Lecture 21: Adjunctions (Cont.)

Lecturer: Max S. New
Scribe: Owen Goebel

March 29, 2023

1 Recap (Adjoint Functors)

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "G"] \end{tikzcd}$$

Suppose we have functors $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{C}$. In the previous lecture, we covered four equivalent definitions of what it means for F to be *left adjoint* to G, abbreviated $F \dashv G$. Two of those definitions are given below:

1. (Definition 1.1) We have $\mathcal{D}(Fc, d) \cong \mathcal{C}(c, Gd)$. (Natural in $c : \mathcal{C}^{\mathrm{op}}$, $d : \mathcal{D}$.) Phrased differently, maps out of F in $\mathcal{D}$ are equivalent to maps into G in $\mathcal{C}$.

2. (Definition 1.2) There exist *"unit"*, $\eta : c \to GFc$, and *"co-unit"*, $\varepsilon : FGd \to d$, such that the following diagrams commute:

$$\begin{tikzcd} F \arrow[r, "F\eta"] \arrow[dr, "\mathrm{id}_F"'] & FGF \arrow[d, "\varepsilon F"] \\ & F \end{tikzcd} \qquad \begin{tikzcd} G \arrow[r, "\eta G"] \arrow[dr, "\mathrm{id}_G"'] & GFG \arrow[d, "G\varepsilon"] \\ & G \end{tikzcd}$$

We also write that F is right adjoint to G if G is left adjoint to F. As will be seen later, adjoint functors are a weakening of the notion of equivalence of categories.

2 Examples with Posets

Here are various examples of adjoint functor pairs between categories that are posets.[1]

[1] Such pairs of adjoint functors are typically called *Galois Connections*, named after the famous mathematician.

2.1 $\mathbb{R}$ and $\mathbb{Z}$

Consider the inclusion i from $(\mathbb{Z}, \leq)$ into $(\mathbb{R}, \leq)$, which is clearly a monotone function:

$$(\mathbb{Z}, \leq) \xhookrightarrow{\quad i \quad} (\mathbb{R}, \leq)$$

i is not an isomorphism, as i is not surjective. In general, it is impossible to define an isomorphism between these two posets as $\mathbb{R}$ and $\mathbb{Z}$ have different cardinalities, so we cannot define an equivalence of categories. However, we *can* find left *and* right adjoints for i, namely the floor and ceiling functions[2], respectively:

$$\begin{tikzcd}
(\mathbb{Z}, \leq) \arrow[r, hook, "i" description] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] & (\mathbb{R}, \leq) \arrow[l, bend right, "F = \lceil \cdot \rceil"'] \arrow[l, bend left, "G = \lfloor \cdot \rfloor"]
\end{tikzcd}$$

We see that $F \dashv i$ using either definition 1.1 or 1.2 (using $r \in \mathbb{R}$, $n \in \mathbb{Z}$):

1. By definition of F, $Fr \leq_{\mathbb{Z}} n$ if and only if $r \leq_{\mathbb{R}} n$. This gives $(\mathbb{Z}, \leq)(\lceil r \rceil, n) \cong (\mathbb{R}, \leq)(r, i(n))$ as desired.
2. We see that $r \leq \lceil r \rceil$ and $\lceil n \rceil \leq n$. These are our unit and co-unit, respectively. The diagrams trivially commute because we are working in posets.

We can similarly prove $i \dashv G$ using either definition:

1. By definition of G, $n \leq Gr$ if and only if $n \leq_R r$.
2. Unit is $\lfloor r \rfloor \leq r$ and co-unit is $n \leq \lfloor n \rfloor$.

Thus we get the chain of adjoints $G \dashv i \dashv F$. In general, chains of adjoints can be arbitrarily long, and they may be cyclic (trivial example: $\mathrm{Id} \dashv \mathrm{Id} \dashv \cdots$). However, F and G are *not* adjoint to each other, it doesn't even make sense as F and G do not point in opposite directions.

2.2 Bool and Bool^X

Let X be a set. Recall that a function from X to the booleans can be viewed as a predicate on X. We will construct adjoint functor pairs between the category of the poset $(\mathrm{Bool}, \Rightarrow)$ and the category of the poset $(\mathrm{Bool}^X, \leq)$ of predicates on X, where $\leq$ is given by pointwise ordering.[3] Define $\Delta : \mathrm{Bool} \to \mathrm{Bool}^X$ by $\Delta(b)(x) := b$, i.e. Δb is a constant predicate.[4] Note that Δ is trivially monotone.

$$\begin{tikzcd}
(\mathrm{Bool}, \Rightarrow) \arrow[r, "\Delta" description] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] \arrow[r, bend right, "G"'] & (\mathrm{Bool}^X, \leq) \arrow[l, bend right, "F"']
\end{tikzcd}$$

[2]The floor function $\lfloor x \rfloor$ returns the greatest integer that is at most x, and the ceiling function $\lceil x \rceil$ returns the smallest integer that is at least x.

[3]$P \leq Q$ iff $Px \Rightarrow Qx$ for all $x \in X$. (Aside: the predicates themselves are not "monotone", as we didn't assume X to have underlying structure.)

[4]The Δ stands for *diagonal*.

We get Δ has left and right adjoints F and G given by

$$FP := \exists x.Px$$
$$GP := \forall x.Px$$

It is straightforward to confirm $F \dashv \Delta$ via Definition 1.1 by noting that $FP \Rightarrow b$ if and only if $P \leq \Delta b$. Similarly, we note that $GP \Rightarrow b$ if and only if $\Delta b \leq P$, thus $\Delta \dashv G$. Thus we get $\exists \dashv \Delta \dashv \forall$.[5]

3 STT Connectives

In the previous lecture, we discussed two examples with the product and the terminal object. In fact, those examples can be seen as generalizations of the previous example, where the generalization goes as follows.

3.1 I-Products

Let $\mathcal{C}$ be a bicartesian category. Let I be a set. We can define an I-indexed category $\mathcal{C}^I$ in one of several ways, such as the functor category out of a discrete category, or as a category where the objects are an I-indexed family of objects in $\mathcal{C}$ where the morphisms are pointwise morphisms. We've seen various examples, such as $\mathcal{C}^2 = \mathcal{C} \times \mathcal{C}$ and $\mathcal{C}^0 = \mathbb{1}$.

Now we generalize the example from 2.2. Instead of a funct*ion* between the posets $(\mathrm{Bool}, \Rightarrow)$ and $(\mathrm{Bool}^X, \leq)$, we define a funct*or* $\Delta : \mathcal{C} \to \mathcal{C}^I$ given by

$$\Delta(c)(i) := c$$
$$\Delta(f)(i) := f$$

For example, in the case of $I = 2$, this means $c \mapsto (c, c)$, and in the case of $I = 0$, this means $c \mapsto ()$. We now generalize the adjoints from section 2.2 to get that Δ has left and right adjoints given by the coproduct (which we will denote as Σ) and product (which we will denote as Π), respectively:

$$\begin{tikzcd}
\mathcal{C} \arrow[r, "\Delta" description] \arrow[r, phantom, shift left=3, "\perp"] \arrow[r, phantom, shift right=3, "\perp"] & \mathcal{C}^I \arrow[l, bend right=40, "\Sigma"'] \arrow[l, bend left=40, "\Pi"]
\end{tikzcd}$$

If we have $\mathcal{C}$ is a bicartesian category, then it will have all such adjoints. We have seen multiple categories that have these chains of adjoints, such as Set, Gluing categories, syntactic categories, Lindenbaum Algebras, and all of them will have this type of adjoint. We will now note how to define the unit and co-unit for Π:

[5] This result was first discovered by Bill Lawvere, and is possibly the first discovery of the link between category theory and logic.

- We can define a unit $\eta^c : \mathcal{C}(c, \Pi_i(\Delta c))$ as a "diagonal" morphism from c to $\Pi_i(\Delta c)$.[6] For example, if $\mathcal{C}$ was the category of sets, η^c would be given by

$$\eta^c(x \in c) := (x, x, \dots)$$

- We can define a co-unit $\varepsilon^c : \mathcal{C}^I(\Delta(\Pi_i \vec{c}), \vec{c})$. For there to be morphism between these two objects, it means for each $i \in I$, we get a morphism from the product $\Pi_i \vec{c}$ to position i of $\vec{c}$. We can do this by letting component i of the morphism be the projection $\pi_i : \Pi_i \vec{c} \to \vec{c}_i$. We then define ε^c by as this pointwise collection of projections, which we can denote as $\vec{\pi}$.

This covers almost all connectives in STT, but it does not cover the exponential. We will address this in the following section.

3.2 Exponentials

Let $\mathcal{C}$ have binary products. Fix $s \in \mathcal{C}$. We will show that the functors $(- \times s)$ and $(-)^s$ are adjoints:

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "- \times s"] \arrow[r, phantom, "\perp"] & \mathcal{C} \arrow[l, bend left, "(-)^s"] \end{tikzcd}$$

When we view this through Definition 1.1, the above diagram becomes less exciting because it just means $\mathcal{C}(\Gamma \times s, a) \cong \mathcal{C}(\Gamma, a^s)$, which is true by definition. We could alternatively prove this by defining a unit $\eta : \mathcal{C}(a, s \Rightarrow (a \times s))$, which can be seen as performing some "stateful" computation, and a co-unit $\varepsilon : \mathcal{C}((s \Rightarrow a) \times s, a)$, which can be seen as function application.

With this, we conclude that if you squint your eyes, every connective in STT can be given by adjunctions. But this isn't too crazy, we already knew they were universal properties and thus they're all representable.

4 Useful facts about adjoints

We discuss some useful facts about adjunctions, including one which generalizes the previously noted fact that coproducts in a cartesian closed category are automatically distributive.

4.1 Uniqueness

In problem set 3, we proved that isomorphisms have unique inverses: if g and g' are inverses of f, then $g = g'$. We've also seen that if F and G form an equivalence of categories, and F and G' form an equivalence of categories, then G and G' are isomorphic. We can obtain a result for adjoints which is similar in spirit: namely,

[6] An important detail here is that this "diagonal" morphism is *within* the category $\mathcal{C}$, rather than Δ which is a functor *between* categories.

if $F \dashv G$ and $F \dashv G'$, then $G \cong G'$. To show this, we need to construct a natural isomorphism $Gd \to G'd$. (In fact, we can give a unique isomorphism that preserves unit and co-unit structure, refer to Riehl for that definition.) Since $F \dashv G'$, there exists a natural isomorphism $i : \mathcal{C}(FGd, d) \to \mathcal{D}(Gd, G'd)$, giving us that

$$\begin{array}{c} Gd \xrightarrow{i\varepsilon} G'd \\ \hline\hline FGd \xrightarrow[\varepsilon]{} d \end{array}$$

Symmetrically, if G has two left adjoints, F and F', then those adjoints are isomorphic.

4.2 Category of adjunctions

We can define a category of adjunctions where the objects are categories and the morphisms are the adjunctions between two categories.[7] In the diagram below, we define composition of $F \dashv G$ and $F' \dashv G'$ by $F' \circ F \dashv G \circ G'$, which we will soon show to be an adjoint pair.

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "\mathrm{Id}"] \arrow[r, phantom, "\perp"] & \mathcal{C} \arrow[l, bend left, "\mathrm{Id}"] \end{tikzcd} \qquad \begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "G"] \arrow[r, bend left, "F'"] \arrow[r, phantom, "\perp"] & \mathcal{E} \arrow[l, bend left, "G'"] \end{tikzcd}$$

Clearly each category $\mathcal{C}$ has an adjoint pair $\mathrm{Id} \dashv \mathrm{Id}$ with itself because $\mathcal{C}(\mathrm{Id}c, c') \cong \mathcal{C}(c, \mathrm{Id}c')$. We get that $\mathrm{Id} \dashv \mathrm{Id}$ is identity for $\mathcal{C}$ because $\mathrm{Id} \circ F = F$ and $G \circ \mathrm{Id} = G$. It suffices to show that any two adjoints can be composed to give another adjoint, or in the case of the above diagram, that $F' \circ F \dashv G \circ G'$. It satisfies definition 1.1:

$$\begin{aligned} \mathcal{E}(F'Fc, e) &\cong \mathcal{D}(Fc, G'e) && (F' \dashv G') \\ &\cong \mathcal{C}(c, GG'e) && (F \dashv G) \end{aligned}$$

As desired.[8]

4.3 Equivalences of categories are stronger than adjunctions

Throughout this lecture and the previous lecture, it's been hinted at that adjunction is a weaker form of equivalence[9], but it may not be immediately clear why this is true because it's not immediately obvious that every equivalence of categories is an adjunction. Consider an equivalence of categories between $\mathcal{C}$ and $\mathcal{D}$:

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "F^{-1}"] \end{tikzcd}$$

[7] More properly, a morphism in this category is a functor with the data that it is a left adjoint, which is more usefully viewed as a 2-category than as a 1-category.

[8] We've made everything so completely abstract that we can't possibly get the proof wrong.

[9] As a note, the converse isn't true. It's possible to have a chain of adjunctions $F \dashv G \dashv F$, where neither F nor G are full or faithful

By equivalence of categories, we have that

$$\begin{aligned} \mathrm{Id} &\cong FF^{-1} \\ FF^{-1} &\cong \mathrm{Id} \end{aligned}$$

which respectively suggest a unit η and co-unit ε. *However,* there is more data to an adjunction! Namely, we'd need our η and ε to satisfy the commuting triangles presented in Definition 1.2, and it's *not always true* that η and ε satisfying the above natural isomorphisms are sufficient to make those triangles commute!

But there is a silver lining. It's possible to modify at most one of η or ε to allow both triangles to commute. We are able to derive that if F and F^{-1} form an equivalence of categories, then $F \dashv F^{-1}$ and $F^{-1} \dashv F$, but you can't guarantee that both η *and* ε will be preserved, you can only preserve one of the two.

We can get a proof using Definition 1.1:

$$\begin{aligned} \mathcal{D}(Fc, d) &\cong \mathcal{C}(F^{-1}Fc, F^{-1}d) && (F^{-1} \text{ is fully faithful}) \\ &\cong \mathcal{C}(c, F^{-1}d) && (\text{precompose with } \varepsilon) \end{aligned}$$

Thus any theorem we prove about adjoint functors, we get the same result for free for equivalence of categories! Thus it's desirable to prove results for adjoints when possible, as those results hold more generally. Speaking of which...

4.4 RAP(L|P) and LAPC

RAPL ("right adjoints preserve limits") is perhaps the most famous basic theorem in category theory, besides maybe the Yoneda Lemma.[10] But we don't know what limits are, so we will focus on products, which are a special case of limits. Dually, there's also LAPC ("left adjoints preserve colimits", or in our case, "left adjoints preserve coproducts").

We get by corollary of LAPC that product by a fixed object, $- \times s$, preserves coproducts and initial object, which could have given us problem 1.2 of problem set 5 for free![11]

Suppose we have the following setup:

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "G"] \end{tikzcd}$$

We want to prove that products in $\mathcal{D}$ are preserved by G. That is, we want to show that if the left-hand side is a product diagram, then the right-hand side is also a

[10]If you go out to your topology or algebra class and notice there's a bunch of adjoint functors flying around, it's nice that you can instantly know that a bunch of properties are preserved by those functors from category theory. "Oh, this is a right adjoint? By corollary, here's a list of things it preserves!"

[11]e.g. we wanted that $0 \times s$ was initial, which follows from LAPC and $- \times s$ being left adjoint.

product diagram.

$$\begin{tikzcd}
& P \arrow[dl, "\pi_1"'] \arrow[d, "\pi_2"] \arrow[dr] & & \mapsto & & GP \arrow[dl, "G\pi_1"'] \arrow[d, "G\pi_2"] \arrow[dr] & \\
d_1 & d_2 & \cdots & \mapsto & Gd_1 & Gd_2 & \cdots
\end{tikzcd}$$

In more detail, if we have an object $c \in \mathcal{C}$ and morphisms $f_i : c \to Gd_i$, we would like there to be a unique morphism $(f_1, f_2, \cdots) : c \to GP$ such that the following diagram commutes:

$$\begin{tikzcd}
& c \arrow[d, dashed, "{(f_1,f_2,\dots)}"'] & \\
& GP \arrow[dl, "G\pi_1"'] \arrow[d, "G\pi_2"] \arrow[dr] & \\
Gd_1 & Gd_2 & \cdots \\
& c \arrow[ul, "f_1"] \arrow[u, "f_2"] \arrow[ur] &
\end{tikzcd}$$

But *oh no!* We don't know anything about GP! What do we do? We take advantage of the fact that $F \dashv G$, which gives us natural isomorphisms i and j:

$$\begin{tikzcd}
\mathcal{D}(Fc, d) \arrow[r, bend right, "j"'] \arrow[r, phantom, "\cong"] & \mathcal{C}(c, Gd) \arrow[l, bend right, "i"']
\end{tikzcd}$$

Therefore there exists unique $(if_1, if_2, \dots)$ such that the following diagram commutes:

$$\begin{tikzcd}
& Fc \arrow[d, dashed, "{(if_1,if_2,\dots)}"'] & \\
& P \arrow[dl, "\pi_1"'] \arrow[d, "\pi_2"] \arrow[dr] & \\
d_1 & d_2 & \cdots \\
& Fc \arrow[ul, "if_1"] \arrow[u, "if_2"] \arrow[ur] &
\end{tikzcd}$$

Applying j, we get the following diagram in $\mathcal{C}$:

$$\begin{tikzcd}
& c \arrow[d, dashed, "{j(if_1,if_2,\dots)}"'] & \\
& GP \arrow[dl, "G\pi_1"'] \arrow[d, "G\pi_2"] \arrow[dr] & \\
Gd_1 & Gd_2 & \cdots \\
& c \arrow[ul, "f_1"] \arrow[u, "f_2"] \arrow[ur] &
\end{tikzcd}$$

However, we have yet to show that the above is a *product* diagram! We'll get to that in a moment. But for now, we can note that the above steps can be compressed into a very slick looking proof:

$$\begin{aligned}
\mathcal{C}(c, GP) &\cong \mathcal{D}(Fc, P) && \text{(by } i\text{)} \\
&\cong \Pi_k \mathcal{D}(Fc, d_k) && \text{(by } \pi_k \circ -\text{)} \\
&\cong \Pi_k \mathcal{C}(c, Gd_k) && \text{(by } j\text{)}
\end{aligned}$$

All the above steps are correct. However, this does not *fully* prove the product is preserved! We still need to show that the projection maps in $\mathcal{C}$ are specifically the $G\pi_k$ that were hit by G.[12] The above 3-line proof has achieved the same results thus far as the diagrams, but we have yet to prove we have a product diagram. No worries! We complete the proof by noting that if $g \circ h \in \mathcal{D}(Fc, d)$, then $j(g \circ h) = Gg \circ j(h)$, thus for $f \in \mathcal{C}(c, GP)$ we get for any k that

$$\begin{aligned}
j(\pi_k \circ i(f)) &= (G\pi_k \circ j(i(f))) \\
&= (G\pi_k \circ f)
\end{aligned}$$

Which holds for $f = j(if_1, if_2, \dots)$ as desired. Now go forth and feel free to apply LAPC and RAPL in your future math endeavors! (If you're a math major.)

[12] It *does*, however, prove the weaker statements that right adjoints preserve terminal objects, as there are no product diagrams to preserve, and it proves that right adjoints of posets preserve meets because there are no "other" morphisms G could hit.