

## Test de laborator

### Instrucțiuni:

La finalizarea examenului, se va pregăti un singur fișier cu extensia `.pl` ce va cuprinde rezolvările tuturor problemelor. Denumirea fișierului va fi următoarea: `NrVarianta_Grupa_Nume_Prenume.pl` (exemplu: `2_141_Ionescu_Raluca.pl`).

Dacă ați lucrat în varianta web, **nu** salvați pagina web, ci copiați manual codul ca text în fișier.

Fișierul va cuprinde pe primele rânduri numărul variantei, grupa și numele complet sub formă de comentarii.

Fișierul se va încărca la final, o singură dată, accesând următorul link:

<https://tinyurl.com/LMC1706>

1. (1 punct) Definiți un predicat `expand_intervals/2` care, pentru o listă de perechi de numere naturale, calculează lista `R` formată în felul următor: Pentru orice `i` mai mic decât lungimea lui `L`, dacă `(N, M)` este perechea de pe poziția `i` din `L`, atunci pe poziția `i` din `R`, se va afla lista tuturor numerelor mai mari strict ca `N` și mai mici sau egale cu `M`.

#### Exemplu:

```
?- expand_intervals([(1, 3), (5, 5), (4, 5), (5, 3), (2, 6)], R).  
R = [[2, 3], [], [5], [], [3, 4, 5, 6]].
```

2. (1,5 puncte) Definiți un predicat `div_concat/2` astfel încât, pentru orice liste `L, R`, `div_concat(L, R)` este adevărat dacă și numai dacă `L` și `R` sunt distincte și există două liste `L1` și `L2`, astfel încât:

- concatenând `L2` la `L1` obținem `L`,
- concatenând `L1` la `L2` obținem `R`,
- cel puțin una dintre `L1` și `L2` are o lungime care divide lungimea lui `L`.

#### Exemplu:

```
?- div_concat([1, 2, 3, 4, 5, 6, 7, 8, 9], R).  
R = [2, 3, 4, 5, 6, 7, 8, 9, 1]  
R = [4, 5, 6, 7, 8, 9, 1, 2, 3]  
R = [7, 8, 9, 1, 2, 3, 4, 5, 6]  
R = [9, 1, 2, 3, 4, 5, 6, 7, 8]
```

```
?- div_concat([1, 2, 3], [2, 1, 3]).  
false
```

3. (1,5 puncte) Considerăm în continuare reprezentarea formulelor logicii propoziționale folosită în laboratorul 5.
  - (a) (0,75 puncte) Scrieți un predicat `assoc_or` care, primind ca argument două formule `Phi, Psi`, este adevărat dacă și numai dacă `Phi` și `Psi` sunt formate doar din disjuncții și variabile și, în

plus,  $\Psi$  se poate obține din  $\Phi$  prin reasocierea parantezelor din ea (dar păstrând ordinea în care apar variabilele în disjuncții).

**Exemplu:**

```
?- assoc_or(sau(sau(a, b), c), sau(a, sau(b, c))).
true
?- assoc_or(sau(sau(a, b), sau(c, d)), sau(a, sau(b, sau(c, d)))).
true
?- assoc_or(sau(a, b), sau(b, a)).
false
?- assoc_or(a, a).
true
?- assoc_or(si(a, si(b, c)), si(si(a, b), c)).
false
```

- (b) (0,75 puncte) Scrieți un predicat `assoc_or_right` care, primind ca argument două formule  $\Phi$ ,  $\Psi$ , formate doar din disjuncții și variabile, este adevărat dacă și numai dacă  $\Psi$  este rezultatul reasocierii la dreapta a tuturor parantezelor din  $\Phi$  (dar păstrând ordinea în care apar variabilele în disjuncții).

**Exemplu:**

```
?- assoc_or_right(sau(a, sau(b, c)), Psi).
Psi = sau(a, sau(b, c))

?- assoc_or_right(sau(sau(a, b), c), Psi).
Psi = sau(a, sau(b, c))

?- assoc_or_right(sau(sau(a, b), sau(b, c)), Psi).
Psi = sau(a, sau(b, sau(b, c)))

?- assoc_or_right(sau(a, sau(sau(b, c), d)), Psi).
Psi = sau(a, sau(b, sau(c, d)))

?- assoc_or_right(sau(a, sau(b, c)), sau(sau(a, b), c)).
false

?- assoc_or_right(sau(a, sau(sau(b, c), d)), sau(sau(a, b), sau(c, d))).
false
```