CLP Final Report 44251017 Huang Jiahui

```
% Define the decision tree structure as a fact
my_tree(tree(outlook, [
    value(sunny, tree(humidity, [
        value(high, leaf(no)),
        value(normal, leaf(yes))
    ])),
    value(overcast, leaf(yes)),
    value(rainy, tree(windy, [
        value(true, leaf(no)),
        value(false, leaf(yes))
    ]))
])).

% Entry point for visualizing the tree
visualize_tree(Tree) :- visualize_tree(Tree, 0).

% Print a tree node with indentation
visualize_tree(tree(Attribute, Children), Indent) :-
    print_indent(Indent),                % Print leading spaces for indentation
    write('Node: '), write(Attribute), nl,
    visualize_children(Children, Indent + 2).  % Increase indentation for children

% Print a leaf node with indentation
visualize_tree(leaf(Label), Indent) :-
    print_indent(Indent),
    write('Leaf: '), write(Label), nl.

% Print all children nodes (list of value/2 terms)
visualize_children([], _).
visualize_children([value(Value, SubTree)|Rest], Indent) :-
    print_indent(Indent),                % Print indentation for the branch value
    write('['), write(Value), write(']'), nl,
    visualize_tree(SubTree, Indent + 2),  % Recursively print the subtree with
increased indentation
    visualize_children(Rest, Indent).     % Continue with the rest of the children

% Print Indent number of spaces
print_indent(0).
print_indent(N) :- N > 0, write(' '), N1 is N - 1, print_indent(N1).
```

result:

```
yes
| ?- my_tree(T), visualize_tree(T).
Node: outlook
  [sunny]
    Node: humidity
      [high]
        Leaf: no
      [normal]
        Leaf: yes
  [overcast]
    Leaf: yes
  [rainy]
    Node: windy
      [true]
        Leaf: no
      [false]
        Leaf: yes
T = tree(outlook,[value(sunny,tree(humidity,[value(high,leaf(no)),value(normal,leaf(yes))])),value(overcast,leaf(yes)),v
alue(rainy,tree(windy,[value(true,leaf(no)),value(false,leaf(yes))]))]) ?
```