

CMSE 462 Assignment 6

(Prolog - III) Tree structures in Prolog using complex terms

Program to be done in groups of two. Pick your partner!

A tree can be represented in Prolog in the following way.

- An empty tree is represented by the atom **empty**.
- A tree with a left child **l** and a right child **r** as node value **v** is represented by **node(l,v,r)**.

Some predicates are given for you below for manipulating binary search trees with this representation.

```
insert(V,empty,node(empty,V,empty)). /* insert a value into a binary search tree */
insert(V, node(L,V2,R), node(L2,V2,R)):- V=<V2, insert(V,L,L2).
insert(V, node(L,V2,R), node(L,V2,R2)):- V>V2, insert(V,R,R2).
```

```
insert_list([],empty). /* insert a list of numbers into an empty binary search tree */
insert_list([H|T],Tree):- insert_list(T, Temp), insert(H,Temp,Tree).
```

```
size(empty, 0). /* size of the tree - how many nodes it has */
size(node(L,_,R),S):- size(L,X), size(R,Y), S is X+Y+1.
```

```
in_order(empty,[]). /* inorder traversal */
in_order(node(L,V,R),Res):- in_order(L,LT), in_order(R,RT), app(LT,[V],X),
app(X,RT,Res).
```

```
app([],L,L). /* append two lists */
app([H|T],L,[H|TL]):- app(T,L,TL).
```

```
min(node(empty,V,empty),V). /* find the minimum value in a binary search tree */
min(node(L,_,R),V):- min(L,V).
```

Implement the following predicates on binary search trees.

find(V,T) succeeds if value V is in the T

post_order(T,L) succeeds if L list containing the values in the tree T in the same order as in a "postorder" traversal.

inner_nodes(T,L) succeeds if L is a list containing the values in the inner nodes of the tree T.

max(T, Max) succeeds if Max is the maximum value in tree T (hint: the maximum is "rightmost value" in the tree)

min(T, Min) succeeds if Min is the minimum value in tree T (hint: the minimum is "rightmost value" in the tree)

height(T,H) succeeds if the height of the tree (length of the longest path from the root to any leaf) is H.