COMPUTER SCIENCE TRIPOS Part IA – 2020 – Paper 1

1 Foundations of Computer Science (avsm2)

You need to write OCaml code to help a local park ranger count the different types of trees present in a region of Cambridgeshire woodland.

(*a*) Define an OCaml type `tree` that can distinguish between an *oak*, *birch* or *maple* tree, and also any other species with an arbitrary `string` name.

[1 mark]

(*b*) Define two OCaml values with the following signatures:

(*i*) `val describe :  tree -> string`
that accepts a `tree` parameter and returns a human-readable `string`.

(*ii*) `val identify :  string -> tree`
that accepts a lowercase `string` parameter and returns a `tree`.

Explain briefly how the OCaml compiler can statically check if you have handled all the input possibilities for the input parameters to `describe` and `identify`.

[4 marks]

(*c*) Define a type `stree` that only distinguishes between three species *oak*, *birch* or *maple* and no others. Implement functions for the following signatures with similar functionality to the earlier `identify` function:

```
val identify_exn :  string -> stree
val identify_opt :  string -> stree option
```

Briefly discuss the tradeoffs between your two approaches.   [5 marks]

(*d*) You now need to implement a simple simulator before starting real surveys. Trees will occur in the following fixed infinite sequence: *oak, birch, oak, maple, maple*, and then repeat from the beginning.

(*i*) Define a function `val spotter :  unit -> stree` that will return the sequence of trees when called multiple times.   [5 marks]

(*ii*) Define a purely functional alternative `spotter` that calculates the next `stree` in sequence, using only the input arguments to the function to calculate the return value. Write down an example application of this function with the input arguments and the expected output result.*(Hint: you may need to pass in the complete sequence as one of the arguments.)*
[5 marks]