

Laborator 8

- Sudoku

În acest laborator vom implementa predicate cu ajutorul cărora să completăm automat o tablă de sudoku.

Sudoku este un joc de tip puzzle reprezentat printr-o grilă formată din 81 de casute (9 casute pe linii și 9 casute pe coloane). Casutele fie sunt goale, fie au câte o cifră din intervalul [1,9]. Jucătorii trebuie să însereze cifre, din același interval, în casutele goale astfel încât acea cifră să nu mai existe pe linia sau coloana care include casuta, respectiv să nu se repete în cadrul de dimensiune 3x3 care include casuta.

	1	2	3	4	5	6	7	8	9
1	1							1	
2									
3			1						
4									
5									
6	1								
7									
8									
9									

Fig 1. Dacă există cifra 1 la poziția 1,1 în sudoku, înseamnă că această cifră nu mai poate fi adăugată nicaieri pe prima linie, prima coloană sau în primul cadran de 3x3 casute.

O configurație inițială de sudoku va fi reprezentată în Prolog astfel:

```
P=[ [7,2,5,9,6,1,4,8,3],  
    [3,6,9,4,2,8,1,7,5],  
    [4,8,1,3,7,5,9,2,6],  
    [9,1,6,2,8,4,5,3,7],  
    [5,7,3,6,1,9,2,4,8],  
    [2,4,n,7,5,3,6,9,1],  
    [1,9,4,n,3,6,7,5,2],  
    [6,3,7,5,4,2,n,1,9],  
    [8,5,2,1,n,7,3,6,4]].
```

Casutele goale sunt reprezentate prin caracterul n.

In fisierul sudokuLabStudenti.pl aveti un cod, incomplet, pentru sudoku. Voi aveti de implementat urmatoarele predicate:

%% TOATE CIFRE

toateCifre/1 % aritate 1

Acest predicat verifica daca in matricea de sudoku avem doar cifre.

Puteti folosi predicatul number/1 pentru a face deosebirea intre cifre si literele folosite pentru a marca spatiile goale.

%% CAUTA LINIE CU POZITIE GOALA

%membruGol/5 % aritate 5

membruGol(Linie, Sudoku,1,PozitieLinie, PozElementGol)

- **Linie** este linia din **Sudoku** care are cel putin o pozitie goala. Porneste cautarea de la **1**.
- **PozitieLinie** ne spune numarul liniei la care **Linie** se afla in **Sudoku**.
- **PozElementGol** ne spune pozitia casutei goale in **Linie**.
- **PozitieLinie** si **PozElementGol** sunt linia si coloana primului element gol din configuratia de sudoku. Considerand exemplul de sudoku de mai sus, **PozitieLinie** va fi 6, iar **PozElementGol** va fi 3.

%% VERIFICA PATRATE

%verificaPatratele/6

verificaPatratele(L1,L2,L3,Start,Stop,E)

- L1,L2,L3 sunt cele 3 linii care contin cadranul (patratul) de 3x3.
- Start reprezinta prima coloana a cadranului.
- Stop este ultima coloana a cadranului.
- E este elementul care nu vrem sa se repete in cadran

%% VERIFICA LINII

%extrageLinie/3

extrageLinie(Coloane, PozElementGol, ColoanaLinie)

- Coloane este matricea de sudoku care a fost transpusa.
- Despre PozElementGol stim din predicatul membruGol ca reprezinta coloana la care se afla o casuta goala. Aceasta, dupa ce a fost transpusa matricea de sudoku, reprezinta acum numarul liniei la care se afla acea casuta goala.
- In **ColoanaLinie** vom stoca linia din matricea **Coloane** care se afla la pozitia **PozElementGol**.

%%

%extrageElemente/2

extrageElemente(Linie, Elemente)

- Data fiind o lista, **Linie**, care contine si cifre si litere, acest predicat va intoarce in lista **Elemente** doar cifrele din aceasta lista. Puteti folosi predicatul number/1.

%% ADAUGA

%adpos/4

adpos(PozElementGol, X, Linie, LinieNoua)

- Linie este o lista
- **LinieNoua** va fi lista rezultata in urma adaugarii lui **X** la pozitia **PozElementGol** in **Linie**.