

PROGRAMACIÓN DECLARATIVA
INGENIERÍA EN INFORMÁTICA

18 de enero de 2012

Apellidos: Nombre:

Ejercicio 1

Consideremos el siguiente programa lógico:

```
p( _, _, 0, 0, [] ). % R1
p(X, Y, s(M), N, [X|L]) :- p(X, Y, M, N, L). % R2
p(X, Y, M, s(N), [Y|L]) :- p(X, Y, M, N, L). % R3
```

1. Construye el árbol de resolución para el programa anterior y la consulta
 $?- p(a, b, s(s(0)), s(0), L).$
2. Utiliza el árbol de resolución construido en el apartado anterior para razonar qué respondería Prolog a la consulta en cada uno de los siguientes supuestos:
 - La regla R2 se sustituye por la regla
 $p(X, Y, s(M), N, [X|L]) :- p(X, Y, M, N, L), !.$
 - La regla R3 se sustituye por la regla
 $p(X, Y, M, s(N), [Y|L]) :- p(X, Y, M, N, L), !.$

Ejercicio 2

La *secuencia de Collatz* se define por recursión como sigue:

$$a_{n+1} = \begin{cases} \frac{a_n}{2}, & \text{si } a_n \text{ es par} \\ 3a_n + 1, & \text{si } a_n \text{ es impar} \end{cases}$$

Se cree, aunque no está demostrado (pero si comprobado hasta números realmente grandes), que sea cual sea el término inicial que se elija, la secuencia siempre termina llegando al número 1.

1. Define el predicado `secuencia_Collatz(+X, -L)` que se verifica si L es la lista de los elementos de la secuencia de Collatz que comienza en X. Por ejemplo:

?- `secuencia_Collatz(1, L).`

L = [1].

?- `secuencia_Collatz(2, L).`

L = [2, 1].

?- `secuencia_Collatz(3, L).`

L = [3, 10, 5, 16, 8, 4, 2, 1].

2. Usando los predicados predefinidos **findall**/3 y **between**/3 define el predicado `longitudes_secuencias(+N, -L)` que se verifica si L es la lista de las longitudes de las secuencias de Collatz cuyo término inicial es un número entre 1 y N. Por ejemplo:

?- `longitudes_secuencias(3, L).`

L = [1, 2, 8].