Functional Programming

BONUS 2 :: Questions

All of these questions can be completed with pen and paper. However, you may find it helpful to use your preferred text editor for some questions. These questions will be identified with (Code).

1 Monoids

1. (a) There is a monoid called the *free monoid*. It has this name because it allows you to give any type a monoid instance for free. Can you think what it is?

   (b) Prove that the *free monoid* is a monoid.

2. A semi–group is a type equipped with an associative binary operation i.e. a Monoid without an identity element.

   Write a type class to represent a Semigroup.

3. (Code) Define two Semigroup and Monoid instances for Booleans that can live in the same Haskell file together without Haskell getting upset.

4. † (a) What is the difference between $(xs +\!\!+\ ys) +\!\!+\ zs$ and $xs +\!\!+\ (ys +\!\!+\ zs)$?

   (b) Which way does Haskell bracket $xs +\!\!+\ ys +\!\!+\ zs$ by default?

5. † (a) (Code) Define the monoid instance for:

   newtype *Function a* = *Function* $(a \to a)$

   † (b) Prove that this obeys the monoid laws.

6. Is it possible to create an instance of a monoid that does not uphold the monoid laws? If so, why, and give an example by defining the instance and saying which laws it breaks. If not, explain how invalid instances are restricted.

† 7. A group $(G, *)$ is a set G together with a binary operation $* : G \to G \to G$ satisfying the following properties:

   1. (Associativity) $\forall\ x, y, z \in G$, $(x * y) * z = x * (y * z)$
   2. (Identity) $\exists\ e \in G$, such that $\forall\ x \in G$: $x * e = x = e * x$
   3. (Inverse) $\forall\ x \in G$, $\exists\ x^{-1} \in G$, such that: $x * x^{-1} = e = x^{-1} * x$

   (a) (Code) Define a type class for groups.

   (b) (Code) Define a sensible instance of your type class that upholds properties of a group.