Software Workshop 1 - Lecture 4 & 5

Ossama Edbali

January 27, 2015

1 Fold

Fold is an expressive way for applying a function in a cumulative way. The fold function replaces `Cons` with `f` and `Nil` with `b` (which is the initial value). The polymorphic type of the function fold is: `f: 'a -> 'b -> 'b`.

2 Binary trees

Binary trees can be created using an interface and two implementations:

- Interface file: `Tree`
- Base case implementation: `EmptyTree<E>`
- Inductive case implementation: `MakeTree<E>`

Binary trees have a root, right and left subtrees. Every node has at most 2 children where nodes with zero children are called leaves.

The selectors are:

- `root` for getting the root of a BT
- `left` for getting the left subtree.
- `right` for getting the right subtree.

One might use the Maybe type in order to deal with these selectors (in case we have an empty tree we would return a `Nothing` object).

3 Binary Search Trees

BSTs are a special case of binary trees where the values of the nodes in the left subtree must be smaller than the root (viceversa for the right subtree).

The class/interface layout is:

- Interface file: `Bst`
- Base case implementation: `Empty<E>`
- Inductive case implementation: `Fork<E extends Comparable<E>>`

4 Notes on generics

Generics are a way in Java to implement polymorphic methods and classes.

```
public class List<E> {
    ...
}
```

In the class List one can use the type `E` in the methods declaration. But if we want to use a different type than `E` in a method declaration we must use this:

```
...
public <B> fold(Function<E, B> f, B init) {
    ...
}
```