n) = (n, n+1)$, we have that $F(\text{nums\_from})(\mu(n)) = F(\text{nums\_from})(n, n+1) = (n, \text{nums\_from}(n+1)) = \iota(\text{nums\_from}(n))$, as we wanted.

More generally, for stream generators of the form $f : X \to X^{\mathbb{N}}$ with $f(x) = h(x) \wedge f(\nu(x))$, giving a co-inductive definition of the stream generator simply entails exhibiting a co-algebra $\mu : X \to X \times X$. This μ is readily seen to be of the form $\mu(x) = (h(x), \nu(x))$.

Generalized Fibonacci sequence We would like to generate the generalized Fibonacci sequence, that is to say, given integers a, b, we would like to generate the stream $(a, b, a+b, b+(a+b), (a+b)+(b+(a+b)), \dots)$. This stream is generated by $\text{fib}(a, b) = a \wedge \text{fib}(b, a+b)$. We see that $\mu(a, b) = (a, (b, a+b))$ is the co-algebra making the following diagram commute:

$$\begin{array}{ccc} \mathbb{N}^{\mathbb{N}} & \xrightarrow{\iota} & \mathbb{N} \times \mathbb{N}^{\mathbb{N}} = F(\mathbb{N}^{\mathbb{N}}) \\ {\scriptstyle \text{fib}}\uparrow & & \uparrow{\scriptstyle F(\text{fib})} \\ \mathbb{N} \times \mathbb{N} & \xrightarrow[\mu]{} & \mathbb{N} \times (\mathbb{N} \times \mathbb{N}) = F(\mathbb{N} \times \mathbb{N}) \end{array}$$

We can verify that this is correct by using equation (5). We have $\pi_1(\mu(a, b)) = \text{hd}(\text{fib}(a, b)) = a$, agreeing with our above definition of μ. We also have $\text{fib}(\pi_2(\mu(a, b))) = \text{tl}(f(a, b)) = \text{fib}(b, a+b)$, implying that $\pi_2(\mu(x)) = (b, a+b)$. Thus $\mu(a, b)$ is in fact $(a, (b, a+b))$.

Merging streams Given two streams in $X^{\mathbb{N}}$, we wouldd like to merge them, alternating between each. Thus, we have merge : $X^{\mathbb{N}} \times X^{\mathbb{N}} \to X^{\mathbb{N}}$ given by $(\sigma, \rho) \mapsto \mathrm{hd}(\sigma) \wedge \mathrm{merge}(\rho, \mathrm{tl}(\sigma))$. The associated μ making the diagram below commute is $\mu(\sigma, \rho) = (\mathrm{hd}(\sigma), \mathrm{merge}(\rho, \mathrm{tl}(\sigma)))$.

$$\begin{array}{ccc} X^{\mathbb{N}} & \xrightarrow{\quad\iota\quad} & X \times X^{\mathbb{N}} = F(X^{\mathbb{N}}) \\ {\scriptstyle \mathrm{merge}}\uparrow & & \uparrow{\scriptstyle F(\mathrm{merge})} \\ X^{\mathbb{N}} \times X^{\mathbb{N}} & \xrightarrow[\mu]{} & X \times (X^{\mathbb{N}} \times X^{\mathbb{N}}) = F(X^{\mathbb{N}} \times X^{\mathbb{N}}) \end{array}$$

Transducers Given an endomorphism $t : X \to X$, we can co-inductively define the "map" function common in functional programming. We have $\mathrm{map}(t, \sigma) = t(\mathrm{hd}(\sigma)) \wedge \mathrm{map}(t, \mathrm{tl}(\sigma))$ and need to find a β such that the following diagram commutes:

$$\begin{array}{ccc} X^{\mathbb{N}} & \xrightarrow{\qquad\iota\qquad} & X \times X^{\mathbb{N}} = F(X^{\mathbb{N}}) \\ {\scriptstyle \mathrm{map}}\uparrow & & \uparrow{\scriptstyle F(\mathrm{map})} \\ (X \to X) \times X^{\mathbb{N}} & \xrightarrow[\beta]{} & X \times ((X \to X) \times X^{\mathbb{N}}) = F((X \to X) \times X^{\mathbb{N}}) \end{array}$$

Using the same technique as the previous examples, we see that $\beta(t, \sigma) = (t(\mathrm{hd}(\sigma)), (t, \mathrm{tl}(\sigma)))$. Using map, we can now define all transducers on infinite streams.

Similarly, we can co-inductively define "odd" and "even", which respectively take the the odd-indexed and even-indexed elements of a stream. The corresponding βs are $\beta_o(\sigma) = (\mathrm{hd}(\sigma), \mathrm{tl}(\mathrm{tl}(\sigma)))$ and $\beta_e(\sigma) = (\mathrm{hd}(\mathrm{tl}(\sigma)), \mathrm{tl}(\mathrm{tl}(\sigma)))$ and the commutative diagram is:

$$\begin{array}{ccc} X^{\mathbb{N}} & \xrightarrow{\beta_o} & X \times X^{\mathbb{N}} = F(X^{\mathbb{N}}) \\ {\scriptstyle \mathrm{odd}}\downarrow & & \downarrow{\scriptstyle F(\mathrm{odd})} \\ X^{\mathbb{N}} & \xrightarrow{\iota} & X \times X^{\mathbb{N}} = F(X^{\mathbb{N}}) \\ {\scriptstyle \mathrm{even}}\uparrow & & \uparrow{\scriptstyle F(\mathrm{even})} \\ X^{\mathbb{N}} & \xrightarrow[\beta_e]{} & X \times X^{\mathbb{N}} = F(X^{\mathbb{N}}) \end{array}$$

13 Monads

Whenever there is an adjunction $F \dashv U$ with $F : \mathcal{C} \to \mathcal{D}$ and $U : \mathcal{D} \to \mathcal{C}$ the composite $UF : \mathcal{C} \to \mathcal{C}$ is a very special type of endofunctor on the category $\mathcal{C}$. It encodes the "free" construction of F but without leaving the category $\mathcal{C}$.

Definition 41. An endofunctor $T : \mathcal{C} \to \mathcal{C}$ is called a **monad** or **triple** if there are two natural transformations $\eta : I \to T$ and $\mu : T^2 \to T$ satisfying the following commuting diagrams:

$$\begin{array}{ccc} T^3X & \xrightarrow{\mu_{TX}} & T^2X \\ {\scriptstyle T\mu_X}\downarrow & & \downarrow{\scriptstyle \mu_X} \\ T^2X & \xrightarrow[\mu_X]{} & TX \end{array} \qquad \begin{array}{ccccc} TX & \xrightarrow{\eta_{TX}} & T^2X & \xleftarrow{T\eta_X} & TX \\ & \searrow & \downarrow{\scriptstyle \mu_X} & \swarrow & \\ & & TX. & & \end{array}$$

I said earlier that monads come from adjunctions. In an adjunction $F \dashv U$ as above, we have two natural transformations $\eta : id_{\mathcal{C}} \to UF$ and $\varepsilon : FU \to id_{\mathcal{D}}$. Clearly we have the natural transformation η as required in the definition of a monad. Where does the natural transformation μ come from? It cannot just be ε; that has the wrong type. However if we sandwich UF between an F and a U we get $FUFU$ which is T^2. Now it is clear that $\mu = U\varepsilon F$. Using naturality of ε one can show that the diagrams for μ and η given in the definition of a monad hold. We will record this as a proposition.

Proposition 42. If $(F, U, \eta, \varepsilon)$ is an adjunction then $(UF, \eta, U\varepsilon F)$ is a monad.

Does every monad come from an adjunction? The answer is "yes", but we will look at some monads in their own right.

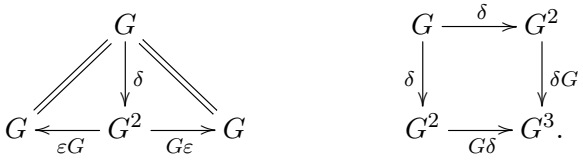
Here is an example of a monad that does not, on the face of it, look like it came from an adjunction. This is the covariant powerset monad. The functor is $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$ which takes a set X to its powerset $\mathcal{P}(X)$; it takes a function $f : X \to Y$ to the function $\mathcal{P}(f) : \mathcal{P}(X) \to \mathcal{P}(Y)$ given by $\mathcal{P}(f)(A \subseteq X) = \{f(a) \mid a \in A\}$. It is easy to check that this really is a functor. What are the natural transformations that show that it is a monad? We define $\eta_X(x) = \{x\}$; this is indeed a map of type $X \to \mathcal{P}(X)$. We define $\mu_X : \mathcal{P}(\mathcal{P}(X)) \to \mathcal{P}(X)$ by

$$\mu_X(\mathcal{A}) = \bigcup_{A \in \mathcal{A}} A.$$

Here $\mathcal{A}$ is a set of subsets of X, i.e. an element of $\mathcal{P}(\mathcal{P}(X))$; we just take their union. These are very intuitive definitions and the fact that the equations required of a monad hold is evident. As an exercise, the reader should verify that the free monoid construction gives a monad on **Set**.

The dual concept is called a *comonad*.

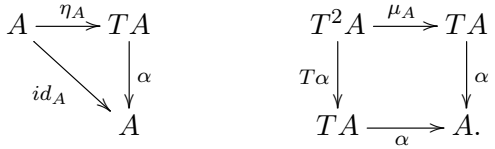
Definition 43. A **comonad** on a category $\mathcal{D}$ is a triple $(G, \varepsilon, \delta))$, where G is an endofunctor on $\mathcal{D}$ and ε and δ are natural transformations: $\varepsilon : G \longrightarrow id_{\mathcal{D}}$, $\delta : G \longrightarrow G^2$. They are required to obey the following equations:



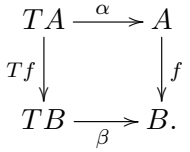
Of course a comonad on a category is the same thing as a monad on the opposite category.

Every monad arises from an adjunction in *at least two distinct ways*. Given a category $\mathcal{C}$ and a monad (T, η, μ) there are two ways of constructing a category $\mathcal{D}$ and an adjunction between $\mathcal{C}$ and $\mathcal{D}$ that reproduce T. These two constructions are called the Eilenberg-Moore construction and the Kleisli construction. I will show the Eilenberg-Moore construction first.

Definition 44. Given a cateogry $\mathcal{C}$ and a monad (T, η, μ) on it we define the category of **Eilenberg-Moore algebras** $\mathcal{C}^T$ as follows. The objects are morphisms from $\alpha : TA \longrightarrow A$ such that the following diagrams commute



A morphism from $\alpha : TA \longrightarrow A$ to $\beta : TB \longrightarrow B$ is a morphism f of $\mathcal{C}$ such that the following daigram commutes



Note that unlike the $T-algebras$ that we introduced earlier which were defined for any functor, these are defined only for monads and have additional requirements. Of course, T-algebras for monads are also T-algbras in the old sense. In this section, henceforth, I will always mean T-algebras for a monad.

We always have some T-algebras: given any object A we have the map $\mu_A : T^2A \rightarrow TA$ which is required to exist for any monad. These are called *free* T-algebras. Given any $\mathcal{C}$-morphism f from A to B we have that Tf is always a $\mathcal{C}^T$ morphism by naturality of μ. Given any T-algebra $\alpha : TA \rightarrow A$, α is also morphism from the free T-algebra to α since the definition of T-algebra requires that the following diagram commutes

$$\begin{array}{ccc} T^2A & \xrightarrow{\mu_A} & A \\ {\scriptstyle T\alpha}\downarrow & & \downarrow{\scriptstyle \alpha} \\ TA & \xrightarrow[\alpha]{} & A. \end{array}$$

We define an adjunction between $\mathcal{C}$ and $\mathcal{C}^T$ as follows. We define a functor $F^T : \mathcal{C} \rightarrow \mathcal{C}^T$ as follows: $F^T(A) = \mu_A : T^2A \rightarrow TA$ and $F^T(f : A \rightarrow B) = Tf$. We omit the easy verification that F^T is a functor. This will be the left part of the adjunction. The functor in the other direction is the forgetful functor $G^T : \mathcal{C}^T \rightarrow \mathcal{C}$ given by $G^T(\alpha : TA \rightarrow A) = A$ and it just maps morphisms in $\mathcal{C}^T$ to themselves but now viewed as morphisms of $\mathcal{C}$. It is immediate that $G^TF^T = T$. We have the natural transformation η given by the monad to serve as the natural transformation $\eta : id_{\mathcal{C}} \rightarrow G^TF^T$. We still need the natural transformation $\varepsilon : F^TG^T \rightarrow id_{\mathcal{C}^T}$. Given an object $\alpha : TA \rightarrow A$ of $\mathcal{C}^T$ we define the component of the natural transformation ε_α to be simply α! Why does this make sense? Recall that $G^T(\alpha) = A$ and $F^T(A) = \mu_A : T^A \rightarrow TA$. A natural transformation from F^TG^T to $id_{\mathcal{C}^T}$ must give, for every object α of $\mathcal{C}^T$ a morphism of $\mathcal{C}^T$ from $F^TG^T(\alpha)$ to $id_{\mathcal{C}^T}(\alpha)$, in other words from μ_A to α. But we have just noted above that α is precisely such a morphism. It is an easy calculation to see that $G^T\varepsilon F^T = \mu$ showing that this adjunction indeed reproduces our original monad.

It is instructive to work out what the T-algebras are for the monad that sends X to X^*.

The other route to obtaining an adjunction is the Kleisli construction. Before I give the construction let me motivate it a bit. Suppose that we have

a category $\mathcal{D}$ with an adjunction $F \dashv G$ with $F : \mathcal{C} \longrightarrow \mathcal{D}$ and suppose that our given monad (T, η, μ) arises as GF. Now consider the hom set $\mathcal{C}(A, TB) = \mathcal{C}(A, GFB) = \mathcal{D}(FA, FB)$. This suggests that we can construct a category $\mathcal{D}$ by using $\mathcal{C}$ objects but redefining the maps.

Definition 45. The **Kleisli category** $\mathcal{C}_T$ of a monad (T, η, μ) has the same objects as $\mathcal{C}$ and a morphism from A to B in $\mathcal{C}_T$ is a morphism from A to TB in $\mathcal{C}$. The identity morphism of an object A is η_A and composition is defined as follows: given morphisms $f : A \longrightarrow B$ and $g : B \longrightarrow C$ in $\mathcal{C}_T$ we define $g \circ f$ in $\mathcal{C}_T$ by

$$\mu_C \circ Tg \circ f.$$

It is straightforward to verify that this really is a category.

A very instructive exercise is to check that the Kleisli category of the powerset monad is the category **Rel**.

References

[AHS90] Jiri Adamek, Horst Herrlich, and George E. Strecker. *Abstract and Concrete Categories: The Joy of Cats*. Dover, 1990.

[Awo06] Steve Awodey. *Category Theory*. Oxford University Press, 2006.

[BW90] M. Barr and C. Wells. *Category Theory for Computing Science*. prentice-Hall, 1990.

[HS73] Horst Herrlich and George E. Straeker. *Category Theory*. Allyn and Bacon, Boston, 1973.

[Jac99] B. Jacobs. *Categorical Logic and Type Theory*. Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam, 1999.

[LS86] J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic*, volume 7 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1986.

[Mac98] S. Mac Lane. *Categories for the Working Mathematician*. Number 5 in Graduate Texts in Mathematics. Springer-Verlag, 1998. Second Edition.

[MM92] S. Mac Lane and Ieke Moerdijk. *Sheaves in geometry and logic.* Springer-Verlag, 1992.