

99 questions/Solutions/54A

From HaskellWiki

< 99 questions | Solutions

(*) Check whether a given term represents a binary tree

In Haskell, we characterize binary trees with a datatype definition:

```
data Tree a = Empty | Branch a (Tree a) (Tree a)
    deriving (Show, Eq)
```

The above tree is represented as:

```
tree1 = Branch 'a' (Branch 'b' (Branch 'd' Empty Empty)
                                (Branch 'e' Empty Empty))
        (Branch 'c' Empty
        (Branch 'f' (Branch 'g' Empty Empty)
                    Empty)))
```

Other examples of binary trees:

```
tree2 = Branch 'a' Empty Empty    -- a binary tree consisting of a root node only
tree3 = nil                       -- an empty binary tree
tree4 = Branch 1 (Branch 2 Empty (Branch 4 Empty Empty))
        (Branch 2 Empty Empty)
```

The type system ensures that all terms of type
Tree a

are binary trees: it is just not possible to construct an invalid tree with this type.
Hence, it is redundant to introduce a predicate to

check this property: it would always return
True

Retrieved from "https://wiki.haskell.org/index.php?title=99_questions/Solutions/54A&oldid=36036"

-
- This page was last modified on 13 July 2010, at 21:41.
 - Recent content is available under a simple permissive license.