

Jeffrey Lansford

11/30/2020

Prolog Lab

We are task to write a Prolog program that will transverse through a Binary tree represented as a list to find 1) the unique leaves of the tree, and 2) the longest path of the tree or depth.

```
% Jeffrey Lansford
% Prolog Lab
% 11/30/20
% Prolog program to find unique leaves of a binary tree
%   and to find the depth of the binary tree

% 1) Tree Unique Function
% Unique function that returns a list with only unique atoms
% return an empty list for an empty list
uniq([], []).
% if head is in list, then continue with the tail
uniq([H|T], L) :-
    member(H, T), !,
    uniq(T, L).
% append head into tail of the list with recursive
uniq([H|T], [H|L]) :-
    uniq(T, L).

% Flatten function to flatten a list
% return empty list for an empty list
flatten([], []).
% return the atom of if list is only an atom
flatten(X, [X]) :-
    atom(X), !.
% flatten head and tail and append the two list together
flatten([H|T], Z) :-
    flatten(H, T1),
    flatten(T, T2),
    append(T1, T2, Z).

% Function to append list
append([], L, L).
append([H|T], L, [H|Z]) :-
    append(T, L, Z).

% Function to provide a list of unique leaves of a binary tree
```

```

% return a empty list for an empty tree
mytreeunique([], []).
% flatten list and then find unique leaves
mytreeunique(L, X) :-
    flatten(L, Z),
    uniq(Z, X).

% 2) Longest Path
% Function to get longest path of a binary tree
% return zero depth for an empty list
mydepth([], X) :-
    X is 0.
% return zero depth for an only a tree with one atom
mydepth(Z, X) :-
    atom(Z),
    X is 0, !.
% do depth on head and tail of the list, then find max and add one.
mydepth([H|T], X) :-
    mydepth(H, A),
    mydepth(T, B),
    Y is max(A, B),
    X is Y+1.

```

```

?- mytreeunique([a,[b,[a,[c,d]]]],X).
X = [b, a, c, d] .

?- mytreeunique([a,[a,[b,[a,[c,[d,e]]]]]],X).
X = [b, a, c, d, e] .

?- mytreeunique([a,a,[a,[a,[a,[a,[a,[a,[b]]]]]]]],X).
X = [a, b] .

```

```

?- mydepth([a,[b,[a,[c,d]]]],X).
X = 8 .

?- mydepth([a,[a,[b,[a,[c,[d,e]]]]]],X).
X = 12 .

?- mydepth([a,[b,[c]]],X).
X = 5 .

```