

# Pràctiques de Prolog per a Llenguatges de Programació (04-05)

Trieu una de les 4 parelles d'exercicis següents. Aquestes pràctiques són per parelles i puntuen fins a 8 si estan ben fetes. Per obtenir 2 punts més podeu presentar-les dins d'una aplicació Visual Basic o bé fer una pràctica de resolució de constraints (demaneu-ho al professor). La data de lliurament per a primera convocatòria és el dia abans de l'examen i per a segona, el dia abans de l'examen de segona convocatòria. Haureu d'enviar el codi per mail i lliurar un document (en paper) amb el codi convenientment comentat.

0. (a) Escriviu un predicat `fact(N,F)` que significa:  $F$  és el factorial del número natural  $N$ , per als casos:
  - quan ens donen instanciat un número natural  $N$  i volem generar-ne el factorial  $F$ . Feu-ho de manera que funcioni correctament si es demanen més respostes, és a dir que falli.
  - quan ambdues variables no estan instaciades i vulguem generar tots els parells  $N, F$ .
  - quan el que ens ve instanciat és el factorial i vulguem calcular-ne la  $N$ .
  - quan tant  $N$  com  $F$  vinguin instanciats i vulguem respostes si/no.

Finalment, feu-ne un que ens permeti tractar tots els casos d'abans. Utilitzeu `var(X)` i `nonvar(X)`.

- (b) Definir el predicat `take(L1, L2, N)` tal que donats una llista  $L1$  i un número  $N$ ,  $L2$  és la llista que conté els elements de  $L1$  agrupats de  $N$  en  $N$ . És a dir,  $L2$  és la llista de llistes de  $N$  elements cadascuna (menys potser la última) tal que al concatenar-les totes, obtenim la llista  $L1$ . Si donem com a dades  $N$  i  $L2$ , `take(L1, L2, N)` ha de

tornar-nos L1. Semblantment,  $\text{take}(L1, L2, N)$  ha de funcionar li les tres variables L1, L2 i N estan instanciades.

Si L1 és [1, 2, 3, 4, 5] i N és 2, llavors  $\text{take}(L1, L2, N)$  ha de respondre  $L2 = [[1,2], [3,4], [5]]$ .

Si N és 2 i L2 és  $[[1,2], [3, 4], [5]]$  llavors  $\text{take}(L1, L2, N)$  ha de respondre  $L1 = [1, 2, 3, 4, 5]$ .

1. (a) Escriviu un predicat  $\text{mcm}(X, Y, M)$  que significa: M és el mínim comú múltiple de X i de Y. Ha de ser capaç de contestar correctament els següents casos:
  - quan ens venen instanciats els tres paràmetres, ha de dir si M és o no el mcm de X i de Y.
  - quan només ens donen instanciats X i Y, ha de calcular-nos la M correctament.
  - Escriviu un predicat  $\text{prod}(L, P)$  que significa: P és el producte dels elements de la llista d'enters L. Ha de poder tant generar P com comprovar una P donada.
- (b) En el joc de "xifres", el jugador disposa d'una llista L de números enters i un objectiu N, que és un altre nombre enter. El jugador ha de trobar una manera d'obtenir N a base de sumar, restar i multiplicar alguns nombres de la llista L. Es pot usar cada nombre tantes vegades com aparegui a la llista L.  
Per exemple, a  $L = [4, 9, 8, 7, 100, 4]$  i  $N = 380$ , el programa hauria de respondre:  $4 * (100 - 7) + 8, ((100 - 9) + 4) * 4, \dots$
2. (a) Definiu el predicat:  $\text{divisors}(N,L)$  tal que donat un natural N, L serà la llista de divisors de N en ordre creixent. Així, si N és 10, L serà: [1,2,5,10]. Ha de respondre YES si una L donada és la llista creixent dels divisors d'un N donat i NO en cas contrari. També ha de poder generar la llista quan li donen ja instanciada la N.
- (b) Resoldre el problema de la coloració de mapes mitjançant el predicat  $\text{coloreja}(M,L,PC)$ , on M serà la llista l'adjacencies entre països, L la llista de possibles colors i PC la llista de països amb color assignat, de manera que la query:

coloreja( [adjacent(portugal, espanya), adjacent(espanya, franca),  
adjacent(espanya, andorra), adjacent(andorra, franca),  
adjacent(franca, espanya)],  
[blau, vermell, verd, groc],  
[color(espanya, E), color(franca, F), color(andorra, V),  
color(portugal, P)] ).

es satisfarà per exemple així:

E = groc,

F = verd,

V = vermell,

P = verd?

3. (a) Definiu el predicat  $\text{daus}(P, N, L)$  on la llista  $L$  expressa una manera de sumar  $P$  punts llençant  $N$  daus. Així si  $P$  és 5 i  $N$  és 2, una solució de  $L$  seria  $[1,4]$ ; la resta de solucions serien  $[3,2]$ ,  $[4,1]$  i  $[2,3]$ . Fixeu-vos que la longitud de les llistes és 2.  $P$  i  $N$  han d'estar inicialment instanciades.
- (b) Tenim una aixeta d'aigua, un cubell de 5 litres i un altre de 8. Es pot vocar el contingut d'un cubell a un altre, omplir el cubell o buidar un cubell del tot. Escriure un programa Prolog que digui amb quina seqüència d'operacions podem obtenir exactament 4 litres d'aigua en el cubell de 8 litres. Resoleu el problema de manera general, és a dir, que es pugui resoldre qualsevol problema de l'estil "amb un primer cubell de  $X$  litres i un segon de  $Y$  litres, obtenir  $Z$  litres exactes en el segon".