

Emlékeztető - egyes beépített eljárások leírása

sort(+L, ?S)

L, S Az argumentum tetszőleges kifejezések listája.

Igaz, ha az L lista @< szerinti rendezése S, ==/2 szerint azonos elemek ismétlődését kiszűrve.

findall(?Gyűjtő, :+Cél, ?Lista)

bagof(?Gyűjtő, :+Cél, ?Lista)

setof(?Gyűjtő, :+Cél, ?Lista)

Gyűjtő - Az argumentum egy tetszőleges kifejezés.

Cél - Egy meghívható kifejezés (**atom** vagy struktúra).

Lista - Tetszőleges kifejezések listája.

findall/3: A Cél összes megoldására Gyűjtő értéke listába gyűjtve Lista.

bagof/3: Lista az összes olyan Gyűjtő behelyettesítés nem üres listája, amely a Cél egy megoldását adja.

setof/3: ugyanaz mint bagof, de Lista (sort/2-vel) rendezett, ismétlődésmentes lista, azaz: setof(Gy, C, L) :- bagof(Gy, C, L0), sort(L0, L).

Mindegyik eljárás a Cél hívás összes megoldását visszaléptetéssel keresi meg. Ha Cél-ban nincs más behelyettesítetlen változó mint Gyűjtő-ben, akkor a bagof lényegében azonos a **findall** eljárással, csak az a különbség, hogy meghíúsul, ha Cél-nak nincs megoldása. Ha Cél-ban van további behelyettesítetlen változó, amely nincs a ^ operátorral lekötve, akkor ezen további változók minden lehetséges érték kombinációjára külön gyűjti ki a Gyűjtő-értékeket Lista-ba, és ezeket visszalépéskor sorolja fel.

Példa:

varos(becs, osztrak).

varos(budapest, magyar).

varos(graz, osztrak).

varos(pecs, magyar).

varos(pozsony, szlovak).

varos(szeged, magyar).

```
| ?- bagof(Varos, Ország ^ varos(Varos, Ország), L),
/* Y ^ Cél olvasata: létezik olyan Y hogy Cél igaz */
L = [becs,budapest,graz,pecs,pozsony,szeged] ? ; no
```

```
| ?- bagof(Varos, varos(Varos, Ország), L).
```

```
L = [becs,graz], Ország = osztrak ? ;
```

```
L = [pozsony], Ország = szlovak ? ;
```

```
L = [budapest,pecs,szeged], Ország = magyar ? ;
```

```
no
```

+Kif =.. ?Lista (=.. az univ eljárás neve)

-Kif =.. +Lista

Kif - Az argumentum egy tetszőleges kifejezés.

Lista - Az argumentum egy lista, az első eleme egy név vagy egy szám, a többi eleme tetszőleges kifejezés. A lista első eleme csak akkor lehet szám, ha több eleme már nincsen.

Igaz, ha Kif = StrNev(A₁,..., A_n) és Lista = [StrNev,A₁,... A_n].

var(X): X (behelyettesítetlen) változó **nonvar**(X): X változó

compound(X): X struktúra (összetett kifejezés)

atom(X): X névkonstans **integer**(X): X szám

Az alábbi feladatok megoldásában, ha másként nem mondjuk, használhat segédeljárásokat, de ezekhez mindig adjon meg fejkommentet. Megoldásában mindig felhasználhatja az előző feladatokhoz megírt eljárásokat.

Megoldásgyűjtő beépített eljárásokkal kapcsolatos feladatok

=====

Egy könyvtári könyvkatalógust egy olyan Prolog listával ábrázolunk, amelynek elemei egy-egy könyvet írnak le a következő formában:

Könyvcím - [Szerző1, Szerző2, ..., SzerzőN] (N >= 1).

A katalógusban szereplő Könyvcím és Szerző adatok tetszőleges változómentes (azaz tömör) Prolog kifejezések lehetnek. Feltételezheti, hogy a könyvcímek mind különbözőek, és hogy a katalógus egy nem-üres lista.

1. Írjon egy olyan Prolog eljárást, amely egy adott könyvkatalógus és szerző esetén visszaadja a szerző katalógusban szereplő könyveinek listáját (tetszőleges sorrendben)!

Ha a szerzőnek egy könyve sem szerepel a katalógusban, akkor az eljárás hiúsuljon meg! (*)

% *konyvei(+Sz, +KKat, ?KL): Az Sz szerző előfordul a KKat katalógusban és a KKat-ban szereplő könyveinek listája KL.*

```
| ?- konyvei(sz1, [k1-[sz1],k2-[sz2,sz1],k3-[sz3,sz1,sz2]], KL).
KL = [k1,k2,k3] ? ;
```

```
no
```

```
| ?- konyvei(sz2, [k1-[sz1],k2-[sz2,sz1],k3-[sz3,sz1,sz2]], KL).
KL = [k3,k2] ? ;
```

```
no
```

```
| ?- konyvei(sz3, [k1-[sz1],k2-[sz2,sz1],k3-[sz3,sz1,sz2]], KL).
KL = [k3] ? ;
```

```
no
```

```
| ?- konyvei(sz, [k1-[sz1],k2-[sz2,sz1],k3-[sz3,sz1,sz2]], KL).
no
```

- a. Írja meg a konyvei/3 eljárást jobbrekurzív módon. Választhat, hogy a visszaadandó könyvlistát előlről vagy hátulról építi (az utóbbi esetben akkumulálásról van szó).

Az előlről épített lista esetén nevezze az eljárást konyvei_e-nek, a hátulról épített esetben pedig konyvei_h-nak! (Természetesen mindkét változatot megírhatja.)

- b. Írja meg a konyvei/3 eljárást megoldásgyűjtő beépített eljárás (**findall**/3, bagof/3 vagy setof/3) felhasználásával! (Ezeket a változatokat konyvei_f, konyvei_b ill konyvei_s névvel láthatja el.) Figyeljen a meghíúsulás fent említett esetére!

2. Írjon egy olyan Prolog eljárást, amely egy adott könyvkatalógusban szereplő összes szerző listáját adja vissza, ismétlődések nélkül, tetszőleges sorrendben!

% *szerzok(+KKat, ?SzL): A KKat katalógusban szereplő szerzők ismétlődésmentes listája SzL.*

```
| ?- szerzok([k1-[sz1],k2-[sz2,sz1],k3-[sz3,sz1,sz2]], SzL).
```

```
SzL = [sz1,sz2,sz3] ? ;
```

```
no
```

Választása szerint megírhatja az eljárást rekurzióval (amelynek nem feltétlenül kell jobbrekurzióval lennie), illetve a megoldásgyűjtő beépített eljárások segítségével. Ha több változatot ír, azokat az előző feladatban javasolt eljárásnév-kiegészítésekkel különböztetheti meg.

3. Írjon egy az 1. feladatbeli könyvei/3 eljárással azonos jelentésű könyveil/3 eljárást, amely nem követeli meg, hogy az első paraméter adott legyen. A könyveil/3 eljárás tehát egy adott katalógus esetén képes felsorolni az abban előforduló szerzőket (tetszőleges sorrendben), könyveik listájával együtt.

```
% könyveil(?Sz, +KKat, ?KL): Az Sz szerző előfordul a KKat katalógusban
% és a KKat-ban szereplő könyveinek listája KL.
```

```
| ?- könyveil(Sz, [k2-[sz2,sz1],k3-[sz3,sz1,sz2],k1-[sz1]], KL).
KL = [k2,k3,k1], Sz = sz1 ? ;
KL = [k2,k3], Sz = sz2 ? ;
KL = [k3], Sz = sz3 ? ;
no
```

A feladatsor elején bevezettük a könyvkatalógus adatstruktúrát. Analóg módon definiálható a szerzőkatalógus fogalma. Ezt egy olyan Prolog listával ábrázoljuk, amelynek elemei egy-egy szerző (adott könyvtárban előforduló) könyveit írják le a következő formában:

Szerző - [Könyvcím1, Könyvcím2, ..., KönyvcímK] (K >= 1).

4. Írjon olyan Prolog eljárást, amely egy adott könyvkatalógust egy vele ekvivalens szerzőkatalógussá alakítja!

```
% kk_szk(+KKat, ?SzKat): A KKat könyvkatalógus és az SzKat
% szerzőkatalógus ugyanazt a könyvhalmazt írják le.
```

```
| ?- kk_szk([k2-[sz2,sz1],k3-[sz3,sz1,sz2],k1-[sz1]], SzKat).
SzKat = [sz1-[k2,k3,k1],sz2-[k2,k3],sz3-[k3]] ? ; no
```

- a. Írja meg a kk_szk/2 eljárást az előző feladatok megoldására építve!
- b. (Szorgalmi, otthoni feladat) Oldja meg a feladatot egyetlen eljárással (segédeljárás nélkül), a megoldásgyűjtő beépített eljárások segítségével!
- c. (Szorgalmi, otthoni feladat) Oldja meg a feladatot egyetlen eljárással (segédeljárás nélkül), a lists könyvtárban definiált eljárások felhasználásával!
- d. (Szorgalmi, otthoni feladat) Oldja meg a feladatot a lists könyvtár és a megoldásgyűjtő beépített eljárások felhasználása nélkül egyetlen segédeljárással! (A megoldásnak nem kell jobbrekurzívnak lennie.)

Az univ beépített eljárással kapcsolatos feladatok
=====

5. Általános Prolog kifejezés részkifejezéseinek vizsgálata

```
% mern(+K, +N): A K általános Prolog kifejezésben előforduló összes egész
% szám határozottan nagyobb mint N (mern = minden egész részkifejezése
% nagyobb mint)
```

```
| ?- mern(1, 1).
no
| ?- mern(1, 0).
yes
| ?- mern(f(X,[1,3,b],g(2,1,a0)), 0).
true ? ;
no
| ?- mern(f(X,[1,3,b],g(2,1,a0)), 1).
no
```

Megjegyzések:

- a. A "K1 Prolog kifejezésben előfordul a K2 kifejezés" relációt reflexívnek tekintjük, azaz egy K kifejezésben önmaga mindenképpen előfordul. Ez a megjegyzés vonatkozik az ezután következő feladatokra is.

- b. Vigyázzon arra, hogy a kifejezésben változók is előfordulhatnak.

6. Általános Prolog kifejezés bizonyos részkifejezéseinek felsorolása

```
% reszatom(+K, ?A): A a K általános Prolog kifejezésben előforduló atom.
```

```
| ?- reszatom(a, X).
X = a ? ;
no
| ?- reszatom(f(X,[1,3,b],g(2,1,a0)), A).
A = b ? ;
A = [] ? ;
A = a0 ? ;
no
```

Megjegyzés: a struktúranevet nem tekintjük a struktúrakifejezés részének.

7. Általános Prolog kifejezés bizonyos részkifejezéseinek akumulálása

```
% osszege(+K, ?Ossz): Ossz a K kifejezésben előforduló egész számok
% összege.
```

```
| ?- osszege(a, S).
S = 0 ? ;
no
| ?- osszege(1, S).
S = 1 ? ;
no
| ?- osszege(f(X,[1,3,b],g(2,1,a0)), S).
S = 7 ? ;
no
```

8. Általános Prolog kifejezés bizonyos részkifejezéseinek átalakítása

```
% novelt(+Kif0, ?Kif): Kif a Kif0 általános Prolog kifejezésből úgy áll
% elő, hogy minden, benne előforduló egész számot egy eggyel nagyobb
% számra cserélünk.
```

```
| ?- novelt(f(1, X, 2, [3,4.0]), Kif).
Kif = f(2,X,3,[4,4.0]) ? ;
no
| ?- novelt(1, Kif).
Kif = 2 ? ;
no
```

TIPPEK:

1. Érdekes lehet az alábbi segédeljárást definiálni:

```
% könyve(?Sz, +KKat, ?K): Az Sz szerző K könyve szerepel a KKat
% katalógusban.
```

- 1a. Ha előlről építi a listát, akkor első közelítésként érdekes lehet a (*) meghíúsulási feltételtől eltekinteni, majd az így kapott megoldást segédeljárásnak átnevezni, és a (*) feltételt utólag ellenőrizni.