Exercise on the use of pushdown stack for parentheses matching

A pushdown stack is modelled by [a] and the operations that apply to a pushdown stack are pop and push. The type of pop is pop :: [a] ->(a,[a]). For instance, when a is char, pop ['a','b','c']= ('a',['b','c']). The type of push is: push :: a -> [a]->((),[a]), and again if a=Char, push 'z' ['a','b'] =((), ['z','a','b']). Observe that push only changes the pushdown and thus the first component of the pair is the empty value ().

The exercise asks to write a program that uses a pushdown automaton in order to check whether a given string is correctly balanced with respect to the round parentheses, i.e., that for each open there is a corresponding closed parenthesis ')' that follows it and, symmetrically, that for each closed parenthesis ')' there is a corresponding open parenthesis '(' that precedes it. For instance the string: "ab)c (" is not correctly balanced, whereas "ac(ab(s)xxxxx)aaaa" is correctly balanced.

We want to write two programs that perform this task. The first one uses the following types:

type Stack=[Char]

pop :: Stack ->(Char,Stack)

push :: Char -> Stack -> ((),Stack)

check_par :: [Char]->Stack->Bool

--takes a string and a Stack and says True/False whn the string is/is not correctly balanced

The second solution must use the ST Monad. The types that are needed are:

type Stato= [Char]

newtype ST a = S(Stato->(a, Stato)),

pop1 :: ST Char

push1 :: ST ()

check_par1 :: [Char] -> ST Bool

check_par1 must be written either with use of >>= or with the do syntax.

Notes:

a) The stack is used in this way: when a '(' is encountered then one pushes a ')', when a ')' is encountered, then one pops a ')'. All other symbols are ignored.

b) one must be careful with the base case: the recursion should terminate when the string is empty and the check of the parentheses is successful when the stack is also empty.

c) there may be strings that cause a pop with empty stack. This will raise a system exception. The exercise does not require that this problem is dealt with.