Lab 2

Functional Programming (ITI0212)

2021.02.02

1. The type \texttt{One} from the lecture is built in to Idris as \texttt{Unit}. The type \texttt{Zero} from the lecture is built in to Idris as \texttt{Void}.
   
   (a) Write a function of type \texttt{Bool -> Unit}. How many different functions of this type are there?

   (b) How many functions are there of type \texttt{Bool -> Bool}? Write them all.

   (c) Write a function of type \texttt{Nat -> Unit}. How many different functions of this type are there?

   (d) How many functions are there of type \texttt{Unit -> Nat}? Write one of them down.

   (e) How many functions are there of type \texttt{Void -> Void}. Write them all down.

   (f) How many functions are there of type \texttt{Nat -> Void}? Write them all down.

   (g) How many functions are there of type \texttt{Void -> Nat}? Write them all down.

2. Recall the \texttt{Shape} type from the lecture:

   \begin{verbatim}
   data Shape : Type where
       Circle : Nat -> Shape
       Rectangle : Nat -> Nat -> Shape
       IsoTriangle : Nat -> Nat -> Shape
   \end{verbatim}

   with the idea being that \texttt{Circle k} is the circle of radius \(k\), \texttt{Rectangle a b} is the rectangle with length \(a\) and width \(b\), and \texttt{IsoTriangle a b} is the isosceles triangle with base width \(a\) (one side) and leg length \(b\) (two sides).

   (a) Write a function \texttt{area} : \texttt{Shape -> Double} that computes the area of a \texttt{Shape}.

   (b) Write a function \texttt{regular} : \texttt{Shape -> Bool} that returns \texttt{True} if the input \texttt{Shape} is regular (that is, all of its sides are of equal length), and returns \texttt{False} otherwise.
(c) Add a type constructor to the Shape type to represent regular $n$-sided polygons. Update your area and regular functions to account for this new type constructor.

(d) Is our representation of isosceles triangles a good one? Put another way, is it possible to specify every isosceles triangle in the way we have chosen? Does every instance of (IsoTriangle a b) : Shape give an isosceles triangle?

3. (a) Write a function \texttt{monus} : Nat -> Nat -> Nat that subtracts the second argument from the first. If the second argument is greater than the first, the result should be zero.

(b) Use pattern matching to write a function \texttt{even} : Nat -> Bool that returns \texttt{True} in case it’s input is an even number, and \texttt{False} otherwise.

(c) Write a function \texttt{odd} : Nat -> Bool that does the same, but for odd numbers.