Category Theory: Notes & Errata

- Lecture 17, Example (a): It is faster and more natural to observe directly that reflexive coequalisers are preserved. To see this, note that the coequaliser in \textbf{Set} of \( f, g : A \to B \) is \( B \) quotiented by the transitive closure of \( f(x) \sim g(x) \) for \( x \in A \), whilst in \textbf{Gp} it is \( B \) quotiented by the transitive closure of \( h_1(x_1)...h_n(x_n) \sim h'_1(x_1)...h'_n(x_n) \) where \( x_1...x_n \in A \) and each \( h_i \) and \( h'_i \) can be either \( f \) or \( g \).

Thus in general \( U \) will not preserve coequalisers. However we have a reflexive coequaliser, so for example given \( f(x)g(y) \sim g(x)f(y) \):

\[
\begin{align*}
g(rf(x).x^{-1}.rg(x).y) &= g rf(x).g(x)^{-1}.grg(x).g(y) \\
&= f(x).g(x)^{-1}.g(x).g(y) \\
&= f(x)g(y)
\end{align*}
\]

\[
\begin{align*}
f(rf(x).x^{-1}.rg(x).y) &= frf(x).f(x)^{-1}.frg(x).f(y) \\
&= f(x).f(x)^{-1}.g(x).f(y) \\
&= g(x).f(y)
\end{align*}
\]

Thus everything regarded as equal in the \textbf{Gp} coequaliser is also seen as equal in the \textbf{Set} coequalisers.

- 5.9 is the \textit{Precise Monadicity Theorem}.

- Lecture 19 page 10: Given \( A \in \text{Set} \) with this structure, we want to define \( \alpha : TA \to A \) to make a member of \( \text{Set}^T \). Now \( T \) is finitary, so as in 6.8 take inclusion \( I : \text{Set}_f \to \text{Set} \) and forgetful \( U : (I \downarrow A) \to \text{Set}_f \); then \( TA = \text{colim}TIU \), so we can define \( \alpha \) by specifying maps:

\[
\alpha_x : TIU(x : I\bar{n} \to A) \to A \\
\omega \mapsto \omega_A(x)
\]

Then the derived algebra has operations \( \omega(x) = \alpha(T\bar{x}(\omega)) = \alpha_{\bar{x}}(\omega) = \omega_A(x) \) as desired. It remains to check that \((A, \alpha) \in \text{Set}^T \).

- Lemma 7.9. It is not clear to us how to modify the functor to give the desired left adjoint. However it is possible to construct directly a left adjoint for \( f^* : \text{Sub}(B) \to \text{Sub}(B) \) by combining the proofs of the previous two results.

- Lecture 24 construction: Note we quotient out the morphisms but not the objects, so the ‘same’ morphism will in fact occur repeatedly between different isomorphic objects.