Homework 3 Hanoi Towers with monads

It is possible to solve the problem using the ST Monad. These are the types that are needed:

type State = (Int, Config)

type Info = (Report, [Move])

newtype ST a = S (State -> (Info, State)

As usual, to "open" an ST value we need the function app:

app :: ST a -> State -> (a,State)

app (ST x) s = x s

Using the type ST a, we can define a new function check with this type:

check :: [Move] -> ST info

The function check uses a function move that tries the next move and that may have the following type:

move :: Move -> ST Report

In case the move has been executed successfully, move returns ST with Ok, otherwise it returns ST with Bad.

In my solution move calls 2 functions, one that checks all possible violations of the current move, returning a value of type ST Bool, and another function, to be called only when the first function guarantees that the move is ok, that executes the move returning the new Config. Since Config :: [[Int]], the function that executes the move uses heavily the operator !! that picks the elements of a list. Recall that, p !! 0, returns the first element of the list p.