Theory of programming languages

Universidad Complutense de Madrid   Academic year 2020/2021

Term rewriting in Maude - Functional modules

Exercise 1 Specify binary search trees in Maude:

(a) Define the sorts for `BTree` (binary trees) and `BSTree` (binary search tree).

(b) Define the subsort relation between the sorts.

(c) Define a unique constructor for each sort. Both values and keys are natural numbers.

(d) Define the membership axiom(s) required to distinguish binary search trees from binary trees.

(e) Define a total function `contains` that checks whether a key appears in a binary tree (it might not be a *search* tree).

(f) Define a total function `add` that, given a binary search tree, a key, and a value, adds the key and the value to the tree. If the key already exists this function adds the previous value to the new one.

(g) Define a partial function `remove` that receives a binary search tree and a key and returns the tree once the element has been removed. If the element is not in the tree the function fails.

Exercise 2 Test that the specification works:

(a) Check that binary trees and binary search trees are assigned the correct sort.

(b) Check all functions. In particular, check what happens with `remove` when an error must be returned.