

# Programare Declarativa — Test 2

Nume: \_\_\_\_\_ grupa: \_\_\_\_\_

16 dec 2016

Se dă tipul de date:

```
data Dom = Empty          -- interval vid          {}
  | Ran a a                -- interval inchis       [a,b]
  | (Dom a) :|: (Dom a)    -- reuniunea a 2 intervale A U B
  | (Dom a) :&: (Dom a)    -- intersectia a 2 intervale A ∩ B
  deriving (Show, Read)
```

reprezentând un domeniu format din reuniuni și intersecții de intervale închise.

Exemplu:

```
exem :: Dom Int          -- ([1,3] U [2,4]) ∩ ([3,5] ∩ {})
exem = (((Ran 1 3) :|: (Ran 2 4)) :&: ((Ran 3 5) :&: Empty))
```

## CERINTE

*Atentie:* anumite funcții necesită constrângeri de tipuri, domeniile fiind definite pe tipuri ordonate de date.

**2p 1)** Scrieți funcția `exist` care primește o valoare de tip `a`, un domeniu de tip `a` și returnează o valoare booleană reprezentând apartenența valorii la domeniul respectiv

```
exist 3 ((Ran 1 3) :|: (Ran 6 10)) == True      -- 3 ∈ [1,3] U [6,10]
exist 5 ((Ran 1 3) :|: (Ran 6 10)) == False     -- 5 ∉ [1,3] U [6,10]
```

**2p 2)** Scrieți funcția `overlap` care primește ca argumente două domenii de tip `a` și returnează o valoare booleană — `True` dacă cele 2 intervale se intersectează, `False` în caz contrar. Se va considera doar cazul de intersecție a două domenii de tip `Ran a a`, pentru celelalte cazuri returnați `False`.

```
overlap (Ran 1 4) (Ran 3 5) == True             -- [1,4] ∩ [3,5] = [3,4]
overlap (Ran 1 4) (Ran 5 6) == False            -- [1,4] ∩ [5,6] = {}
overlap Empty (Ran 3 4) == False                 -- {} ∩ [3,4] = {}
```

**2p 3)** Scrieți funcția `normalize` care primește ca argument un domeniu de tip `a` și îl normalizează prin aplicarea proprietății de distributivitate:  $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$

```
normalize (((Ran 1 2) :|: (Ran 3 4)) :&: (Ran 2 3))
== (Ran 1 2 :&: Ran 2 3) :|: (Ran 3 4 :&: Ran 2 3)
```

**1.5p 4)** Scrieți funcția `optimize` care primește ca argument un domeniu normalizat de tip `a` și îl optimizează prin calculul intersecțiilor și reuniunilor, inclusiv eliminarea valorii `Empty`.

Puteți folosi funcția `overlap` și orice altă funcție din librăria standard.

```
input:  ( ([1,5] :&: [0,7]) :|: ([3,9] :&: [0,7]) ) :|: ( ([8,11] :&: [9,14]) :|: {} )
output: [1,7] :&: [9,11]
```

**1.5p 5)** Scrieți modulul `Principal` în care citiți un domeniu și afișați acest domeniu optimizat. *Atenție:*

Domeniu derivă `Show` și `Read`. Signatura funcției `main` este: `main :: IO (Dom Int)`

```
Principal> main          -- asteapta input
(((Ran 1 2) :|: (Ran 3 4)) :&: (Ran 3 5)) -- introducem input-ul
Ran 3 4                   -- output-ul functiei afisat la consola
```