Polymorphic Trees

You should use polymorphism to avoid such repetitions.

How should you avoid such repetitions?

Here, you will have to write a separate version for each of the above tree.

For e.g., if you need a function to compute, say, the number of nodes in a

data Structure = Tree Shape | Interbranch InterTree

But if you do this, you may have to repeat it for other types of data.

data Structure = Tree Shape | Interbranch InterTree

You may want to store some data in your trees. E.g., store numbers in leaf

nodes:

2. A branch node with two leaf-type children is also a tree.
1. A leaf node is a tree.

Recursive Data Structure: Tree

A binary tree consists of leaf nodes and branch (internal) nodes:

• A leaf node has no children.
• A leaf node is just that, a leaf node.

data Tree = leaf | branch (branch, leaf) leaf

This is coded in Haskell as:

Here is how it looks like conceptually:

Branch (Branch (Branch (leaf, leaf), leaf), leaf)

Here is an example tree expression:

Example Tree

Branch (Leaf, leaf)

Leaf
To stuff one type of data into branches and another type into leaves:

- \( \text{data \ tree a \ (tree a \ (tree a))} \)

- \( \text{data \ tree a = \ tree a} \)

To stuff data into both kinds of nodes:

- \( \text{data \ tree a = \ Leaf a} \)

- \( \text{data \ tree a \ = \ Leaf a \ \ (Leaf a \ \ (Leaf a \ \ (Leaf a \ \ Leaf a)))} \)

- \( \text{data \ tree a \ = \ Leaf a \ \ (Leaf a \ \ Leaf a \ \ (Leaf a \ \ Leaf a \ \ Leaf a))} \)

- \( \text{data \ tree a \ = \ Leaf a \ \ (Leaf a \ \ Leaf a \ \ Leaf a \ \ Leaf a \ \ Leaf a \ \ Leaf a \ \ Leaf a \ \ Leaf a \ \ Leaf a)} \)

It looks like:

\[
\text{Example:}
\text{Leaf} 1 \quad \text{Leaf} 2 \quad \text{Leaf} 3
\]

Functions for Polymorphic Trees

- \( \text{totaltree} \ \ \text{tree a} \) in a tree:
  \begin{align*}
  \text{totaltree} \ \ \text{tree a} & = \ \ \text{totaltree} \ \ \text{tree a} \ \ \text{tree a} \\
  & = \ \ \text{totaltree} \ \ \text{tree a} \ \ \text{tree a} \\
  & = \ \ \text{totaltree} \ \ \text{tree a} \ \ \text{tree a}
  \end{align*}

The parameter \( \text{tree a} \) has the type variable \( a \) because we do not care what type the leaves are in the leaves.

In a tree:

- \( \text{totaltree} \ \ \text{tree a} \) is a function that counts the number of nodes (both leaves and branches).