

1. Determine o tipo da expressão abaixo.

(1,0) (a) $(\text{map } (\backslash x \rightarrow \text{length } x)).(\text{foldr } (/) 1)$

2. Um móbile é constituído de pendentes, fios e barras. Em cada uma das extremidades de uma barra de um mobile é preso um fio, no qual pode estar pendurado um pendente ou uma nova barra. Barras e fios são considerados elementos sem peso e pendentes possuem um peso (representado por um valor inteiro). O seguinte tipo de dado pode ser usado para representar um móbile:

$\text{data Mobile} = \text{Pendente Int} \mid \text{Barra Mobile Mobile}$

O peso de um mobile é igual à soma dos pesos de todos os seus pendentes. Um móbile é balanceado se ele consiste de um único pendente ou se os pesos dos móveis pendurados nas duas extremidades da sua barra são iguais e esses móveis são balanceados. Defina as seguintes funções sobre móveis:

(1,5) (a) $\text{peso} :: \text{Mobile} \rightarrow \text{Int}$, que retorna o peso do móbile dado como argumento.

(1,5) (b) $\text{balanceado} :: \text{Mobile} \rightarrow \text{Bool}$, que determina se o móbile dado como argumento é ou não balanceado

3. Leia a descrição abaixo e, em seguida, faça o que se pede.

Um fabricante de sorvetes contratou você para simular parte do processo de produção deles. Na produção, a mistura de dois ingredientes (o aromatizante e o espessante) acontece apenas quando o recipiente de mistura refrigerado (RMR) está disponível, ou seja, eles são retirados de diferentes depósitos quando podem ser efetivamente misturados. Para o RMR ficar disponível, é necessário que ele seja esvaziado, o que acontece com o giro do RMR a fim de retirar o sorvete. Assim, as operações de retirada do sorvete e de mistura dos ingredientes precisam do RMR de forma exclusiva. Defina as operações para misturar os ingredientes e esvaziar o RMR. Considere que os ingredientes ficam guardados em depósitos distintos e que vamos diferenciar a quantidade que utilizamos de cada um deles. Além disso, vamos abstrair o tempo necessário para misturar ingredientes e retirar sorvete do recipiente.

(3,5) (a) Implemente, em Haskell, o que foi descrito acima, utilizando *mutable variables* (MVar) e memória transacional (TVar), como você achar necessário.

(2,5) (b) Apresente uma solução em Java. Pode-se utilizar classes da API de concorrência de Java, não envolvendo tipos primitivos atômicos.

$(.) :: (b \rightarrow c) \rightarrow (a \rightarrow b) \rightarrow a \rightarrow c$
 $\text{map} :: (a \rightarrow b) \rightarrow [a] \rightarrow [b]$
 $\text{foldr} :: (a \rightarrow b \rightarrow b) \rightarrow b \rightarrow [a] \rightarrow b$