Project 1: ordered trees

Stefan Blom

30.3.2007

Problem

We consider three types of ordered trees:

- AVL trees
- (2,4) trees
- Red Black trees

For background information on those trees, look at the material on e-campus[1]. Some code fragments are available on the course main site[2]. Hickey's book contains an insert for Red-Black trees.

For each of these trees, you must implement:

- `insert` takes a tree and an element as arguments and returns a tree whose elements are the elements of the old tree plus the new element.
- `remove` takes a tree and an element as arguments and returns a tree. If the element does not occur in the given tree then the given tree is returned. Otherwise a new tree is returned whose elements are the elements of the old tree minus *one* occurrence of the new element.
- `get` takes a tree as argument and returns `None` if the tree is empty and `Some(a,t)` otherwise, where `a` is the least element of the given tree and `t` is a tree whose elements are the elements of the given tree minus one occurrence of `a`.

To test these functions, you must implement insertion sort:

[1] `http://e-campus.uibk.ac.at/`

[2] `http://cl-informatik.uibk.ac.at/teaching/ss07/fp/course_material.php`

- `insertion_sort` has 4 arguments:
  - empty constant
  - insert function
  - get function
  - list

  It returns a sorted version of the list, which it obtains by first inserting all elements into the empty structure and then building a list by getting all elements from the structure:

  ```
  s := empty
  for e in list do
    s := insert s e
  end
  list := []
  loop
    case get(s) of
      | None -> return list
      | Some(a,t) -> list.append(a) ; s := t
  end
  ```

  Please note that you own implementation should be recursively defined and *not use loops*.

Rules

- You can work alone or in pairs.
- If you work alone then you may choose to skip either (2,4) trees or Red Black trees.
- A working implementation means a passing grade.
- Elegance and readability count as well.
- Carefully selected auxiliary functions help to reduce your workload and improve readability and elegance.