

Universidade Federal de Goiás
INF - Instituto de Informática

Disciplina: Programação Funcional
Curso: Ciências da Computação

Professor: Daniel Ventura
Data: 02/12/2024

Atividade 9: **Raciocinando sobre Programas** (data de entrega: 09/12/2024)

1. Considere a seguinte declaração de tipos para árvores binárias rotuladas:
- ```
data Arv a = Vazia | No a (Arv a) (Arv a)
```

Dada a função para listar por ordem os elementos numa árvore binária:

```
listar :: Arv a → [a]
listar Vazia = []
listar (No x esq dir) = listar esq ++ [x] ++ listar dir
```

Empregue a técnica para eliminar concatenações apresentada na aula teórica para derivar uma versão mais eficiente desta função.

*Sugestão:* sintetize uma definição recursiva da função auxiliar `listarAcc :: Arv a → [a] → [a]` tal que `listarAcc t xs = listar t ++ xs`.

2. Considere o tipo `Arv a` definido no exercício anterior.
- (a) Defina uma função `soma`, que calcule a soma dos valores da árvore (numérica) dada.
  - (b) Defina a função `foldArv :: (a → b → b → b) → b → Arv a → b` tal que `foldArv f v t` se comporte como a função `foldr`, mas opere sobre árvores.
  - (c) Mostre que, para qualquer árvore numérica `t`, `soma t = foldArv soma3 0 t`, onde `soma3 x y z = x + y + z`.
3. Usando indução sobre listas, mostre para quaisquer funções `f` e listas finitas `xs` e `ys` que:
- (a) `map f (xs++ys) = map f xs++map f ys`.
  - (b) `map f (reverse xs) = reverse (map f xs)`.
- Sugestão:* use a propriedade provada no item anterior.