SOLUTION NOTES

Foundations of Computer Science 2003 Paper 1 Question 6 (LCP)

This question assesses basic ML programming skills, and specifically Lecture 15 (Linked Data Structures). It also covers Lecture 8 (Dictionaries).

(a) Bookwork from Lecture 8. Each branch of the tree carries a $(key, value)$ pair. Its left subtree holds smaller keys; the right subtree holds greater keys. If the tree remains reasonably balanced, then update and lookup both take $O(\log n)$ for a tree of size n. Lookup in the binary search tree goes to the left subtree if the desired key is smaller than the current one and to the right if it is greater. Update is similar, and the new tree is recursively constructed. The path from the root to the affected node is copied and the rest of the new tree is shared with the old one.

(b) Here is the datatype of mutable binary trees.

```
datatype 'a mtree = Lf
                  | Br of 'a * 'a mtree ref * 'a mtree ref;
```

(c) Mutable binary search trees are ordered as above. Since the left and right subtrees are mutable, new tree nodes can be added without copying existing nodes. Each tree node will carry the pair of a key and a reference to a value; thus, an assignment statement can update an existing dictionary entry with no need for copying.

The lookup operation is almost identical to the one in the lecture notes. However, there are now references to the left and right subtrees and to the dictionary entries.

```
exception Missing of string;

fun lookup (Br ((a,rx),rmt1,rmt2), b) =
      if      b < a then  lookup(!rmt1, b)
      else if a < b then  lookup(!rmt2, b)
      else !rx
  | lookup (Lf, b) = raise Missing b;
```

The update operation is differs from the one in the lecture notes.

```
fun update (rmt, b:string, y) =
  case !rmt of
    Br ((a,rx),rmt1,rmt2) =>
      if      b < a then  update(rmt1,b,y)
      else if a < b then  update(rmt2,b,y)
      else (*a=b*) rx := y
  | Lf => (rmt := Br((b, ref y), ref Lf, ref Lf));
```

If the key is found, we update the dictionary entry. If it isn't, then we have a pointer to a leaf, which we update with a new node.