Topics in Koszul Duality, Michaelmas 2019, Oxford University

Lecture 3: The Barr-Beck Theorem

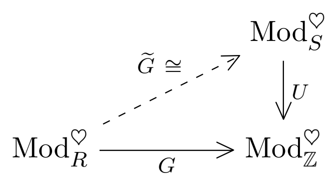
Let R be a ring. Last week, we have reformulated three algebraic properties of left R-modules $Q \in \mathrm{Mod}_R^{\heartsuit}$ in terms of categorical conditions on the associated functor

$$G = \mathrm{Map}_R(Q, -) : \mathrm{Mod}_R^{\heartsuit} \longrightarrow \mathrm{Mod}_{\mathbb{Z}}^{\heartsuit} = \mathrm{Ab}$$

to abelian groups. More specifically, we have seen:

Q is finitely presented	$\leftrightsquigarrow$	G preserves filtered colimits, i.e. Q is compact;
Q is projective	$\leftrightsquigarrow$	G preserves reflexive coequalisers;
Q is a generator	$\rightsquigarrow$	G is conservative.

If these conditions are satisfied, we wish to prove Proposition 2.1 from last lecture, asserting that the natural lift $\widetilde{G}$ of the functor G to $S = \mathrm{End}_Q(R)^{op}$-modules is an equivalence:



To give a categorical construction of $\mathrm{Mod}_S^{\heartsuit}$ and $\widetilde{G}$, we will need to recall some basic notions.

3.1. **Monads.** Monads provide a way of axiomatising algebraic structures that is convenient for certain abstract arguments. We illustrate this with a simple example:

Example 3.1 (Groups)**.** Traditionally, groups are defined as sets X with a binary multiplication $(x, y) \mapsto x \cdot y$, a unary inverse $x \mapsto x^{-1}$, and a unit e satisfying various axioms.

We could also choose a less economical approach, and specify many more operations, e.g.

$$(1) \qquad (x_1, x_2, x_3) \mapsto x_1 \cdot x_3^{10} \cdot x_2^{-1}, \qquad (x_1, x_2, x_3, x_4) \mapsto x_1^4 \cdot x_2^2 \cdot x_3 \cdot x_4^{-15}, \qquad \text{etc.}$$

More precisely, consider the endofunctor $T_{\mathrm{Gp}} : \mathrm{Set} \to \mathrm{Set}$ sending a set X to the set of all formal expressions

$$T_{\mathrm{Gp}}(X) \;:=\; \{\; x_1^{a_1} x_2^{a_2} \dots x_k^{a_k} \;\;|\;\; k \geq 0,\; x_i \in X,\; a_k \in \mathbb{Z} - \{0\},\; x_i \neq x_{i+1} \text{ for all } i. \;\}$$

Here the empty word ( ) is considered a valid element of the set $T_{\mathrm{Gp}}(X)$.

In our uneconomical approach to groups, defining all operations as in (1) amounts to specifying a single map $\alpha : T_{\mathrm{Gp}}(X) \to X$ sending a formal expression $x_1^{a_1} x_2^{a_2} \dots x_k^{a_k}$ to the value of the corresponding product $x_1^{a_1} \cdot x_2^{a_2} \cdot \dots \cdot x_k^{a_k}$ in X.

However, not all such maps $\alpha : T_{\mathrm{Gp}}(X) \to X$ define valid group structures on the set X, as we have not yet imposed any of the group axioms. To fix this, we exhibit additional structure on the endofunctor T_{Gp} by specifying the following natural maps for all sets X:

$$\eta_X : X \to T_{\mathrm{Gp}}(X) \qquad\qquad \mu_X : T_{\mathrm{Gp}}(T_{\mathrm{Gp}}(X)) \to T_{\mathrm{Gp}}(X).$$

Exercise. Before turning the page, have a guess what these maps are.

The first map η_X takes an element $s \in X$ to the corresponding one-letter word in $T_{\mathrm{Gp}}(X)$. The second map μ_X sends a "word of words" $(x_{11}^{a_{11}} \dots x_{1k_1}^{a_{1k_1}})^{b_1} \dots\dots (x_{n1}^{a_{n1}} \dots x_{nk_n}^{a_{nk_n}})^{b_n}$ in $T_{\mathrm{Gp}}(T_{\mathrm{Gp}}(X))$ to the corresponding word in $T_{\mathrm{Gp}}(X)$ given by

$$\underbrace{(x_{11}^{a_{11}} \dots x_{1k_1}^{a_{1k_1}}) \dots (x_{11}^{a_{11}} \dots x_{1k_1}^{a_{1k_1}})}_{b_1} \dots\dots \underbrace{(x_{n1}^{a_{n1}} \dots x_{nk_n}^{a_{nk_n}}) \dots (x_{n1}^{a_{n1}} \dots x_{nk_n}^{a_{nk_n}})}_{b_n}$$

Here, we have implicitly simplified this word by reducing subwords of the form $x^a x^b$ to x^{a+b}.

Exercise. The maps η_X and μ_X are natural in X and satisfy the following identities:

$$\mu_X \circ T_{\mathrm{Gp}}(\mu_X) \cong \mu_X \circ \mu_{T_{\mathrm{Gp}}(X)}, \qquad \mu_X \circ \eta_{T_{\mathrm{Gp}}(X)} = \mathrm{id}_{T_{\mathrm{Gp}}(X)} = \mu_X \circ T_{\mathrm{Gp}}(\eta_X).$$

Using the natural transformations η and μ, we can now formulate a condition for when a map $\alpha : T_{\mathrm{Gp}}(X) \to X$ defines a group structure on X:

Exercise. Given a map $\alpha : T_{\mathrm{Gp}}(X) \to X$, the operations $(x, y) \mapsto \alpha(xy)$, $x \mapsto \alpha(x^{-1})$, $e = \alpha(\,)$ define a group structure on X if and only if $\alpha \circ \eta_X = \mathrm{id}_X$ and $\alpha \circ \mu_X = \alpha \circ T_{\mathrm{Gp}}(\alpha)$.

We therefore obtain a second definition of what a group is, namely a set X together with a map of sets $T_{\mathrm{Gp}}(X) \to X$ satisfying $\alpha \circ \eta_X = \mathrm{id}_X$ and $\alpha \circ \mu_X = \alpha \circ T_{\mathrm{Gp}}(\alpha)$.

Definitions of this kind can also be given for most other algebraic structures of interest (like modules, rings, Lie algebras, ...). We therefore axiomatise this situation:

Definition 3.2 (Monads)**.** A *monad* on a category $\mathcal{C}$ is an associative algebra object in the monoidal category $\mathrm{End}(\mathcal{C})$ of endofunctors (with the composition product $\circ$).

Concretely, this means that a monad is an endofunctor $T : \mathcal{C} \to \mathcal{C}$ equipped with natural transformations $\mathrm{id}_{\mathcal{C}} \to T$ and $\mu : T \circ T \to T$ such that the following two diagrams commute:

$$\begin{array}{ccc} T\circ T\circ T & \xrightarrow{T(\mu)} & T\circ T \\ {\scriptstyle \mu_T}\downarrow & & \downarrow{\scriptstyle \mu} \\ T\circ T & \xrightarrow[\mu]{} & T \end{array} \qquad\qquad \begin{array}{ccc} T & \xrightarrow{\eta_T} & TT \\ {\scriptstyle T(\eta)}\downarrow & \searrow^{\mathrm{id}} & \downarrow{\scriptstyle \mu} \\ T^2 & \xrightarrow[\mu]{} & T \end{array}$$

Definition 3.3 (Algebras over monads)**.** An *algebra* over a monad T on $\mathcal{C}$ is a T-module object in the $\mathrm{End}(\mathcal{C})$-tensored category $\mathcal{C}$. Concretely, this means that an algebra is a pair $(A \in \mathcal{C}, \alpha : T(A) \to A)$ for which the following two diagrams commute:

$$\begin{array}{ccc} A & \xrightarrow{\eta_A} & T(A) \\ & {\scriptstyle \mathrm{id}_A}\searrow & \downarrow{\scriptstyle \alpha} \\ & & A \end{array} \qquad\qquad \begin{array}{ccc} T(T(A)) & \xrightarrow{T(\alpha)} & T(A) \\ {\scriptstyle \mu_A}\downarrow & & \downarrow{\scriptstyle \alpha} \\ T(A) & \xrightarrow[\alpha]{} & A \end{array}$$

We write $\mathrm{Alg}_T(\mathcal{C})$ for the category of T-algebras in $\mathcal{C}$.

In Example 3.1, we constructed a monad T_{Gp} acting on $\mathcal{C} = \mathrm{Set}$ whose category of algebras $\mathrm{Alg}_{T_{\mathrm{Gp}}}(\mathrm{Set})$ is equivalent to the category of groups.

In the next exercise, we will construct similar monads for other algebraic structures:

Exercise 3.4.

a) Define a monad T_{Ab} on the category of sets Set such that $\mathrm{Alg}_{T_{\mathrm{Ab}}}(\mathrm{Set})$ is equivalent to the category $\mathrm{Ab} = \mathrm{Mod}_{\mathbb{Z}}^{\heartsuit}$ of abelian groups.

b) Define a monad T_{Ring} on the category Ab such that $\mathrm{Alg}_{T_{\mathrm{Ring}}}(\mathrm{Ab})$ is the category of rings.

c) Given a ring R, define a monad T_{Ring} on Ab whose category of algebras is equivalent to the category of (left) R-modules.

3.2. **Monadic Adjunctions.** In Example 3.1, we have adopted the perspective that the monad T_{Gp} can be used as a tool for defining the notion of a group.

We could also reverse this logic and try to define the monad T_{Gp} assuming that we already know what a group is. To this end, recall the following standard notion from category theory (which we will later generalise to higher categories):

Definition 3.5 (Adjunctions)**.** An adjunction consists of functors $F : \mathcal{C} \leftrightarrows \mathcal{D} : G$ together with natural transformations $\eta : \mathrm{id}_{\mathcal{C}} \to GF$ (the "unit"), $\epsilon : FG \to \mathrm{id}_{\mathcal{D}}$ (the "counit") for which the following diagrams commute:

The functor F is called the left adjoint, whereas G is called a right adjoint; we write $F \dashv G$.

Remark 3.6. Fix an adjunction (F, G, η, ϵ) as in Definition 3.5. For any pair of objects $X \in \mathcal{C}$ and $Y \in \mathcal{D}$, we obtain natural isomorphisms

$$\mathrm{Map}_{\mathcal{D}}(FX, Y) \cong \mathrm{Map}_{\mathcal{C}}(X, GY)$$

Indeed, given $f : FX \to Y$ in $\mathcal{D}$, we attach the map $\overline{f} : X \to GY$ defined by $\overline{f} = Gf \circ \eta_X$. Conversely, to a map $g : X \to GY$, we attach the map $\overline{g} = \epsilon_Y \circ Fg : FX \to Y$.

In fact, specifying natural isomorphisms $\mathrm{Map}_{\mathcal{D}}(FX, Y) \cong \mathrm{Map}_{\mathcal{C}}(X, GY)$ leads to an equivalent definition of adjunctions

Example 3.1 (continued)**.** There is a free-forgetful adjunction $\mathrm{Free} : \mathrm{Set} \leftrightarrows \mathrm{Gp} : \mathrm{Forget}$ between the category of sets and the category of groups. The right adjoint Forget sends a group to its underlying set, and the left adjoint Free builds the free group on a given set. The unit $\eta_X : X \to \mathrm{Forget}(\mathrm{Free}(X))$ embeds a set X into the free group generated by X. The counit $\epsilon_G : \mathrm{Free}(\mathrm{Forget}(G)) \to G$ takes a formal product $g_1^{a_1} \dots g_n^{a_n}$ in the free group on the set G and computes the corresponding product $g_1^{a_1} \cdot \ldots \cdot g_n^{a_n}$ in the group G.

We note that the endofunctor $T_{\mathrm{Gp}} : \mathrm{Set} \to \mathrm{Set}$ defined above is equal to the composite $\mathrm{Forget} \circ \mathrm{Free}$. The transformation $\mathrm{id}_{\mathrm{Set}} \to T_{\mathrm{Gp}}$ agrees with the unit η of the adjunction, and the monad multiplication $\mu : T_{\mathrm{Gp}} \circ T_{\mathrm{Gp}} \to T_{\mathrm{Gp}}$ is given by $G\epsilon_F : GFGF \to GF$.

The functor $\mathrm{Gp} \to \mathrm{Alg}_{T_{\mathrm{Gp}}}(\mathrm{Set})$ sending a group G to the T_{Gp}-algebra

$$\left(\mathrm{Forget}(G)\ ,\ T_{\mathrm{Gp}}(\mathrm{Forget}(G)) \xrightarrow{\mathrm{Forget}(\epsilon_G)} \mathrm{Forget}(G)\right)$$

gives the equivalence between groups and T_{Gp}-algebras mentioned above.

Indeed, we obtain a monad for every adjunction:

Exercise 3.7 (Monads from adjunctions)**.** Given an adjunction $F : \mathcal{C} \leftrightarrows \mathcal{D} : G$ with unit $\eta : \mathrm{id}_{\mathcal{C}} \to GF$ and counit $\epsilon : FG \to \mathrm{id}_{\mathcal{D}}$, show that the endofunctor $T = GF$ is equipped with the structure of a monad with unit $\eta : \mathrm{id}_{\mathcal{C}} \to GF$ and multiplication $G\epsilon_F : T \circ T \to T$.

Exercise 3.8. Show that all monads in Exercise 3.4 are induced by corresponding forgetful-free adjunctions.

Exercise 3.9. Given a monad T on a category $\mathcal{C}$, consider the functor $\mathrm{Free}_T : \mathcal{C} \to \mathrm{Alg}_T(\mathcal{C})$ sending an object $X \in \mathcal{C}$ to the T-algebra $(TX,\ T(T(X)) \xrightarrow{\mu_X} T(X))$.

a) Prove that Free_T is a left adjoint to the forgetful functor $\mathrm{Forget}_T : \mathrm{Alg}_T(\mathcal{C}) \to \mathcal{C}$.
b) Verify that the adjunction $\mathrm{Free}_T \dashv \mathrm{Forget}_T$ induces the monad T.

This implies the interesting fact that *any* monad is induced by an adjunction.

Notation 3.10. We will usually denote the free T-algebra on an object $X \in \mathcal{C}$ by $T(X)$ instead of $\mathrm{Free}_T(X)$. Moreover, we will often drop the functor Forget_T from our notation.

If $T = GF$ is a monad obtained from an adjunction $F \dashv G$, we always obtain a functor

$$\widetilde{G} : \mathcal{D} \to \mathrm{Alg}_T(\mathcal{C})$$

sending an object $X \in \mathcal{D}$ to the T-algebra $(G(X)\ ,\ T(G(X)) \xrightarrow{G(\epsilon_X)} G(X))$.

Definition 3.11. The adjunction $F \dashv G$ is *monadic* if $\widetilde{G} : \mathcal{D} \to \mathrm{Alg}_T(\mathcal{C})$ is an equivalence.

In the case of groups, we have seen in Example 3.1 that the forgetful-free adjunction is monadic (thereby giving an alternative definition of groups as T_{Gp}-algebras).

However, not all adjunctions share this desirable property:

Exercise 3.12 (A non-monadic adjunction)**.** There is an adjunction $F : \mathrm{Set} \rightleftarrows \mathrm{Top} : G$ between sets and topological spaces: the right adjoint G sends a space to its underlying set of points; the left adjoint F equips a set with the discrete topology.

Show that this adjunction is *not* monadic. *Hint:* what does G do to isomorphisms?

3.3. **Morita theory as an adjunction.** We return to our toy example of Proposition 2.1 from last lecture, where we fixed a ring R and a compact projective generator $Q \in \mathrm{Mod}_R^{\heartsuit}$.

We can now give a purely categorical construction of the category Mod_S and the functor $\widetilde{G} : \mathrm{Mod}_R^{\heartsuit} \to \mathrm{Mod}_S^{\heartsuit}$ for $S = \mathrm{End}_R(Q)^{op}$, as desired.

To this end, observe that the tensor-hom-adjunction

$$Q \otimes (-) : \mathrm{Mod}_{\mathbb{Z}}^{\heartsuit} \leftrightarrows \mathrm{Mod}_R^{\heartsuit} : \mathrm{Map}_R(Q, -)$$

induces a monad T on $\mathrm{Ab} \cong \mathrm{Mod}_{\mathbb{Z}}^{\heartsuit}$ sending M to $\mathrm{Map}_R(Q, Q \otimes M)$. Hence $T(\mathbb{Z}) = \mathrm{End}_R(Q)$.

In fact, we can use the conditions on Q to identify the endofunctor T more explicitly. Since the right adjoint $G = \mathrm{Map}_R(Q, -)$ preserves biproducts, filtered colimits, and reflexive coequalisers, it must preserve small colimits. As this is also true for the left adjoint $Q \otimes (-)$, we deduce that the monad $T : \mathrm{Ab} \to \mathrm{Ab}$ preserves small colimits.

Exercise 3.13. Use Exercise 1.3 from Lecture 1 to identify $\mathrm{Alg}_T(\mathrm{Ab})$ with the category of left modules Mod_S over the ring $S = \mathrm{End}_R(Q)^{op}$.

3.4. **The Barr-Beck theorem.** To prove Proposition 2.1 from last class, we are therefore reduced to showing that the tensor-hom-adjunction $Q \otimes (-) : \mathrm{Mod}_{\mathbb{Z}}^{\heartsuit} \leftrightarrows \mathrm{Mod}_R^{\heartsuit} : \mathrm{Map}_R(Q, -)$ is monadic. This follows from the following much more general and important result:

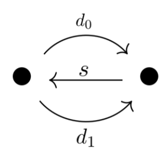
Theorem 3.14 (Barr-Beck theorem, crude version)**.**
Assume that an adjunction $F : \mathcal{C} \leftrightarrows \mathcal{D} : G$ satisfies the following two properties:

a) $\mathcal{D}$ admits and G preserves reflexive coequalisers;
b) G is conservative (i.e. reflects isomorphisms).

Then $(F \dashv G)$ is monadic (cf. Definition 3.11), i.e. $\widetilde{G} : \mathcal{D} \xrightarrow{\cong} \mathrm{Alg}_T(\mathcal{C})$ is an equivalence.

For the proof, we will need two slightly different notions of coequalisers. First, we recall:

Definition 3.15 (Reflexive coequaliser)**.** A *reflexive pair* in a category $\mathcal{C}$ is a diagram consisting of two arrows $d_0, d_1 : X_1 \rightrightarrows X_0$ and a common section $s : X_0 \to X_1$:



A *reflexive coequaliser* is the colimit of a reflexive pair; it agrees with the coequaliser of d_0, d_1.

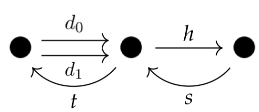
We need a second notion of coequaliser, which looks similar, but is in fact quite different:

Definition 3.16 (Split coequaliser)**.** Two parallel arrows $d_0, d_1 : X_1 \rightrightarrows X_0$ in a category $\mathcal{C}$ are called a *split pair* if there exist arrows

$$h : X_0 \to X_{-1}, \qquad s : X_{-1} \to X_0, \qquad t : X_0 \to X_1$$

satisfying the following identities:

$$hd_0 = hd_1 \qquad hs = \mathrm{id}_{X_{-1}} \qquad d_0 t = \mathrm{id}_{X_0} \qquad d_1 t = sh$$



Exercise 3.17. Show that in the situation of Definition 3.16, $X_1 \rightrightarrows X_0 \to X_{-1}$ is a coequaliser. Deduce that it is preserved by any functor – we call this an *absolute colimit.*

Using split coequalisers, we can build canonical free resolutions of algebras over monads:

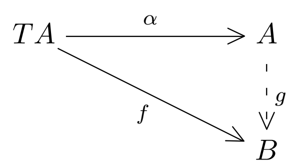
Proposition 3.18 (Free resolutions)**.** Fix a monad T on a category $\mathcal{C}$ and a T-algebra $(A, \alpha : T(A) \to A)$. The following diagram of T-algebras is a coequaliser in $\mathrm{Alg}_T(\mathcal{C})$:

$$T(T(A)) \underset{\mu_A}{\overset{T(\alpha)}{\rightrightarrows}} T(A) \xrightarrow{\alpha} A \tag{2}$$

Here, we have used the free functor $\mathcal{C} \to \mathrm{Alg}_T(\mathcal{C})$ from Exercise 3.9 (using Notation 3.10), which sends an object $X \in \mathcal{C}$ to the free T-algebra $(T(X),\ T(T(X)) \xrightarrow{\mu_X} T(X))$ on X.

Proof. Observe that after applying the forgetful functor $\mathrm{Alg}_T(\mathcal{C}) \to \mathcal{C}$, the above diagram is part of a split coequaliser with maps $s = \eta_A : A \to T(A)$ and $t = \eta_{T(A)} : T(A) \to T(T(A))$.

To verify that (2) is also a coequaliser in $\mathrm{Alg}_T(\mathcal{C})$, assume we are given a T-algebra $(B, \beta : T(B) \to B)$ together with a map of T-algebras $f : TA \to B$ with $f \circ T(\alpha) = f \circ \mu_A$. By Exercise 3.17, there is a unique $g = f \circ \eta_A$ in $\mathcal{C}$ such that the following triangle commutes:



Hence, it suffices to check that g is a map of T-algebras, which follows from the computation

$$\beta \circ Tf \circ T(\eta_A) = f \circ \mu_A \circ T(\eta_A) = f = f \circ \mu_A \circ \eta_{TA} = f \circ T(\alpha) \circ \eta_{TA} = (f \circ \eta_A) \circ \alpha.$$

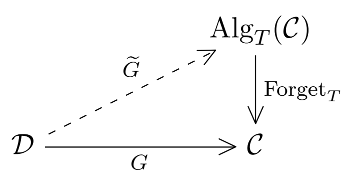
Here, we have used that f is a map of T-algebras, the monad axioms for T, and the naturality of η. $\square$

With these free resolutions at our disposal, we can now prove the Barr-Beck theorem.

Proof of Theorem 3.14. We proceed in three main steps.

Step 1: Left adjoint $\widetilde{F}$ to $\widetilde{G}$.

We have a commuting triangle



where both G and Forget_T admit left adjoints (cf. Exercise 3.9).

As left adjoints of commuting right adjoints commute, we know that *if* $\widetilde{G}$ admits a left adjoint $\widetilde{F}$, then its value on free T-algebras must be given by $\widetilde{F}(T(X)) = F(X)$.

Since left adjoints also preserve small colimits, Proposition 3.18 motivates us to define the value of $\widetilde{F}$ on a general T-algebra (A, α) as the following coequaliser in $\mathcal{D}$:

$$F(T(A)) \underset{\epsilon_{FA}}{\overset{F(\alpha)}{\rightrightarrows}} F(A) \xrightarrow{\theta} \widetilde{F}(A) \tag{3}$$

This makes sense as $F(T(A)) \underset{\epsilon_{FA}}{\overset{F(\alpha)}{\rightrightarrows}} F(A)$ is a reflexive pair in $\mathcal{D}$ with common section $F\eta_A$. One easily extends this definition to morphisms of T-algebras.

To verify that $\widetilde{F}$ is indeed left adjoint to $\widetilde{G}$, we make the following computation:

$$\begin{array}{c}
\widetilde{F}(A,\alpha) \to B \\ \hline
FA \xrightarrow{f} B \text{ s.t. } f \circ F(\alpha) = f \circ \epsilon_{FA} \\ \hline
A \xrightarrow{\overline{f}} B \text{ s.t. } \overline{f} \circ \alpha = G(\epsilon_B)G(F\overline{f}) \\ \hline
(A,\alpha) \to \widetilde{G}(B) = (GB, G\epsilon_B).
\end{array}$$

In the second step, we have used that $\overline{f} \circ \alpha = \overline{f \circ F(\alpha)} = \overline{f \circ \epsilon_{FA}} \overset{3)}{=} G(f) \overset{4)}{=} G(\epsilon_B)G(F(\overline{f}))$. Here $\overline{(\ )}$ denotes the adjoint bijection on morphisms introduced in Remark 3.6. The first two equalities are straightforward; equalities 3) and 4) follow from the commutative diagrams

$$3)\quad \begin{array}{ccc}
GFA & \longrightarrow & GB \\
{\scriptstyle \eta_{GFA}}\downarrow & \searrow^{\mathrm{id}} & \uparrow{\scriptstyle Gf} \\
GFGFA & \xrightarrow[G\epsilon_{FA}]{} & GFA
\end{array}
\qquad\qquad
4)\quad \begin{array}{ccc}
FA & \xrightarrow{F\overline{f}} & FGB \\
 & \searrow_{f} & \downarrow{\scriptstyle \epsilon_B} \\
 & & B
\end{array}\ .$$

Step 2: The unit $\mathrm{id}_{\mathrm{Alg}_T(\mathcal{C})} \to \widetilde{F} \circ \widetilde{G}$ *is an equivalence.*

Given $(A,\alpha) \in \mathrm{Alg}_T(\mathcal{C})$, we have a reflexive coequaliser $F(T(A)) \underset{\epsilon_{FA}}{\overset{F(\alpha)}{\rightrightarrows}} F(A) \xrightarrow{\theta} \widetilde{F}(A)$.

Using that G preserves reflexive coequalisers, we obtain another coequaliser diagram

$$GF(GF(A)) \underset{G\epsilon_{FA}}{\overset{GF(\alpha)}{\rightrightarrows}} GF(A) \xrightarrow{G\theta} G\widetilde{F}(A)$$

As in the proof of *Proposition* 3.18, the following diagram admits a splitting:

$$GF(GF(A)) \underset{G\epsilon_{FA}}{\overset{GF(\alpha)}{\rightrightarrows}} GF(A) \xrightarrow{\alpha} A$$

Hence, we have computed the coequaliser of $GF(GF(A)) \underset{G\epsilon_{FA}}{\overset{GF(\alpha)}{\rightrightarrows}} GF(A)$ in two ways, and obtain an isomorphism

$$\begin{array}{ccc}
GFA & \xrightarrow{G\theta} & G\widetilde{F}(A,\alpha) \\
 & \searrow_{\alpha} & \downarrow{\scriptstyle \cong} \\
 & & A
\end{array}$$

We can therefore identify A with $G\widetilde{F}(A,\alpha)$.

Next, we check that $G\epsilon_{\widetilde{F}(A,\alpha)} = \alpha$. Since $\alpha = G\theta$, it suffices to check that $\epsilon_{\widetilde{F}(A,\alpha)} = \theta$. This follows from the following computation:

$$\theta = \theta \circ F\alpha \circ F\eta_A = \theta \circ \epsilon_{FA} \circ F\eta_A = \epsilon_{\widetilde{F}(A,\alpha)} \circ FG(\theta) \circ F\eta_A = \epsilon_{\widetilde{F}(A,\alpha)}$$

In the first and last step, we used the algebra axiom for (A, α), in the second the adjunction axiom relating unit and counit, in the third a naturality square for ϵ.

Altogether, we have verified that $\widetilde{G}(\widetilde{F}(A,\alpha)) = (G\widetilde{F}(A,\alpha), G\epsilon_{\widetilde{F}(A,\alpha)}) \cong (A, \alpha)$.

Step 3: The counit $\widetilde{G} \circ \widetilde{F} \to \mathrm{id}_{\mathcal{D}}$ is an equivalence.
By definition, we have a coequaliser diagram computing $\widetilde{F}(\widetilde{G}(B))$:

$$FGFGB \;\underset{\epsilon_{FGB}}{\overset{FG\epsilon_B}{\rightrightarrows}}\; FGB \xrightarrow{\theta} \widetilde{F}(\widetilde{G}(B)) \tag{4}$$

By the universal property, the map $\epsilon_B : FGB \to B$ induces a map $\tau : \widetilde{F}(\widetilde{G}(B)) \to B$.

Applying the functor G to the entire situation, we obtain a diagram

$$\begin{array}{ccccc} GFGFGB & \underset{G\epsilon_{FGB}}{\overset{GFG\epsilon_B}{\rightrightarrows}} & GFGB & \longrightarrow & G\widetilde{F}(\widetilde{G}(B)) \\ & & & \searrow & \downarrow \\ & & & & GB \end{array}$$

The top line is a coequaliser as G preserves reflexive coequalisers. The diagram

$$GFGFGB \;\underset{G\epsilon_{FGB}}{\overset{GFG\epsilon_B}{\rightrightarrows}}\; GFGB \to \; GB$$

is a split coequaliser (cf. Proposition 3.18). Together, these facts imply that the map $G\widetilde{F}(\widetilde{G}(B)) \to GB$ is an isomorphism, which shows that $\widetilde{F}(\widetilde{G}(B)) \cong B$ as G is conservative. □

In fact, we have almost proven a sharp version of the Barr-Beck theorem. To state it, we need the following notion:

Definition 3.19. Given a functor $G : \mathcal{D} \to \mathcal{C}$, a parallel pair $d_0, d_1 : X_1 \rightrightarrows X_0$ is said to be G-split if $G(d_0), G(d_1) : X_1 \rightrightarrows X_0$ is a split pair in the sense of Definition 3.16.

We can now state the desired refinement:

Theorem 3.20 (Barr-Beck theorem, precise version)**.**
An adjunction $F : \mathcal{C} \leftrightarrows \mathcal{D} : G$ is monadic if and only if it has the following two properties:

a) $\mathcal{D}$ admits and G preserves coequalisers of G-split pairs; this means that whenever a pair $d_0, d_1 : X_1 \rightrightarrows X_0$ has the property that $G(X_1), G(X_0) : G(X_1) \rightrightarrows G(X_0)$ is part of a split coequaliser diagram, then $d_0, d_1 : X_1 \rightrightarrows X_0$ admits a colimit, which G preserves.
b) G is conservative (i.e. reflects isomorphisms).

Exercise 3.21. Taking inspiration from the proof of the crude Barr-Beck Theorem 3.14, prove Theorem 3.20.