

COMPUTER SCIENCE TRIPOS Part IA – 2012 – Paper 1

1 Foundations of Computer Science (LCP)

This question has been translated from Standard ML to OCaml

Recall that a dictionary of $(key, value)$ pairs can be represented by a binary search tree. Define the *union* of two binary search trees to be any binary search tree consisting of every node of the given trees.

- (a) Write an OCaml function `union` to return the union of two given binary search trees. [Note: You may assume that they have no keys in common.] [6 marks]

Define a *slice* of a binary search tree to be a binary search tree containing every $(key, value)$ node from the original tree such that $x \leq key \leq y$, where x and y are the given endpoints.

- (b) Write an OCaml function `takeSlice` to return a slice – specified by a given pair of endpoints – from a binary search tree. [4 marks]

- (c) Write an OCaml function `dropSlice` to *remove* a slice from a binary search tree: given a tree and a pair of endpoints, it should return the binary search tree consisting of precisely the nodes such that $x > key$ or $key > y$. [Hint: First consider the simpler task of deleting a node from a binary search tree.] [8 marks]

- (d) The tree t need not be identical to that returned by

```
union (takeSlice (x, y) t)
      (dropSlice (x, y) t)
```

Briefly explain how such an outcome is possible. [2 marks]

[Note: All OCaml code must be explained clearly and should be free of needless complexity.]