6 Homework for Introduction to Category Theory (week 6)

6.1 Powerset monad

We consider the category $\mathbb{Set}$. Recall the powerset of a set X:

$$\mathcal{P}(X) = \{U \mid U \subseteq X\}.$$

1. Show that this extends to a functor $\mathcal{P}\colon \mathbb{Set} \to \mathbb{Set}$.

2. Show that is also extends to a contravariant functor $\mathcal{P}'\colon \mathbb{Set}^{\mathrm{op}} \to \mathbb{Set}$.

We will be using the covariant functor in the remainder of the exercise. You will show that $\mathcal{P}$ can be given a monad structure.

3. For each set X, guess sensible functions of the following type:

$$\eta_X\colon X \to \mathcal{P}(X)$$
$$\mu_X\colon \mathcal{P}(\mathcal{P}(X)) \to \mathcal{P}(X)$$

4. Show that $(\mathcal{P}, \eta, \mu)$ is a monad.

   Hint: There are two ways to prove this. You can show that η and μ are natural transformations and that the associativity ($\mu \circ \mathcal{P}(\mu) = \mu \circ \mu_{\mathcal{P}}$) and unitality ($\mu \circ \eta_{\mathcal{P}} = \mathrm{id}_{\mathcal{P}}$ and $\mu \circ \mathcal{P}(\eta) = \mathrm{id}_{\mathcal{P}}$) laws hold. Alternatively, you can show that $\mathcal{P}$ arises from an adjunction $F \dashv G$ to some other category. The alternative way requires a bit of creativity.

5. Try to describe the Kleisli category $\mathcal{K}\ell(\mathcal{P})$. More specifically, if we interpret statements of some programming language as arrows in $\mathcal{K}\ell(\mathcal{P})$, then what type of computation does it describe?

6. Show that $\mathcal{K}\ell(\mathcal{P}) \cong \mathbb{Rel}$. (Recall that $\mathbb{Rel}$ is the category of sets and relations between sets.)

6.2 Arrows between monads

In this exercise we will relate the powerset monad $(\mathcal{P}, \eta, \mu)$ as defined above with the distribution monad $(\mathcal{D}, \eta, \mu)$ as defined in the lecture. By abuse of notation we denote the units and multiplications of both monads by η and μ. (This should not be confusing as long as we draw the relevant diagrams.)

1. For each set X, define a function $\phi_X\colon \mathcal{D}(X) \to \mathcal{P}(X)$ which gives all elements with positive probability.

2. Show that this defines a natural transformation $\phi\colon \mathcal{D} \implies \mathcal{P}$.

3. Show that the following diagram (of natural transformations) commutes. Hint: It suffices to show that the diagram commutes for each set X.

$$\begin{array}{ccc} \mathcal{D}^2 & \xrightarrow{\phi_{\mathcal{D}}} & \mathcal{P}\mathcal{D} \\ {\scriptstyle \mathcal{D}(\phi)}\downarrow & & \downarrow{\scriptstyle \mathcal{P}(\phi)} \\ \mathcal{D}\mathcal{P} & \xrightarrow[\phi_{\mathcal{P}}]{} & \mathcal{P}^2 \end{array}$$

   This defines a natural transformation $\phi^2\colon \mathcal{D}^2 \implies \mathcal{P}^2$ (by either going down-right, or right-down in the diagram).

4. Show that the following two diagrams commute.

$$\begin{array}{ccc} \mathrm{id}_{\mathsf{Set}} & \xrightarrow{\eta} & \mathcal{D} \\ {\scriptstyle \mathrm{id}}\downarrow & & \downarrow{\scriptstyle \phi} \\ \mathrm{id}_{\mathsf{Set}} & \xrightarrow[\eta]{} & \mathcal{P} \end{array} \qquad\qquad \begin{array}{ccc} \mathcal{D}^2 & \xrightarrow{\mu} & \mathcal{D} \\ {\scriptstyle \phi^2}\downarrow & & \downarrow{\scriptstyle \phi} \\ \mathcal{P}^2 & \xrightarrow[\mu]{} & \mathcal{P} \end{array}$$

What this shows is that ϕ "behaves well" with the monad structures of $\mathcal{D}$ and $\mathcal{P}$. Such a natural transformations can be considered as arrows between monads, written as

$$\phi\colon (\mathcal{D}, \eta, \mu) \implies (\mathcal{P}, \eta, \mu).$$

5. Show that ϕ induces a functor

$$\Phi\colon \mathcal{K}\ell(\mathcal{D}) \to \mathcal{K}\ell(\mathcal{P}).$$

   Hint: Use the previous results to show that Φ preserves identities and compositions (as is required for a functor).

This last result means something for program semantics. If we are given a probabilistic program, where statements are interpreted by arrows in $\mathcal{K}\ell(\mathcal{D})$, then we automatically get a nondeterministic semantics via Φ (i.e., by forgetting the precise probabilities). It also shows that this abstract semantics is compositional (functoriality means that composition is preserved).