O$_L$X^{UM}— Um micro sistema de leilões em Haskell

Universidade do Minho
LEI —Licenciatura de Engenharia Informática
UC8201N6 — Lab. de Informática I
Ano lectivo de 2012/13

Nov. 2012

1 Tipos

O$_L$X^{UM} é um sítio para compra e venda em suporte internet que funciona em regime de leilão. Começa por oferecer uma colecção de items para serem leiloados,

$$\begin{array}{l}\textbf{data}\ House = House\ \{\\ \quad hrunning :: Running,\\ \quad hfinished :: Finished\\ \quad \}\ \textbf{deriving}\ (Show, Eq)\end{array}$$

onde cada

$$\textbf{type}\ Running = [Auction]$$

é a lista dos ítens ainda por leiloar, e

$$\textbf{type}\ Finished = [Auction]$$

contem os ítens cujo leilão já acabou. Cada ítem do leilão é registado de acordo com a seguinte estrutura,

$$\begin{array}{l}\textbf{data}\ Auction = Auction\ \{\\ \quad actid\ \ \ :: Int,\\ \quad actowner :: String,\\ \quad actdesc\ \ :: String,\\ \quad actvalue\ \ :: Int,\\ \quad actbidder :: String\\ \quad \}\ \textbf{deriving}\ (Show, Eq, Ord)\end{array}$$

a saber:

- $actid$ — o inteiro que identifica o ítem que está/foi leiloado
- $actowner$ — o identificador do seu dono
- $actdesc$ — a sua descrição
- $actvalue$ — o seu valor
- $actbidder$ — o identificador de quem o quer comprar

1.1 Exemplos

Dá-se de seguida um exemplo de leilão:

```
myhouse = House [a2, a1] [a3] where
  a1 = Auction 1 "Peter" "TV" 60 ""
  a2 = Auction 2 "Mary" "laptop" 120 ""
  a3 = Auction 3 "John" "phone" 85 "Peter"
```

Este exemplo pode ser usado para testes.

2 Acções

Colocar novo ítem no leilão:

```
auctionNew :: House → String → String → Int → House
auctionNew h owner desc value =
  let newid = length (hrunning h) + length (hfinished h) + 1
      newa = Auction newid owner desc value ""
  in House (newa : hrunning h) (hfinished h)
```

Oferta para um particular item do leilão:

```
auctionBid :: House → Int → String → Int → House
auctionBid h id bidder bid =
  let r = filter ((≢ id) ∘ actid) (hrunning h)
      curr = head (filter ((≡ id) ∘ actid) (hrunning h))
      newa = Auction (actid curr) (actowner curr) (actdesc curr) bid bidder
  in House (newa : r) (hfinished h)
```

Fechar o leilão de um ítem, dando-o como leiloado ou cancelado, caso não tenha havido nenhum *actbidder* interessado em comprá-lo:

```
auctionFinish :: House → Int → House
auctionFinish h id =
  let r = filter (λi → (actid i) ≢ id) (hrunning h)
      curr = filter (λi → (actid i) ≡ id) (hrunning h)
  in House r (curr ++ (hfinished h))
```

3 Tipos e funções auxiliares

Função que dá o total já leiloado, total por leiloar e total sem oferta:

```
tots :: House → Tot
tots h = Tot f r l where
  r = foldr (+) 0 (map actvalue (hrunning h))
  f = foldr (+) 0 [actvalue x | x ← hfinished h, actbidder x > ""]
  l = foldr (+) 0 [actvalue x | x ← hfinished h, actbidder x ≡ ""]
```

onde

```
data Tot = Tot {tfin :: Int, trun :: Int, tlost :: Int} deriving Show
```

Função genérica para ordenar sequências de registos segundo um dos seus atributos:

```
sortOn :: (Ord b) ⇒ (a → b) → [a] → [a]
sortOn f l = [l !! i | i ← map snd x] where x = sort (zip (map f l) [0..])
```