1999 Introduction to Functional Programming

p12q10
GMB

Model Answer

1.
```
datatype 'a tree = Lf
                 | Br of 'a*'a tree*'a tree;
```

2.
```
fun post Lf              = []
  | post (Br(v,t1,t2)) = post t1 @ post t2 @ [v];
```

3.
```
fun post2 t = let fun postord (Lf, vs) = vs
                    | postord (Br(v,t1,t2), vs) =
                          postord (t1, postord (t2, v::vs))
              in
                 postord(t,[])
              end;
```

4. We prove that $\forall t, l. post(t) \mathbin{@} l = postord(t, l)$, by induction on t.

   (a) Base case. $t = Lf$.

$$\begin{aligned} post(Lf) \mathbin{@} l &= [\,] \mathbin{@} l \\ &= l \\ &= postord(Lf, l) \end{aligned}$$

   (b) Inductive case. $t = Br(v, t1, t2)$.

$$\begin{aligned} post(Br(v,t1,t2)) \mathbin{@} l &= (post(t1) \mathbin{@} (post(t2) \mathbin{@} [v])) \mathbin{@} l \\ &= post(t1) \mathbin{@} (post(t2) \mathbin{@} ([v] \mathbin{@} l)) \\ &= post(t1) \mathbin{@} (post(t2) \mathbin{@} (v :: l)) \\ &= post(t1) \mathbin{@} (postord(t2, v :: l)) \\ &= post(t1, postord(t2, v :: l)) \\ &= postord(Br(v, t1, t2), l) \end{aligned}$$

5.
```
datatype 'a mtree = MTree of 'a * ('a mtree) list;
```

6.
```
fun map(f,[])=[]
  | map(f,x::xs)=f(x)::map(f,xs);

fun foldr(f,[],e)     = e
  | foldr(f,x::xs,e) = f(x,foldr(f,xs,e));

fun post3 (MTree(n,[])) = [n]
  | post3 (MTree(n,ts)) = foldr(op@,map(post3,ts),[])@[n];
```