Lecture 18: Extensions to STT, Monoid Actions

Lecturer: Max S. New
Scribe: Jonathan Moore

March 27, 2023

1 Setup

1.1 Last Time

Last time we discussed a weakening of an isomorphism between two categories $\mathcal{C}, \mathcal{D}$, which we called equivalence. Instead of requiring the strict equalities $F^{-1} \circ F = Id_{\mathcal{C}}, F \circ F^{-1} = Id_{\mathcal{D}}$, we only ask that there are natural isomorphism between them ($\cong$). As we saw last time, there are many natural uses for this weakened notion of "equality"

1.2 Today

Today we are going to weaken this notion even further. This weakening is also very natural; in fact, we have actually already seen several examples of this in previous lectures. Rather than an equivalence of categories, we will have a pair of "Adjoint Functors".

2 Adjoint Functors

To gain some intuition into what adjoint functors are, we will start with a concrete example:

2.1 Concrete Example (Sets and Moniods)

Let us look at two categories we have used frequently in the past: the category of Sets, and the category of Monoids. As we saw previously, there is a functor, U, from Mon to Set that takes the underlying set of elements from the monoid.

We also talked about a functor from sets to monoids called the "free monoid", F. The free monoid on a set, X, is called FX, and is the monoid of finite sequences of elements of X. Additionally, we had $\eta : X \to U(FX)$ where we define $\eta(x) = [x]$.

$$\begin{tikzcd} \text{Set} \arrow[r, bend left, "F"] & \text{Mon} \arrow[l, bend left, "U"] \end{tikzcd}$$

Note that there are many possibilities for the definition of $\eta(x)$, but this one is special because it has the following universal property:

$$\forall i : X \to UM . \exists ! E[i] : FX \to M \text{ s.t. the following diagram commutes:}$$

$$\begin{tikzcd} U(FX) \arrow[rr, "U(E[i])"] & & UM \\ & X \arrow[ul, "\eta"] \arrow[ur, "i"'] & \end{tikzcd}$$

The homomorphism $E[i]$ is constructed as follows:

$$\begin{aligned} E[i]([x_0, x_1, ...]) &= E[i]([x_0], [x_1], ...) \\ &= E[i]([x_0]) \cdot E[i]([x_1]) \cdot ... \\ &= i(x_0) \cdot i(x_1) \cdot ... \end{aligned}$$

and

$$E[i]([\ ]) = e$$

Now that we have a construction from Mon to Set, we can show that this is, in fact, a functor. We're given a function $f : X \to Y$ (in Set), and we need to define a morphism of monoids:

$$F[f] : FX \to FY$$

Note that we don't need to define F explicitly, since we can construct it simply by specifying what to do with the underlying set. In this case, $F = E[\eta \circ F]$ (which can be proven to be a functor).

Now we have two functors, one from Mon to Set (U), and one from Set to Mon (F). But is this pair of functors an equivalence of categories? No! For example:

$$X \in Set_0$$
$$UFX = \text{finite sequences of Xs}$$

If this was an equivalence, then by definition $X \cong UFX$ for all sets. However, if we take the set 1, then we get that X is simply a one element set, but UFX is a monoid of the natural numbers; the two are clearly not isomorphic as one is finite and the other is infinite.

We will now weaken the requirement for an isomorphism, which is simply a morphism in one direction. However, we already have a morphism from $X \to UFX$, which is η.

$$\eta^X : X \to UFX$$

Lemma: η is a natural transformation (using the previous definition of F):

$$\eta : Id_{\text{Set}} \implies U \circ F$$

What about the other direction? Again it is not the case that we can construct a natural isomorphism between M and FUM for similar reasons: the one element monoid gets taken by FU to the monoid of natural numbers. But we can construct a natural homomorphism, this time from $F(U(M))$ to M. Note that $F(U(M))$ is a sequence of elements of a monoid $[m_0, m_1, ...]$. One way to think about it is that it is a syntactic expression which represents

$$m_0 \text{ “}\times\text{” } m_1 \text{ “}\times\text{” } ...$$

We can then "remove the quotation marks" and make it actual multiplication, so we map $[m_0, m_1, ...]$ to $m_0 \times m_1 \times ...$ (and we map the empty sequence to e). We will call this homomorphism ε.

Formally,

$$\varepsilon : F(U(M)) \to M$$
$$\varepsilon([m_0, m_1, ...]) = m_0 \cdot m_1 \cdot ...$$
$$\varepsilon([\,]) = e$$

and in fact this definition of ε is natural in M:

$$\varepsilon : F \circ U \implies Id_{\text{Mon}}$$

Now we have a similar, but weakened version of equivalence. Rather than saying that the two functors must be naturally isomorphic to Id, we instead have a "directed" version of that with natural transformations:

$$\eta : \text{Id}_{\text{Set}} \Rightarrow UF$$
$$\varepsilon : FU \Rightarrow \text{Id}_{\text{Mon}}$$

However, there is still something that we are missing from equivalence: given F, F^{-1} is determined up to unique natural isomorphism. This is not the currently case for our adjoint functors. We still want something similar to be true, though, so we will construct equations that relate η and ε together.

We want a morphism $F \implies F \circ U \circ F$, which we can construct by applying F to each morphism η. This is defined as follows:

$$F\eta : F \implies FUF$$

$$(F\eta)^X : FX \to FUFX$$
$$(F\eta)^X := F(\eta)^X$$

Note that this is also natural, since applying a functor to any natural transformation results in another natural transformation.

We also want a morphism in the other direction; $F \circ U \circ F \implies F$ using ε, which we can define as:

$$\varepsilon F : FUF \implies F$$
$$(\varepsilon F)^X : FUFX \to FX$$
$$\varepsilon(F)^X := \varepsilon^{FX}$$

This is also a natural transformation since we are just restricting the domain of ε, which is a natural transformation.

This results in the following diagrams (both of which commute):

$$\begin{tikzcd}
F \arrow[rr, Rightarrow, "F\eta"] \arrow[dr, Rightarrow, "Id_F"] & & F \circ U \circ F \arrow[dl, Rightarrow, "\varepsilon F"] \\
& F &
\end{tikzcd}$$

$$\begin{tikzcd}
U \arrow[rr, Rightarrow, "\eta U"] \arrow[dr, Rightarrow, "Id_U"] & & U \circ F \circ U \arrow[dl, Rightarrow, "U\varepsilon"] \\
& U &
\end{tikzcd}$$

3 Definition of Adjoint Functors

We can now define what a pair of adjoint functors is. Informally, it is simply two functors with a pair of natural transformations which satisfy these two triangle identities (the diagrams above). Below are 4 formal definitions, all of which are equivalent (they are in bijection; i.e. there are bijective functions between the sets defined in each).

The first definition abstracts the structure we've seen so far; however, it can be very tedious to construct and/or difficult to use. When actually working with adjoint functors, the second is potentially easier to use while remaining symmetric. Finally the 3rd and 4th are the easiest to construct a pair of adjoint functors with, but are not symmetric (they are each other's "mirrors").

3.1 Definition 1

a. $F : \mathcal{C} \to \mathcal{D}$

b. $G : \mathcal{D} \to \mathcal{C}$

c. $\eta : Id_{\mathcal{C}} \implies G \circ F$

d. $\varepsilon : F \circ G \implies Id_{\mathcal{D}}$

e. The following diagrams commute:

$$\begin{tikzcd} F \arrow[rr, Rightarrow, "F\eta"] \arrow[dr, Rightarrow, "Id_F"'] & & F \circ G \circ F \arrow[dl, Rightarrow, "\varepsilon F"] \\ & F & \end{tikzcd}$$

$$\begin{tikzcd} G \arrow[rr, Rightarrow, "\eta G"] \arrow[dr, Rightarrow, "Id_G"'] & & G \circ F \circ G \arrow[dl, Rightarrow, "G\varepsilon"] \\ & G & \end{tikzcd}$$

3.2 Definition 2 (Potentially Easier to Use)

a. $F : \mathcal{C} \to \mathcal{D}$

b. $G : \mathcal{D} \to \mathcal{C}$

c. $\mathcal{D}(Fc, d) \cong \mathcal{C}(c, Gd)$

where both sides of the natural isomorphism are from $\mathcal{C}^{\mathrm{OP}} \times \mathcal{D} \to \mathrm{Set}$.

Note that because of this definition, we call F the "left adjoint" and G the "right adjoint". The notation for this is $F \dashv G$.

3.3 Definition 3 (Easier to Construct, but Asymmetric)

An adjunction from $\mathcal{C}$ to $\mathcal{D}$ consists of:

a. $G : \mathcal{D} \to \mathcal{C}$ (a *functor*)

b. $F_0 : \mathcal{C}_0 \to \mathcal{D}_0$ (a *function*)

   where F_0 is not a functor, but rather a function from the objects of $\mathcal{C}$ to the objects of $\mathcal{D}$

c. $\forall c \in \mathcal{C}_0 . \eta^c \in \mathcal{C}(c, G(F_0 c))$

d. $\forall c \in \mathcal{C}_0 . \forall i : \mathcal{C}(c, Gd) . \exists! E[i] : \mathcal{D}(F_0 c, d)$ s.t. the following diagram commutes:

$$\begin{tikzcd} GF_0c \arrow[rr, "G(E[i])"] & & Gd \\ & c \arrow[ul, "\eta^c"] \arrow[ur, "i"'] & \end{tikzcd}$$

Note that this is just a generalization of our first concrete example with Set and Mon!

3.4 Definition 4 (Asymmetric the Other Way)

An adjunction from $\mathcal{C}$ to $\mathcal{D}$ consists of:

a. $F : \mathcal{C} \to \mathcal{D}$ (a *functor*)

b. $G_0 : \mathcal{D}_0 \to \mathcal{C}_0$ (a *function*)

c. $\forall d \in \mathcal{D}_0 . \varepsilon^d \in \mathcal{D}(F(G_0 d), d)$

d. $\forall d \in \mathcal{D}_0 . \forall j : \mathcal{D}(Fc, d) . \exists! I[j] : \mathcal{C}(c, G_0 d)$ s.t. the following diagram commutes:

$$\begin{tikzcd} Fc \arrow[rr, "F(I[j])"] \arrow[dr, "j"'] & & F(G_0d) \arrow[dl, "\varepsilon^d"] \\ & d & \end{tikzcd}$$

4 Closing Thoughts

4.1 Adjoint Functors and Universal Properties

It turns out that Definitions 3 and 4 are explicitly stating a universal property of objects. Using definition 4 as a starting point,

$$F : \mathcal{C} \to D,\ \text{fix } d \in \mathcal{D}_0$$

we can define a predicator on $\mathcal{C}$ called $\widetilde{G}(d)$ as follows:

$$\widetilde{G}(d)(c) := \mathcal{D}(Fc, d)$$

Additionally, given $g \in \widetilde{G}(d)(c)$ and $f : \mathcal{C}(c', c)$, we can define:

$$g * f \in \widetilde{G}(d)(c'), g * f := g \circ Ff$$

Note that Parts c and d of Definition 4 is exactly proving that this predicator is representable (constructing a universal element). Furthermore, almost all universal properties that we have seen in class are actually instances of adjoint functors!

4.2 Examples

4.2.1 Terminal Object

$\mathcal{C}$ has a terminal object iff the unique functor from $\mathcal{C}$ into the trivial category, $\mathbb{1}$:

$$\begin{tikzcd} * \arrow[loop left, "\mathrm{id}"] & \mathcal{C} \arrow[l, "!"'] \end{tikzcd}$$

has a right adjoint.

A right adjoint would be some G from $*$ to $\mathcal{C}$:

$$\begin{tikzcd} * \arrow[loop left, "\mathrm{id}"] \arrow[r, bend left, "G"] & \mathcal{C} \arrow[l, bend left, "!"] \end{tikzcd}$$

Now we will use Definition 2, which means that:

$$\mathbb{1}(!c, *) \cong \mathcal{C}(c, G*)$$

but since $\mathbb{1}$ is the trivial set, the left side is naturally isomorphic to $\{id_*\}$. So we get that there is exactly one morphism from c to $G*$, which is exactly our definition of a terminal object. Thus we could define a termial object as a kind of adjoint functor.

4.2.2 Binary Products

$\mathcal{C}$ has all binary products iff:

$$\begin{tikzcd} \mathcal{C} \times \mathcal{C} & \mathcal{C} \arrow[l, "\Delta"'] \end{tikzcd}$$

has a right adjoint, where Δ is defined as: $\Delta c = (c, c); \Delta f = (f, f)$.

This right adjoint would be the "times" functor, which would construct a product from two objects:

$$\begin{tikzcd} \mathcal{C} \times \mathcal{C} \arrow[r, bend left, "\times"] & \mathcal{C} \arrow[l, bend left, "\Delta"] \end{tikzcd}$$

The adjunction would say that:

$$\mathcal{C}(c, a \times b) \cong (\mathcal{C} \times \mathcal{C}(\Delta c, (a, b)) \cong \mathcal{C}(c, a) \times \mathcal{C}(c, b)$$

which is exactly what we wanted, except that it should also be natural in c. However, this follows from the bijection between Definition 4 and Definition 3!