

Nagyhatékonyságú Deklaratív Programozás
4. gyakorlat

2013. 11. 14.

Globális felhasználói korlátok
=====

%%%%%%%% Minták %%%%%%%%%

```
:- use_module(library(clpfd)).  
:- multifile clpfd:dispatch_global/4.  
:- discontinuous clpfd:dispatch_global/4.
```

%%

```
% leq(X,Y):-  
% X #=< Y globális korlátként megvalósítva  
leq(X,Y):-  
    fd_global(leq(X,Y),void,[min(X),max(Y)]).
```

```
clpfd:dispatch_global(leq(X,Y),State,State,Actions):-  
    dispatch_leq(X,Y,Actions).
```

```
dispatch_leq(X,Y,Actions):-  
    fd_min(X,MinX), fd_max(X,MaxX),  
    fd_min(Y,MinY), fd_max(Y,MaxY),  
    ( number(MaxX), number(MinY), MaxX =< MinY ->  
        Actions = [exit]  
    ; Actions = [X in MinX..MaxY, Y in MinX..MaxY]  
    ).
```

%%

```
% sign(X,S):-  
% S = sign(X) globális korlátként megvalósítva  
sign(X,S):-  
    S in -1..1,  
    fd_global(sign(X,S),void,[minmax(X),minmax(S)]).
```

```
clpfd:dispatch_global(sign(X,S),State,State,Actions):-  
    dispatch_sign(X,S,Actions).
```

```
dispatch_sign(X,S,Actions):-  
    fd_min(X,MinX0), sign_of(MinX0,MinS),  
    fd_max(X,MaxX0), sign_of(MaxX0,MaxS),  
    fd_min(S,MinS0), sign_min_max(MinS0,MinX,_),  
    fd_max(S,MaxS0), sign_min_max(MaxS0,_,MaxX),  
    Actions = [X in MinX..MaxX, S in MinS..MaxS|Exit],  
    ( max(MinS0,MinS) == min(MaxS0,MaxS) -> Exit = [exit]  
    ; Exit = []  
    ).
```

```
% sign_of(X, S): X egész vagy végtelen érték előjele S  
sign_of(inf, S) :- !, S = -1.  
sign_of(sup, S) :- !, S = 1.  
sign_of(X, S) :- S is sign(X).
```

```
% sign_min_max(S, Min, Max): sign(x) = S <=> x in Min..Max  
sign_min_max(-1, inf, -1).  
sign_min_max(0, 0, 0).  
sign_min_max(1, 1, sup).
```

1. Globális korlátként írd meg egy egyneg(L) predikátumot, amely azt fejezi ki, hogy az L lista elemei között pontosan egy negatív szám van. L korlát-változók listája. Figyelj arra, hogy a globális korlát kilépjen, mielőst a tárból következik a feltétel igaz vagy hamis volta (mint az alábbi példák mindegyikében)!

Példák:

```
| ?- egyneg([A,B,C]), C #>= 2, B #> 0.  
    A in inf..-1, B in 1..sup, C in 2..sup ?
```

```
| ?- egyneg([A,B,C]), C #=< -2.  
    A in 0..sup, B = 0..sup, C in inf..-2 ?
```

```
| ?- egyneg([A,B,C]), C #< 0, B = -2.  
    no
```

```
| ?- egyneg([A,B,C]), C #> 0, B = 2, A #> C.  
    no
```

2. Globális korlátként írd meg egy tobblmint2(L) predikátumot, amely azt fejezi ki, hogy az L lista elemei között több az 1-es érték, mint a 2-es. Figyelj arra, hogy a globális korlát kilépjen, mielőst a tárból következik a feltétel igaz vagy hamis volta!

```
| ?- assert(clpfd:full_answer).  
yes  
| ?- X in 0..2, Y in 1..4, Z in 0..4, tobblmint2([X,Y,Z]).  
user:tobblmint2([X,Y,Z]),  
X in 0..2,  
Y in 1..4,  
Z in 0..4 ? ;  
no
```

```
| ?- X in 0..2, Y in 1..4, Z in 0..4, tobblmint2([X,Y,Z]), Y #> 1.  
user:tobblmint2([X,Y,Z]),  
X in 0..1,  
Y in 2..4,  
Z in(0..1)\/(3..4) ? ;  
no
```

```
| ?- X in 0..2, Y in 1..4, Z in 0..4, tobblmint2([X,Y,Z]), Y #> 1, X = 1,  
X = 1,  
user:tobblmint2([1,Y,Z]),  
Y in 2..4,  
Z in(0..1)\/(3..4) ? ;  
no
```

```
| ?- X in 0..2, Y in 1..4, Z in 0..4, tobblmint2([X,Y,Z]), X = 1, Y #> 2.  
X = 1,  
Y in 3..4,  
Z in(0..1)\/(3..4) ? ;  
no
```

```
| ?- X in 0..2, Y in 2..4, Z in 0..4, tobblmint2([X,Y,Z]), X = 1, Y = 2.  
X = 1,  
Y = 2,  
Z = 1 ? ;  
no
```

```
| ?- X in 0..2, Y in 1..4, Z in 0..4, U = 2, tobblmint2([X,Y,Z,U]), Z #\= 1.  
U = 2,  
X = 1,  
Y = 1,  
Z in{0}\/(3..4) ? ;  
no  
| ?-
```