99 questions/Solutions/68

From HaskellWiki

< 99 questions | Solutions

Preorder and inorder sequences of binary trees. We consider binary trees with nodes that are identified by single lower-case letters, as in the example of problem P67.

a) Write predicates preorder/2 and inorder/2 that construct the preorder and inorder sequence of a given binary tree, respectively. The results should be atoms, e.g. 'abdecfg' for the preorder sequence of the example in problem P67.

b) Can you use preorder/2 from problem part a) in the reverse direction; i.e. given a preorder sequence, construct a corresponding tree? If not, make the necessary arrangements.

c) If both the preorder sequence and the inorder sequence of the nodes of a binary tree are given, then the tree is determined unambiguously. Write a predicate pre_in_tree/3 that does the job.

```
treeToPreorder :: Tree Char -> String
treeToPreorder = preorder
    where preorder Empty = ""
          preorder (Branch x l r) = x : preorder l ++ preorder r


treeToInorder :: Tree Char -> String
treeToInorder = inorder
    where inorder Empty = ""
          inorder (Branch x l r) = inorder l ++ x : inorder r

-- Given a preorder string produce a binary tree such that its preorder string
-- is identical to the given one.
preToTree :: String -> Tree Char
preToTree "" = Empty
preToTree (c:cs) = Branch c Empty (preorderToTree cs)

-- Given a preorder and an inorder string with unique node chars produce the
-- corresponding binary tree.
preInTree :: Monad m => String -> String -> m (Tree Char)
preInTree [] [] = return Empty
preInTree po@(x:xs) io = do (lio,_:rio) <- return $ break (== x) io
                            (lpo,rpo) <- return $ splitAt (length lio) xs
                            l <- preInTree lpo lio
                            r <- preInTree rpo rio
                            return $ Branch x l r
preInTree _ _ = fail "woops"
```