

Laboratorul 8 - Programare Logică și Funcțională

Seria 36

Martie 2024

În acest laborator vom începe să studiem limbajul Haskell. Pentru a putea executa programele, vom utiliza mediul de programare GHC (Glasgow Haskell Compiler), care include două componente: GHCi (interpretor) și GHC (compiler).

Lista comenzilor GHCi:

1. `ctrl + s` - clear prompt;
2. `ctrl + c` - break;
3. `:m + "module name"` - import module;
4. `:m - "module name"` - remove module;
5. `:t` - type;
6. `:?` - help;
7. `:l` - load;
8. `:q` - quit.

Având un interpretor deschis, putem evalua direct expresiile:

```
Prelude> 2 + 3
5
Prelude> False || True
True
Prelude> x
<interactive>:10:1: error: Variable not in scope: x
Prelude> x = 3
Prelude> x
3
Prelude> y = x + 1
Prelude> y
4
```

Pentru reprezentarea listelor, avem aceeași scriere ca în Prolog, cu mențiunea că, în Haskell, elementele unei liste trebuie să fie de același tip. Avem la dispoziție funcțiile `head` și `tail`:

```
Prelude> head [1,2,3]
1
Prelude> head "abcd"
'a'
Prelude> tail "abcd"
'bcd'
```

Pentru a interoga tipurile de date, folosim comanda `:t`.

```
Prelude> :t True
True :: Bool
```

Fișierele pe care le vom scrie au extensia `.hs` și pot fi editate cu orice editor. Fie următorul conținut al unui fișier pe care îl denumim `myFile.hs`:

Exerciții:

1. Definiți o funcție care să calculeze suma a trei numere.
2. Definiți o funcție care să determine maximul a patru numere.
3. Modelați jocul **Rock, Paper, Scissors**, definind un tip de date **Choice**, definit astfel:

```
data Choice
  = Rock
  | Paper
  | Scissors
deriving (Eq, Show)
```

(precizarea `deriving (Eq, Show)` ne spune că, pentru datele de acest tip, avem definite, în mod natural, relația de egalitate și o instanță pentru a obține un șir de caractere)

Definiți, în mod similar, un tip algebric **Result** cu variantele **Victory**, **Defeat** și **Draw**.

Implementați funcția `game :: Choice -> Choice -> Result` care întoarce **Victory** când primul argument câștigă, **Defeat** când al doilea câștigă, și **Draw** în caz de egalitate.