

Laboratórios de Informática I

2024/2025

Licenciatura em Engenharia Informática

Ficha 2
Programação Funcional em Haskell

Funções não recursivas

Crie um ficheiro "Ficha2.hs" para escrever as funções a definir na aula de hoje. Relembre que no mesmo ficheiro não poderá ter duas funções diferentes com o mesmo nome. Sempre que for pedida a reformulação de uma função anterior, terá de alterar o nome da função. O mesmo acontece com o nome dos tipos de dados.

Tarefas

1. Considere o seguinte tipo para representar movimento unitário (deslocamento de uma unidade num dos eixos de um sistema Cartesiano):

```
data Movimento = Norte | Sul | Este | Oeste deriving Show
```

Vamos representar pontos como coordenadas num plano cartesiano:

```
type Ponto = (Double, Double)
```

- (a) Defina uma função que recebe um ponto e um movimento e calcula as coordenadas do ponto após o movimento:

```
move :: Ponto -> Movimento -> Ponto
```

- (b) Defina uma função que calcule a distância entre dois pontos:

```
dist :: Ponto -> Ponto -> Double
```

- (c) Defina uma função que recebe dois pontos e calcula qual dos pontos se encontra mais a sul. Em caso de igualdade, escolha um deles.

2. Considere uma janela quadrangular e um sistema de coordenadas Cartesiano com origem no canto inferior esquerdo dessa janela. Reescreva a função `move` definida anteriormente de forma a garantir que o ponto não se desloca para fora da janela. Para além do ponto e do movimento a realizar, a função recebe o comprimento do lado da janela.

3. Defina uma função que recebe um ponto¹ e o comprimento do lado da janela, e calcula quais seriam as coordenadas desse ponto se a origem passasse a ser o canto superior esquerdo da janela. Considere que não há coordenadas negativas.
4. Repita o exercício anterior para o caso de a origem passar a estar no centro da janela.
5. Considere agora um ponto que se desloca numa linha reta horizontal a velocidade constante. Vamos representar o vetor velocidade por um valor do tipo `Double`, positivo se o movimento for da esquerda para a direita e negativo se for em sentido contrário. Defina uma função que recebe um ponto, um vetor velocidade e um intervalo de tempo, e calcula as coordenadas do ponto ao fim desse tempo.

```
type Velocidade = Double
type Tempo = Double
move :: Ponto -> Velocidade -> Tempo -> Ponto
```

6. Repita o exercício anterior para o caso de um movimento vertical a uma velocidade constante (positiva no movimento ascendente e negativa no movimento descendente).
7. Repita o exercício anterior, assumindo que o vetor velocidade v é descrito pelas componentes nos eixos vertical e horizontal (v_x, v_y) .

```
type Velocidade = (Double, Double)
move :: Ponto -> Velocidade -> Tempo -> Ponto
```

8. Considere o seguinte tipo de dados para representar figuras geométricas num plano:

```
data Figura =
    Circulo Ponto Double |
    Rectangulo Ponto Ponto |
    Quadrado Ponto Double    deriving (Show,Eq)
```

sendo um círculo descrito pelo centro e raio, um rectângulo descrito pelos dois vértices de uma diagonal e um quadrado descrito pelo vértice superior esquerdo e o lado. Os lados do quadrado e do rectângulo são paralelos aos eixos.

- (a) Defina uma função que dado um ponto e uma figura geométrica, verifique se o ponto está dentro (incluindo as margens) ou fora dessa figura.
- (b) Defina uma função que dada uma figura calcula o menor quadrado que a contém. Caso a figura recebida seja um rectângulo, o canto superior esquerdo do rectângulo deverá ser o canto superior esquerdo do quadrado resultado.

```
menorQuadrado :: Figura -> Figura
```

¹Com as coordenadas no sistema do exercício anterior

- (c) Defina uma função que dada uma figura calcula o maior círculo nela contido. Caso receba um rectângulo, o círculo resultado deverá estar encostado ao lado esquerdo e ao lado superior do rectângulo.

`maiorCirculo :: Figura -> Figura`

- (d) Defina uma função que recebe duas figuras geométricas do mesmo tipo e verifica se a primeira está contida na segunda (e.g. se o primeiro círculo está contido no segundo círculo). Se as figuras recebidas forem de tipo diferente (um círculo e um quadrado, por exemplo), deve retornar False.

`contida :: Figura -> Figura -> Bool`

- (e) Repita a função anterior para o caso de as figuras poderem ser de tipo diferente.
- (f) Defina uma função que recebe uma figura geométrica e um factor de escala, e calcula a nova figura geométrica (ampliada ou reduzida). Se a figura for um círculo, o centro deve ficar inalterado. Se for um quadrado ou um rectângulo, o vértice inferior esquerdo deve permanecer inalterado.

`zoom :: Figura -> Double -> Figura`

9. Resolva outros exercícios da Ficha 1 de Programação Funcional.