Introduction to Computer Science

Assignment 07

Constructor University,
Bremen, Germany

April 3, 2024

- Marks are only awarded for producing the correct solutions and justifying how you arrived at them.
- Please write legibly. Illegible answers will NOT be marked.

Question 1

Alice is in a room with two closed doors labelled "Tea Party" and "Rabbit Hole", respectively. The exit from Wonderland is behind one of these doors, and there is a trap behind the other door. In the same room, there are two Cheshire cats that can answer only yes/no questions. One of these cats always tells the truth while his twin brother always answers with falsehoods. Both Cheshire cats know which door is the exit, and which one is the trap. Alice can ask questions in sequence (one at the time) to the Cheshire cats in any order. Alice does not know which cat tells the truth and which does not.

1. Explain why the question: "*Do you tell the truth?*" will be answered "*yes*" independently of the cat that is asked.
   [2 marks]
2. Consider the question: "*If I ask the other cat whether he is the liar, what will be his answer?*". Explain how this question will be answered by each of the cats.
   [2 marks]
3. For this problem, an algorithm is a sequence of yes/no questions together with a decision of whether there is a next question, and if so, whether this next question is for the same cat that has answered or the other. Starting asking questions to any of the cats, present an algorithm (with the *minimum* number of questions) for finding out the exit door. **[3 marks]**

Question 2

An algebraic data type representation for *binary trees* in Haskell can be specified by:

```
data BinTree a = EmptyTree | Node a (BinTree a) (BinTree a)
deriving(Show)
```

A *binary search tree* (BST) is a binary tree that is either: (i) an *empty* tree or (ii) the left and right subtrees are BST with the elements of the left subtree less than or equal to the root node, and the elements of the right subtree greater than or equal to the root node. The following function, inserts elements in a binary tree respecting this BST order:

```
insertTree :: (Ord a) => a -> BinTree a -> BinTree a
insertTree x EmptyTree = Node x (EmptyTree) (EmptyTree)
insertTree x (Node y left right)
    | x == y    = Node x left right
    | x < y     = Node y (insertTree x left) right
    | x > y     = Node y left (insertTree x right)
```

Consider BSTs whose elements (if any) have been inserted using uniquely function `insertTree`. Define a Haskell function:

```
findBST :: :: (Ord a) => a -> BinTree a -> Bool
```

that tells us whether a given element is in a given BST.

[3 marks]