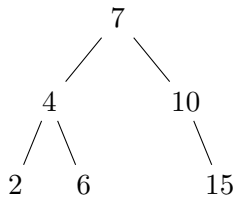
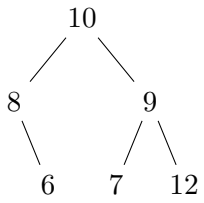
Theory of programming languages

Formal Methods in computer science

Assignments 1: Functional modules in Maude — Academic year 2021/2022

A *binary tree* is a data structure in which each node has at most two children. Each node store a different *key* that we'll assume to be a natural number. When each internal node of the binary tree store a key greater than all the keys in the node's left subtree and less than those in its right subtree, the binary tree is called *binary search tree*. For example, the binary tree on the left is a binary search tree, whereas the binary tree on the right isn't.





In this assignment we will specify in Maude both, binary trees and binary search trees. In particular we ask for:

1. Define sorts for binary trees and binary search trees.
2. Define the subsort relation between the sorts and the membership axiom(s) required to distinguish binary search trees from binary trees.
3. Define the following operators:
   (a) a total function `empty` that indicates if a binary tree is empty or not.
   (b) a total function `contains` that given a binary tree and a key indicates if the key is in the tree.
   (c) a partial function `root` that gives the key of the root of the tree.
   (d) a total function `depth` that gives the depth of the binary tree (e. g., both trees in the example above have depth 3).
   (e) a total function `width` that gives the width of the binary tree (e. g., both trees in the example above have width 6).
   (f) a total function `add` that given a binary search tree and a key, adds the key to the tree. If the key already exists this function doesn't change the binary search tree.
   (g) a total function `del` that given a binary search tree and a key, modifies the tree removing that key. If the key is not inside the tree, the function doesn't change it.
4. Check that binary trees and binary search trees are assigned the correct sort.
5. Check the operators.