Category Theory exercise sheet 8

October 30, 2022

1 Category theory

1. So when do we have an isomorphism?

   Definition 1.0.1. A **split monomorphism** is a morphism $f : X \longrightarrow Y$ which is right inverse to some morphism $g : Y \longrightarrow X$. That is, $g \circ f = \mathrm{id}_X$.

   A **split epimorphism** is left inverse to some morphism $h : Y \longrightarrow X$. That is, $f \circ h = \mathrm{id}_Y$.

   Prove that an epic section is an isomorphism. Prove also that a monic retraction is also an isomorphism.

2. Show that a morphism $f : X \longrightarrow Y$ is a split epimorphism in a category $\mathscr{C}$ if and only if for all $Z \in \mathscr{C}$ post composition

   $$\begin{aligned} \mathrm{Hom}(Z, X) &\longrightarrow \mathrm{Hom}(Z, Y) \\ h &\longrightarrow f \circ h \end{aligned}$$

   defines a surjective function.

   Show that f is a split monomorphism if and only if for all $Z \in \mathscr{C}$ pre-composition

   $$\begin{aligned} \mathrm{Hom}(Y, Z) &\longrightarrow \mathrm{Hom}(X, Z) \\ h &\longrightarrow h \circ f \end{aligned}$$

   defines a surjective function.

2 Mathematics

1. What are the monomorphisms in the category of fields?

2. The following conceals some research-level problem.

   Consider the functors $\underline{\mathrm{Ab}} \longrightarrow \underline{\mathrm{Group}}$ (inclusion), $\underline{\mathrm{Ring}} \longrightarrow \underline{\mathrm{Ab}}$ (forgetting the multiplication), $(\_^{\times}) : \underline{\mathrm{Ring}} \longrightarrow \underline{\mathrm{Group}}$ (inclusion), $\underline{\mathrm{Mod}_R} \longrightarrow \underline{\mathrm{Ab}}$ (forgetful). Determine which functors are full, which are faithful, and which are essentially surjective. Do any define equivalence of categories?

3 Computer science

Speaking loosely, a computer program `F` induces a function IO(`F`) by its input/output behaviour. Given two programs `P`, `Q` it is in general too turse to analyse these two functions on the level of their underlying functions IO(`P`), IO(`Q`) because these two programs might differ in their complexity, or their resource consumption, or their side effects, etc. We will discuss *notions of computation* later in the course more thoroughly, but for now, we will attempt to model some of these concepts in the category $\underline{\mathrm{Set}}$ of sets and functions. For instance, non-termination can be modelled by introducing a formal symbole $\perp$ which represents failure.

$$\begin{aligned} \eta_X : X &\longrightarrow X \coprod \{\perp\} \\ x &\longmapsto x \end{aligned}$$

More generally, given a set E of *exceptions*, we can relate the set of values a program takes X to the same set along with its exceptions in the following way.

$$\begin{aligned} \eta_X : X &\longrightarrow X \coprod E \\ x &\longmapsto x \end{aligned}$$

Lastly, we can relate a set X to its nondeterministic counterpart as follows:

$$\begin{aligned} \eta_X : X &\longrightarrow \mathcal{P}_{\mathrm{fin}}(X) \\ x &\longmapsto \{x\} \end{aligned}$$

Clearly, all three of these maps are injective. By proving that a morphism f in $\underline{\text{Set}}$ is a monomorphism if and only if it is injective, show that in fact all these morphisms are monomorphisms.