Recursivitate in PROLOG

In PROLOG, exista 2 tipuri de recursivitate pe care le introducem aici astfel:

- Orice predicat care se autoapeleaza este unul *recursiv*.
- Un predicat *recursiv* ale carui alternative (implicite/explicite) se termina cu autoapel il numim *recursiv la urma* (tail recursive)

Asa cum se va vedea in exemplele urmatoare, predicatele recursive la urma sunt mai performante (mai eficiente d.pd.v. computational) decat celelalte.

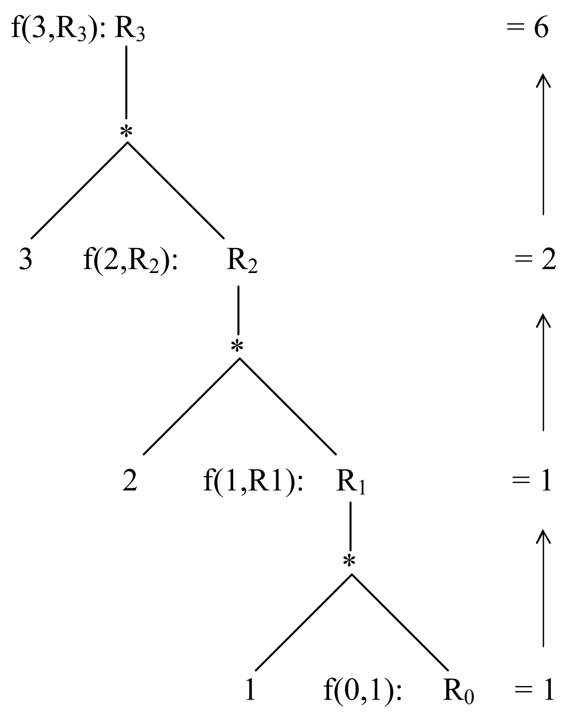
Sa pornim de la o implementare simplu-recursiva a factorialului:

```
f(integer, real)
f.(0,1).
f(N,R) :- N>0, Ni=N-1, f(Ni, Ri), R=Ri*N.
```

Scrieti arborele de cautare corespunzator executiei acestui predicat! (a se vedea cursul si seminarul).

Vom insista mai jos pe o subparte a arborelui de mai sus corespunzatoare ultimelor doua instructiuni ale clauzei care implementeaza factorialul in exemplul de mai sus, subparte care evidentiaza dependenta rezultatului final de rezultatele intermediare obtinute recursiv. Considerma urmatorul exemplu de interogare: f(3,X).

Apelul $f(3,R_3)$ defineste R3 ca rezultat al inmultirii dintre 3 si un rezultat R_2 apriori necunoscut al apelului $f(2,R_2)$. Mai departe, R_2 este si el definit recursiv, si mecanismul se propaga identic pana la apelul $f(0,R_1)$ care unifica R_1 la 1.
Urmeaza apoi evaluarea nodurilor arborelui pana la gasirea lui R_3.
Calculul rezultatului R_3 corespunzator celui mai exterior apel implica deci o dubla parcurgere (o extindere si o evaluare) a unui arbore definit de relatia de legatura intre rezultatele intermediare la apel si autoapel: R=Ri*N.



Eliminarea dublei parcurgeri ar fi posibila daca rezultatul cautat ar fi disponibil ca atare in cel mai interior autoapel si daca la orice nivel, propagarea autoapelurilor s-ar face

recursiv inainte (prin trimiterea informatiei utile in urmatorul autoapel si nu prin referirea la un rezultat apriori necunoscut), ceea ce revine la lipsa oricarei instructiuni care sa succeada autoapelului.
In general, trecerea la recursivitate la urma a unui predicat se face prin redefinirea lui cu o aritate suficient de mare care sa asigure suficiente variabile pentru transmiterea informatiilor utile la autoapel. Tehnica de scriere a predicatelor recursive la urma este exemplificata de urmatorul predicat:

Factorialul lui N este R **daca** R poate fi calculat recursiv la urma iterand intre 0 si N, pornind de la un rezultat intermediar initializat cu 1:

```
f2(N,R) :- f2(N,0,1,R). % (Cf, Ci, Ri, R)
```

unde: Cf-contor final, Ci-contor initial, Ri-rezultat intermediar initializat, R-rezultat final). Odata cu redefinirea trebuie declarat si noul antet f2(int,int,real,real).
Noul predicat de aritate 4 este asumat recursiv la urma, deci se termina in autoapel:

```
f2(N,Ci,Ri,R) :- ………………., f2(N,Cia,Ria,R).
```

unde autoapelul are primii trei parametri legati la valorile actualizate.
Restul instructiunilor vor fi teste care conditioneaza propagarea autoapelului,

```
f2(N,Ci,Ri,R) :- Ci<N,………………., f2(N,Cia,Ria,R).
```

respectiv instructiuni de legatura intre valorile parametrilor de apel si valorile actualizate trimise in autoapel:

```
f2(N,Ci,Ri,R) :- Ci<N, Cia=Ci+1, Ria=Ri*Cia, f2(N,Cia,Ria,R).
```

Adaugand faptul de oprire, predicatul f2 devine:

```
f2(N,N,Ri,Ri). % (i,i,i,o)
f2(N,Ci,Ri,R) :- Ci<N, Cia=Ci+1, Ria=Ri*Cia, f2(N,Cia,Ria,R).
```

Faptul de oprire impune ca la atingerea contorului final N, rezultatul intermediar calculat (acumulat in pozitia 3 in lista de argumente) sa iasa pe pozitia de rezultat final.

Exercitii:

1. Folosind „trace"-ul, analizati executia celor doua predicate factorial definite mai sus. Ce constatati?
2. Sa se scrie un predicat de aritate 3, recursiv la urma, care sa calculeze factorialul.
3. Sa se scrie un PRU (predicat recursiv la urma) pentru calculul numarului lui Euler ca limita a sirului $(1 + 1/n)^n$.
4. Sa se scrie un PRU pentru calculul numarului lui Euler ca suma a seriei $\sum(1/n!)$, cu n in N.
5. Care este valoarea maxima a lui n pentru care suma de la punctul (4) poate fi calculata?
6. Definiti un PRU de aritate 1 pentru calculul numarului lui Euler prin suma $\sum(1/n!)$, cu n intre 0 si valoarea determinata la punctul 5.

7. Reformulati predicatul de la punctul 6 astfel incat sumarea sa se faca incepand cu indicele cel mai mare si terminand cu zero.
8. Comparati aproximatiile numarului lui Euler obtinute la punctele 6 si 7. Comentati.
9. Dupa cate autoapeluri predicatul de la punctul 6 va produce o aproximare cu 14 zecimale exacte a numarului lui Euler? Dar cu 8? Dar cu 5?
10. Aceeasi intrebare ca mai sus pentru predicatul de la punctul 3.