

Programming Paradigms

Practical session, Week 8

Department of Informatics
University of Beira Interior

Chapter 10: Interactive Programming (IO ())

1. Redefine

```
putStr :: String -> IO ()
```

using a list comprehension and the library function

```
sequence_ :: [IO a] -> IO ()
```

2. Define an action

```
adder :: IO ()
```

that reads a given number of integers from the keyboard, one per line, and displays their sum. For example:

```
> adder
How many numbers? 5
1
3
5
7
9
The total is 25
```

Hint: start by defining an auxiliary function that takes the current total and how many numbers remain to be read as arguments. You will also likely need to use the library functions `read` (that casts strings to another type) and `show`.

3. Redefine `adderr` using the function

```
sequence :: [IO a] -> IO [a]
```

that performs a list of actions and returns a list of the resulting values.

4. Implement the game of `nim` in Haskell, where the rules of the game are as follows:

- For a given board size n , the board comprises of n rows of stars with the first row having n stars, the second $n - 1$ and so on. For example, for a board size of 5, the program should use and show the following board:

```
1: * * * * *
2: * * * *
3: * * *
4: * *
5: *
```

- Two players take turns to remove one or more stars from the end of a single row.
- The winner is the player who removes the last star or stars from the board.

Some points to consider:

- Represent the board as a list of n integers that give the number of stars remaining on each row. For example, the a board size of 5, the initial board is `[5, 4, 3, 2, 1]`.
- To represent a board use the type

```
type Board = [Int]
```

- Using recursion, define a

```
putBoard :: Board -> IO ()
```

that displays `nim` boards of any size.

Hint: first define an auxiliary function that takes the current row number as an additional argument.

Implement the full game. Make your code as simple and as readable as possible. It must follow a functional programming approach.