An implementer's variable $A$ holds a binary tree representation as follows. If the tree is empty, $A = [\text{nil}]$. If the tree has left subtree $L$ and right subtree $R$ and root value $n$, then $A = [L; n; R]$. The tree

```
   3
  / \
 2   7
```

is represented as $A = [[[\text{nil}]; 2; [[\text{nil}]; 5; [\text{nil}]OPSIS[[\text{nil}]; 7; [\text{nil}]]]]$. The tree must be reimplemented using implementer's variable $B$ as follows. If the tree is empty, $B = 0$. If the tree has left subtree $L$ and right subtree $R$ and root value $n$, then

$B = \text{"left"$\rightarrow L | "root"$\rightarrow n | "right"$\rightarrow R$

The same example tree is represented as

```
B = \text{"left"$\rightarrow ("left" \rightarrow 0 \\
| \text{"root"$\rightarrow 2 \\
| \text{"right"$\rightarrow ("left" \rightarrow 0 \\
| \text{"root"$\rightarrow 5 \\
| \text{"right"$\rightarrow 0 ) ) \\
| \text{"root"$\rightarrow 3 \\
| \text{"right"$\rightarrow ("left" \rightarrow 0 \\
| \text{"root"$\rightarrow 7 \\
| \text{"right"$\rightarrow 0 ) }
```

(a) What is the data transformer?
(b) A user has natural variable $n$ and the operation

```
root \equiv n := A 1
```

which assigns to $n$ the root value. Use your transformer from part (a) to transform $root$. 

```