

Nagyhatékonyságú Deklaratív Programozás
3. gyakorlat

2012. 10. 18.

Írd meg az alábbi fejkommenteknek megfelelő Prolog eljárásokat! Törekedj minél hatékonyabb megoldásra! Használd a SICStus clpfd könyvtárát!

A 2. feladattól kezdve szükség lehet a reifikáció ill. logikai korlátok használatára, segítségül alább bemásoljuk az előadásdiák vonatkozó részleteit.

--- REIFIKÁCIÓ: KORLÁTOK TÜKRÖZÉSE ---

Egy korlát tükrözése (reifikációja):

- a korlát igazságértékének "tükrözése" egy 0-1 értékű korlát-változóban; jelölése: $C \# \Leftrightarrow B$, jelentése: B tartománya 0..1, és B csakkor 1, ha C igaz.
- példa: $(X \# \geq 3) \# \Leftrightarrow B$ jelentése: B az $X \geq 3$ egyenlőség igazságértéke (B eleve a 0..1 intervallumra szűkül, 0 = hamis, 1 = igaz).

Megjegyzések

- Az ún. formula-korlátok (az eddig ismerttetett aritmetikai és halmaz-korlátok) mind tükrözhetőek.
- A globális korlátok (pl. `all_different/1`, `all_distinct/1`) nem tükrözhetőek.
- A tükrözött korlátok is "közönséges" korlátok, csak definíciójuk és végrehajtásuk módja speciális.
- Példa: a 0..5 tartományon a $(X \# \geq 3) \# \Leftrightarrow B$ korlát teljesen megegyezik a $B \# = X/3$ korláttal.

--- LOGIKAI KORLÁTOK ---

Logikai korlát argumentuma lehet:

- egy B változó, B automatikusan a 0..1 tartományra szűkül;
- egy tetszőleges tükrözhető aritmetikai- vagy halmazkorlát;
- egy tetszőleges logikai korlát.

A logikai korlátok:

#\ Q	negáció
P #/\ Q	konjunkció
P #\ Q	kizáró vagy
P #\/ Q	diszjunkció
P #=> Q	implikáció
Q #<= P	implikáció
P #<=> Q	ekvivalencia

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 ne hozzon létre választási pontot (ne címkézzen)!

```
| ?- Mx = [[3,1|_|_], sudoku_simple(Mx, 3), append(Mx, _V), labeling([],_V).  
Mx = [[3,1,2],[1,2,3],[2,3,1]] ? ;  
Mx = [[3,1,2],[2,3,1],[1,2,3]] ? ; no
```

2. Egy számlistában lokális szélsőértéknek hívunk egy elemet, ha mindkét szomszédjánál határozottan nagyobb, vagy mindkettőnél határozottan kisebb.

% szeszam(+L, ?K): az L listában levő szélsőértéknek 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
```

3. 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
```

4. Az alábbi feladat megoldását megelőzően javasoljuk "lakótelepi panoráma" típusu rejtvények megoldását, ilyenek találhatóak pl. itt:

http://5mp.eu/fajlok/logikairejtveny/lakoteleplb_www.5mp.eu_.pdf
http://5mp.eu/fajlok/logikairejtveny/lakotelep.560.568_www.5mp.eu_.pdf

% 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, sít.

% Az eljárás sorolja fel az összes megoldást, úgy hogy csak a
% `labeling/2` könyvtári eljárás hívása hoz 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
```

5. % square(+N, +XSumList, +YSumList, ?Res):- Res egy N*N-es mátrix,
% XSumList valamint YSumList pedig N hosszú listák. A Res mátrix elemei
% az 1..N intervallumba esnek, és teljesül rájuk, hogy a (vízszintesen
% vagy függőlegesen) oldalszomszédos elemek mindig különbözőek. Res
% sorainak összegét rendre XSumList elemei adják. Hasonlóan, Res
% oszlopainak összegei YSumList-ben vannak felsorolva. Továbbá, ha
% bármely sorból illetve oszlopból kiválasztunk három egymást követő
% (A,B,C) elemet, akkor ezekre teljesül az alábbi tulajdonság: ha az A-B
% szám 3-mal osztva 1 maradékot ad, akkor C páros, különben páratlan.

% Az eljárás sorolja fel az összes megoldást, úgy hogy csak a
% `labeling/2` könyvtári eljárás hívása hoz létre választási pontot!

```
| ?- square(4, Res, [12,_,_,8], [6,_,_,_]).  
Res = [[1,4,3,4],[2,3,1,3],[1,4,3,4],[2,1,2,3]] ? ;  
Res = [[1,4,3,4],[2,3,1,3],[1,4,3,4],[2,1,4,1]] ? ; no
```

Lakótelepi panoráma

Ábrák egy lakótelepet szimbolizál, ahol minden négyzet egy házat jelöl. Az egyes épületek magassága 1-től 4-ig (a nagyobb ábrákban 5-ig) terjed, és mind különböző. Az ábrák köré írt számok azt jelzik, hogy abban a sorban/oszlopban abból az irányból hány ház látható. (Egy magasabb ház eltakarja az alacsonyabbat.)

Minta:

