

# Aula Teórico-prática 8

## Programação Funcional

LEI 1º ano

1. Considere as seguintes definições:

```
type Ponto = (Float,Float)  -- (abscissa,ordenada)
type Poligonal = [Ponto]    -- linha poligonal
type Vector = (Float,Float) -- vector aplicado na origem

data Figura = Rectangulo Ponto Float Float  -- canto superior esq.,largura, altura
            | Triangulo Ponto Ponto Ponto   -- os 3 vértices
```

```
distancia :: Ponto -> Ponto -> Float
distancia (a,b) (c,d) = sqrt (((c-a)^2) + ((b-d)^2))
```

- (a) Defina uma função que calcule o comprimento de uma linha poligonal.
- (b) Defina uma função que converta um elemento do tipo `Figura` na correspondente linha poligonal.
- (c) Defina funções para
  - i. testar se uma linha poligonal é fechada;
  - ii. testar se uma linha poligonal corresponde a um triângulo;
  - iii. testar se uma linha poligonal corresponde a um retângulo;
- (d) Defina uma função `poli2Fig :: Poligonal -> Maybe Figura` que converte (se possível) uma linha poligonal numa figura.
- (e) Defina uma função `triangula` que, dada uma linha poligonal fechada e convexa, calcule uma lista de triângulos cuja soma das áreas seja igual à área delimitada pela linha poligonal.
- (f) Suponha que existe uma função `areaTriangulo` que calcula a área de um triângulo.

```
areaTriangulo (x,y,z) = let a = distancia x y
                          b = distancia y z
                          c = distancia z x
                          s = (a+b+c) / 2  -- semi-perimetro
                          in sqrt (s*(s-a)*(s-b)*(s-c))  -- formula de Heron
```

Usando essa função, defina uma função que calcule a área delimitada por uma linha poligonal fechada e convexa.

- (g) Defina a função `translacao :: Vector -> Poligonal -> Poligonal` que faz a translação da linha poligonal segundo o vector dado.
- (h) Defina uma função `zoom2` que, dada uma linha poligonal, dê como resultado uma linha poligonal semelhante e com o mesmo ponto inicial mas em que o comprimento de cada segmento de recta é multiplicado por 2. Por exemplo, o rectângulo `[(1,1), (1,3), (4,3), (4,1), (1,1)]` deverá ser transformado em `[(1,1), (1,5), (7,5), (7,1), (1,1)]`
- (i) Defina uma versão mais geral da função anterior, por forma a que seja possível fazer zoom-in ou zoom-out da figura e com o factor de ampliação ou de redução que se quiser.

2. Uma representação alternativa para as linhas poligonais consiste em representar estas linhas com um ponto de origem e uma sequência de vectores.

`type Poly = (Ponto, [Vector])`

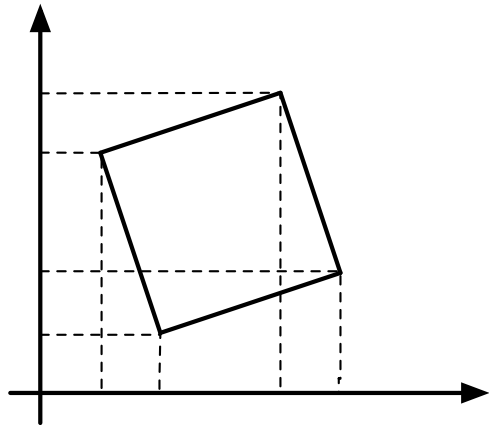
Assim, o quadrado

`[(2,1), (5,2), (4,5), (1,4), (2,1)]`

poderia ser representado pelo ponto inicial (2,1) e pela seguinte lista de vectores

`[(3,1), (-1,3), (-3,-1), (1,-3)]`

Note que nesta representação, as funções de translação e duplicação de uma linha poligonal são muito mais simples.



- Use a função `zipWith` (pré-definida) para definir a função de conversão de elementos do tipo `Poligonal` em `Poly`.
- Defina ainda a função que faz a conversão inversa.
- Use estas funções de conversão para redefinir as funções de translação e zoom de poligonais.