

2010.10.11 Ås 2010.10.13.

-----  
ÅM~Mrjon olyan Prolog eljárÅst, amely megfelel az adott fejkomentnek. A feladat megoldÅsÅhoz felhasznÅlhat korÅbbi sorszÅmÅs feladatokban, illetve elÅlzÅl gyakorlatokon feladott feladatokban szereplÅl eljárÅsokat. Ne hasznÅljon beÅspÅ-tett Ås kÅsnyvtÅri eljárÅsokat, kivÅsve az alÅbbi beÅspÅ-tett eljárÅsokat:

A <rel> B:  
Az A Ås B aritmetikai kifejezÅsek ÅrtÅske kÅszÅst fennÅll a <rel> relÅciÅl.  
<rel> a kÅsvetkezÅik valamelyike: < > = < > = := = \=

X is A:  
X egyesÅ-thetÅl az A aritmetikai kifejezÅs ÅrtÅskÅsvel.

X = Y:  
X Ås Y egyesÅ-thetÅl.

length(L, I):  
Az I egÅsz az L lista hosszÅval azonos. Ha L zÅrt vÅsgÅt lista, vagy I adott egÅsz, akkor az eljárÅs determinisztikus, azaz legfeljebb egy megoldÅst ad.

var(X):  
Az X argumentum (a var/1 meghÅ-vÅsÅnak pillanatÅban) egy behelyettesÅ-tetlen vÅltozÅl.

Ha segÅdeljÅrÅs definiÅlÅsa szÅksÅges, azt kÅzlÅsn jelezzÅk.

A fejkomentekben hasznÅlt Åsn. mÅldjelÅslÅssek magyarÅzata:

\*: az argumentum tÅsmÅr bemenÅl, azaz nem lehet vÅltozÅl, Ås nem is tartalmazhat vÅltozÅlt;

+: az argumentum bemenÅl, azaz nem lehet vÅltozÅl, de tartalmazhat vÅltozÅlt (termÅszetesen csak akkor, ha az argumentum egy struktÅsra);

-: az argumentum kimenÅl, azaz vÅltozÅl;

?: az argumentum lehet kimenÅl Ås bemenÅl is.

### 1. BeszÅrÅs rendezett listÅba

% insert\_ord(\*RL0, +Elem, RL): Az RL monoton nÅsvÅ szÅmlista Ågy Åll elÅl, hogy az RL0 szigorÅsan nÅsvÅ szÅmlistÅba beszÅrjuk az Elem szÅmot, % feltÅsve hogy Elem nem eleme az RL0 listÅnak; egyÅsbkÅsnt RL = RL0.

| ?- insert\_ord([1,3,5,8], 6, L).  
L = [1,3,5,6,8] ? ;  
no  
| ?- insert\_ord([1,3,5,8], 3, L).  
L = [1,3,5,8] ? ;  
no

HasznÅljon feltÅsteles szerkezetet!

### 2. Lista adott sorszÅmÅs eleme

% nthl\_1(+N, ?L, ?E): Az L lista N. eleme E (1-tÅll szÅmozva az elemeket).

Az eljárÅs nevÅsben szereplÅl elsÅl 1-es arra utal, hogy 1-tÅll szÅmozuk az elemeket, a mÅsodik egyes pedig arra, hogy ez az elsÅl vÅltozata egy feladatsornak.

| ?- nthl\_1(3, [a,b,c], E).  
E = c ? ;  
no  
| ?- nthl\_1(3, L, E).  
L = [\_A,\_B,\_E|\_C] ? ;

no

### 3. Adott lista sorszÅmozott eleme

% nthl\_2(?N, +L, ?E): Az L zÅrt vÅsgÅt lista N. eleme E (1-tÅll szÅmozva az elemeket).

| ?- nthl\_2(N, [a,b,c], E).  
E = a,  
N = 1 ? ;  
E = b,  
N = 2 ? ;  
E = c,  
N = 3 ? ;  
%  
no

### 4. Lista adott hosszÅ prefixuma

Egy L n-elemÅt lista prefixumÅnak nevezÅnk egy listÅt, ha az az L elsÅl k elemÅt tartalmazza (az L-beli sorrend megtartÅsÅval), ahol 0 <= k <= n.

% prefix\_length(+Whole, ?Prefix, +Length): A Whole zÅrt vÅsgÅt lista prefixuma a % Prefix lista, amelynek hossza Length.

| ?- prefix\_length([a,b,c,d,e], Prefix, 3).  
Prefix = [a,b,c] ? ;  
no

VizsgÅlja meg, hogy megoldÅsa mÅtkÅdik-e nyÅ-lt vÅsgÅt listÅkra, Ås lÅstrehoz-e vÅsgtelen vÅlasztÅsi pontot!

### 5. Lista adott helyen kezdÅldÅl szuffixuma

Egy L lista szuffixumÅnak nevezÅnk egy listÅt, ha az az L utolsÅl valahÅny elemÅt tartalmazza, az L-beli sorrend megtartÅsÅval.

% suffix\_before(+Whole, ?Suffix, +Before): A Whole zÅrt vÅsgÅt lista azon % szuffixuma a Suffix lista, amelyet megelÅzÅl elemek szÅma Before.

| ?- suffix\_before([a,b,c,d,e], Suffix, 3).  
Suffix = [d,e] ? ;  
no

VizsgÅlja meg, hogy megoldÅsa mÅtkÅdik-e nyÅ-lt vÅsgÅt listÅkra, Ås lÅstrehoz-e vÅsgtelen vÅlasztÅsi pontot!

### 6. RÅszslista kÅszÅs

% sublist(+Whole, ?Part, +Before, +Length): A Whole zÅrt vÅsgÅt lista azon % (folytonos) rÅszslistÅja Part, amely elÅtt Before szÅmÅs elem Åll Ås % amelynek hossza Length.

| ?- sublist([a,b,c,d,e], Part, 1, 3).  
Part = [b,c,d] ? ;  
no

### 7. RÅszslistÅk felsorolÅsa

% sublist(+Whole, ?Part, ?Before, ?Length, ?After): A Whole zÅrt vÅsgÅt lista % olyan (folytonos) rÅszslistÅja Part, amely elÅtt Before Ås amely utÅn After % szÅmÅs elem Åll, Ås amelynek hossza Length.

| ?- sublist([a,b], Part, Before, Length, After).  
Part = [], After = 2, Before = 0, Length = 0 ? ;  
Part = [a], After = 1, Before = 0, Length = 1 ? ;  
Part = [a,b], After = 0, Before = 0, Length = 2 ? ;  
Part = [], After = 1, Before = 1, Length = 0 ? ;  
Part = [b], After = 0, Before = 1, Length = 1 ? ;  
Part = [], After = 0, Before = 2, Length = 0 ? ;  
no

=====  
Szorgalmi/otthoni feladatok  
=====

8. Lista sorszámozott eleme: tásbb irányban mátkásdál eljárás

Gondolja meg, hogy mennyire hatáskony a 3. feladatra adott megoldása abban az esetben ha N egy adott szám.

A var/1 beáspá-tett eljárás a feltételes szerkezet használatával káosz-tse el a (2, és 3. feladatbeli) nthl\_1 és nthl\_2 eljárások egy olyan nthl\_3 kombinációját, amely minden input-output párdban helyesen és hatáskonyan mátkásdik (ha az első kást argumentum behelyettesítetlen, akkor -- jogosan -- vágtelen sok megoldást adjon vissza).

9. Beszárás listába adott helyre

% insert\_nth(+N, ?L0, ?E, ?L): Az L lista N. eleme E, ezt megelízál  
% elemei azonosak L0 1., 2., ... N-1. elemével, ezt kásvetál elemei  
% azonosak L0 N., N+1., ... elemeivel.

```
| ?- insert_nth(3, [a,b,c,d,e], i, L).  
L = [a,b,i,c,d,e] ? ;  
no  
| ?- insert_nth(3, [a,b,c,d,e], X, L).  
L = [a,b,X,c,d,e] ? ;  
no  
| ?- insert_nth(3, [1], 2, L).  
no  
| ?- insert_nth(3, L0, X, L). % (*)  
L = [_A,_B,X|_C],  
L0 = [_A,_B|_C] ? ;  
no
```

Elm-^Qszár oldja meg a feladatot feltételes szerkezet használatával nášlkázl!  
Adjon olyan megoldást is amely feltételes szerkezetet használ (a megoldásban ez insert\_nth\_2 néven szerepel).

10. Beszárás adott listába

% insert\_nth\_3(?N, +L0, ?E, ?L): Az L zárt végű lista N. eleme E, ezt  
% megelízál elemei azonosak L0 1., 2., ... N-1. elemével, ezt kásvetál  
% elemei azonosak L0 N., N+1., ... elemeivel.

Az insert\_nth\_3 eljárás ugyanazt az eredményt adja mint az insert\_nth az utábbinál felsorolt példák esetében, kivéve a (\*)-gal jelölt, és mátkásdik az alábbi példákra is:

```
| ?- insert_nth_3(N, [1], 2, L).  
L = [2,1],  
N = 1 ? ;  
L = [1,2],  
N = 2 ? ;  
no  
| ?- insert_nth_3(N, [1], X, L).  
L = [X,1],  
N = 1 ? ;  
L = [1,X],  
N = 2 ? ;  
no
```

Ezt a feladatot kihagyhatja ha helyette megoldja az alábbi 11. feladatot, amely ennek egy általános-tása.

11. Beszárás adott listába vagy adott lista-eredménnyel

Az insert\_nth\_4 eljárás jelentése (és fejkommentje) ugyanaz mint az 9. és 10. feladatbeli eljárás, de mátkásdnie kell akkor is, ha zárt végű az a lista, amelybe beszárunk, és akkor is ha a beszárás eredménye adott.

A 9. és 10. feladatokban felsorolt összes példa esetén a megadott eredményt adja, kivéve a (\*)-gal jelöltet, és mátkásdik az alábbi példákra is:

```
| ?- insert_nth_4(N, L0, X, [a,b]).  
N = 1,  
X = a,  
L0 = [b] ? ;  
N = 2,  
X = b,  
L0 = [a] ? ;  
no
```

Vizsgálja meg, hogy mennyire hatáskony ez a megoldás abban az esetben ha N egy megadott szám.

A var/1 beáspá-tett eljárás a feltételes szerkezet használatával káosz-tsen egy olyan insert\_nth/4 változatot, amely hatáskonyan mátkásdik ha az 1., 2. és 4. argumentumok kászázl legalább az egyik adott (szám ill. zárt végű lista). A megoldásban ez a változat insert\_nth\_5 néven szerepel.

=====

Tippek:

3. feladat

A member/2 eljárás kást argumentumát cserélje meg, és egáosz-tse ki egy első N argumentummal, amely a beszárás helyét számlálja. Mivel N lehet behelyettesítetlen változó is, ezért tekintse kimenő argumentumnak. (Nem baj, ha az eljárás nem jobbrekurzív.)

10. feladat

A select/3 eljárás argumentumainak átrendezésével káosz-tsen egy insert/3 eljárást:

% insert(L0, E, L): L az L0 listából az E elem beszárásával ján lástre.

Ezt az eljárást egáosz-tse ki egy N első argumentummal, és kásvesse a 3. feladathoz adott tippet.