Task 2: Pretty Printing of Binary Numbers (Haskell - 8 Points)

May 22, 2022

Introduction

The goal of this assignment is to implement a Haskell program that reads a positive integer from the standard input, converts it into its binary representation, and displays this representation as a text image where each digit (i.e., 0 and 1) is represented as a 4x4 table of characters. More precisely, the text images of digits 0,1 are captured in Haskell as follows:

```haskell
type Img = [String]

zero :: Img
zero = [".##.",
        "#..#",
        "#..#",
        ".##."]

one :: Img
one =  ["...#",
        "..##",
        "...#",
        "...#"]
```

Implement a function `main :: IO ()` that works as follows:

1. It displays a message `Enter integer:`,
2. then let the user enter an integer *n* (you may assume only valid inputs),
3. converts *n* into its binary representation,
4. displays the binary representation using the above text-images. The particular digits must be separated by a column consisting of the character `'.'`.

Below you can see an example if you execute the main function and the user enters the number 12.

```
> main
Enter integer:\n
12
...#....#..##...##.\n
..##...##.#..#.#..#\n
...#....#.#..#.#..#\n
...#....#..##...##.\n
```

All the displayed lines must end with the new-line character `'\n'` as is depicted above. So you can display them e.g. by `putStrLn`. No trailing whitespaces are allowed at the ends of lines.