

Polimorfismo

Até então, as funções que criamos possuíam tipos específicos. Por exemplo, funções para strings que retornam strings e não podem ser usadas com inteiros. Essas funções são classificadas como monomórficas e são diferentes de algumas funções pré-definidas como `head`, `tail`, `elem` e várias outras. Se tentarmos ver o tipo dessas funções pré definidas, vamos encontrar algo como `(head :: [a] -> a)`. O "a", na definição de `head`, significa "qualquer tipo". `Head` e várias outras funções polimórficas, são capazes de receber elementos de qualquer tipo e retornar elementos desse mesmo tipo.

Outras funções polimórficas sobre listas:

`length` retorna o tamanho de uma lista

`length [1,2,3]` retorna 3

`reverse` retorna a lista invertida

`reverse "abc"` retorna "cba"

`elem` retorna um booleano informando se um elemento faz parte de uma lista

A função abaixo é uma implementação da função `head`

```
returnHead :: [ a ] -> a
```

```
returnHead ( x:xs ) = x
```