

Nagyhatékonyságú Deklaratív Programozás

GYAKORLÓ FELADATOK

CLPFD – aritmetikai és reifikált korlátok

Írd meg az alábbi fejkommenteknek megfelelő Prolog eljárásokat! Törekedj minél hatékonyabb megoldásra! Használd a SICStus CLP(FD) könyvtárát!

1. % sudoku_simple(?Matrix, +N): Matrix egy N*N-es mátrix, amely 1 és N közé eső számokból áll. Minden sorban és oszlopban a számok páronként különbözőek. Az eljárás címkézzen!

```
| ?- Mx = [[3,1|_|_|], sudoku_simple(Mx, 3).  
Mx = [[3,1,2],[1,2,3],[2,3,1]] ? ;  
Mx = [[3,1,2],[2,3,1],[1,2,3]] ? ;  
no
```
2. % magic(+N, ?Mx): Mx egy N oldalhosszú bűvös négyzet, vagyis az 1..N*N számokat pontosan egyszer tartalmazza, és minden sor, oszlop és átló összege ugyanaz. Az eljárás címkézzen!

```
| ?- magic(3, LL).  
LL = [[2,7,6],[9,5,1],[4,3,8]] ? ;  
LL = [[2,9,4],[7,5,3],[6,1,8]] ? ;  
LL = [[4,3,8],[9,5,1],[2,7,6]] ? ;  
LL = [[4,9,2],[3,5,7],[8,1,6]] ? ;  
LL = [[6,1,8],[7,5,3],[2,9,4]] ? ;  
LL = [[6,7,2],[1,5,9],[8,3,4]] ? ;  
LL = [[8,1,6],[3,5,7],[4,9,2]] ? ;  
LL = [[8,3,4],[1,5,9],[6,7,2]] ? ;  
no
```
3. % p(A,B,C): ha az A-B szám 3-mal osztva 1 maradékot ad, akkor C páros, különben páratlan. A p/3 eljárás ne címkézzen!

Írd meg ezt a korlátot reifikáció nélkül is, p2/3 néven!

```
| ?- domain([A,B], 1, 3), C in 0..1, p(A, B, C), labeling([], [A,B,C]), write(A-B-C), nl, fail.  
1-1-1  
1-2-1  
1-3-0  
2-1-0  
2-2-1  
2-3-1  
3-1-1  
3-2-0  
3-3-1  
no
```
4. Egy számlistában lokális szélsőértéknek hívunk egy elemet, ha mindkét szomszédjánál határozottan nagyobb, vagy mindkéténél határozottan kisebb.

% szeszam(+L, ?K): az L, csupa különböző elemből álló listában levő lokális szélsőértékek száma K. Az eljárás ne hozzon létre választási pontot (ne címkézzen)!

```
| ?- L=[1,_,_,_], domain(L, 1, 4), szeszam(L, 2), labeling([], L).  
L = [1,3,2,4] ? ;  
L = [1,4,2,3] ? ; no
```
5. Egy számlistában balról láthatónak hívunk egy elemet, ha az határozottan nagyobb az összes öt megelőző elemnél.

% latszam(+L, ?K): az L listában levő balról látható elemek száma K. Az eljárás ne hozzon létre választási pontot (ne címkézzen)!

```
| ?- L=[_,_,2,_], domain(L, 1, 4), all_distinct(L), latszam(L, 3), labeling([], L).  
L = [1,3,2,4] ? ; no
```
6. % panorama(+N, +Latvanyok, ?Lakotelep): Lakotelep egy N*N-es mátrix, amely egy lakótelep alaprajzát adja ki. A mátrix elemei az egyes épületek magasságát mutatják. Minden sorban és minden oszlopban különböző magasságúak az épületek és ezek a magasságok az 1..N tartományból kerülnek ki. Latvanyok egy olyan lista, amelynek elemei bal(I,K), felul(J,K), jobb(I,K), alul(J,K) alakú Prolog kifejezések, ahol I, J és K egyaránt az 1..N intervallumba esik. A Latvanyok listában előforduló bal(I,K) elem azt a korlátozást fejezi ki, hogy a lakótelep I-edik sorát balról nézve K ház látszik, a felul(J,K) azt, hogy a lakótelep J-edik oszlopát felülről nézve K ház látszik, stb. Az eljárás sorolja fel az összes megoldást. Csak a labeling/2 könyvtári eljárás hívása hozzon létre választási pontot!

```
| ?- panorama(4, [bal(1,2),bal(3,2),felul(2,3),felul(4,3),jobb(2,3),jobb(4,1),alul(1,3),alul(3,3)], Mx).  
Mx = [[3,1,4,2],[4,2,3,1],[2,4,1,3],[1,3,2,4]] ? ; no
```