

**PROGRAMACIÓN LÓGICA Y FUNCIONAL**  
**EXAMEN PARCIAL: PROGRAMACIÓN LÓGICA**  
**GIJÓN 24 DE ENERO DE 2001**

1. Considérese el siguiente programa definido

- |  |              |
|--|--------------|
| ① $p(g(X), Y) \leftarrow r(X), s(f(Y)).$ | ⑤ $s(f(a)).$ |
| ② $p(X, X) \leftarrow r(f(X)).$          | ⑥ $s(f(b)).$ |
| ③ $r(X) \leftarrow p(f(X), g(X)).$       | ⑦ $s(c).$    |
| ④ $r(b).$                                |              |

Se pide:

- i) Construir el SLD-árbol asociado al objetivo  $\leftarrow p(X, Y)$ , lo suficiente para identificar las soluciones obtenidas y las ramas infinitas (si las hubiera).
- ii) Para la primera refutación alcanzada, obténgase la composición de sustituciones realizadas en ella así como la “sustitución respuesta computada” asociada.
- iii) Introducir un único corte y/o un cambio de orden de dos cláusulas para que no se obtengan derivaciones fallidas ni ramas infinitas, y se obtenga el mismo **conjunto de soluciones**.

2. Definir el predicado  $f/n$  que calcule la siguiente función utilizando **necesariamente recursión de cola**.

$$f(X, Y) = \begin{cases} X + f(X-1, Y) & \text{si } X > Y \\ X + Y & \text{si } X = Y \\ Y + f(X, Y-X) & \text{si } X < Y \end{cases}$$

3. Definir en Prolog **todos** los siguientes predicados, se valorará la sencillez y el aprovechamiento de los recursos de Prolog. Si se emplea algún predicado auxiliar también tendrá que ser definido, no se pueden usar predicados predefinidos de Prolog.

- i)  $solo1(Web, ListaVisitas)$  recibe una dirección Web y una lista de direcciones y devuelve cierto si *Web* sólo aparece una vez en la lista.

La lista viene representada en Prolog por una lista de átomos, ej:

```
?- solo1('www.uniovi.es', ['www.yahoo.com', 'www.microsoft.com', 'www.internautas.org', 'www.uniovi.es']).
```

yes

```
?- solo1('www.uniovi.es', ['www.yahoo.com', 'www.microsoft.com', 'www.internautas.org', 'www.uniovi.es', 'www.uniovi.es']).
```

no

- ii)  $muy\_interesantes(ListaVisitas, ListaInteresantes)$  recibe una lista de visitas y devuelve la lista de direcciones Webs cuyas visitas duraron más de 15 minutos.

En esta ocasión cada visita, elemento de la lista, es una estructura de la forma:

$w(DirWeb, TiempoConexión)$

donde *DirWeb* es el átomo con la dirección y *Tiempo* es el tiempo en segundos que estuvo conectado a esa página.

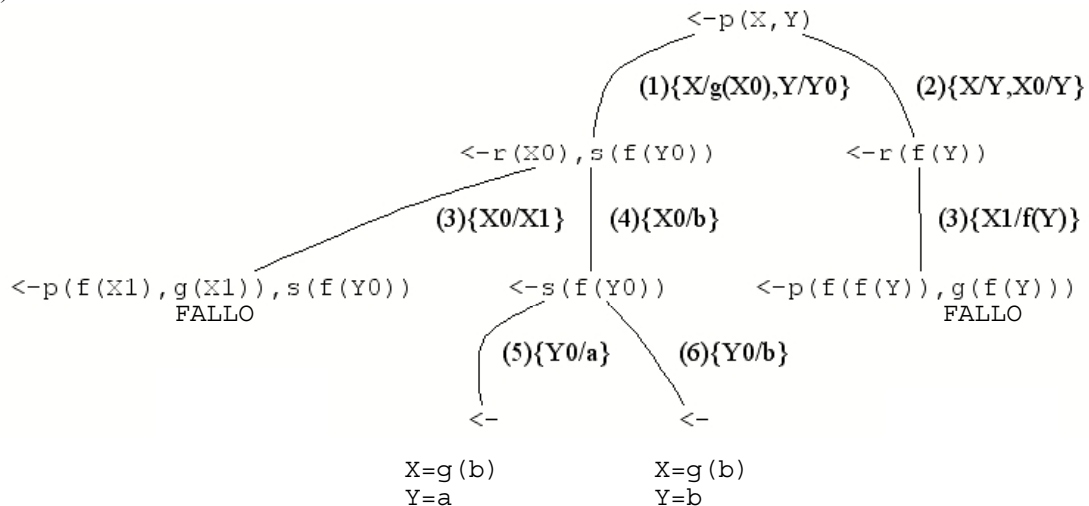
Ejemplo:

```
?- muy_interesantes([w('www.yahoo.com', 8658), w('www.microsoft.com', 220), w('www.uniovi.es', 6532)], L).
```

L = ['www.yahoo.com', 'www.uniovi.es']

**Puntuación: 1 (3 pts), 2 (2 pts), 3 (5 pts)**

1.i) SLD-ARBOL



1.ii) S.C. = {X/g(b), Y/a, X0/b, Y0/a} S.R.C. = {X/g(b), Y/a}

1.iii) Intercambiamos la ③ y la ④ e insertamos un corte en la cláusula ① entre r/1 y s/1 para así obtener las dos refutaciones y no obtener la rama fallida al resolver por ③.

① p(g(X), Y) ← r(X), 1, s(f(Y)).      ⑤ s(f(a)).

② p(X, X) ← r(f(X)).      ⑥ s(f(b)).

③ r(b).      ⑦ s(c).

④ r(X) :- p(f(X), g(X)).

2) Definir f(X,Y,Z).

f(X,X,Acu,Z) :- Z is Acu+X+X.

f(X,Y,Acu,Z) :- X>Y, Acu2 is Acu+X, X2 is X-1, f(X2,Y,Acu2,Z).

f(X,Y,Acu,Z) :- Y>X, Acu2 is Acu+Y, Y2 is Y-X, f(X,Y2,Acu2,Z).

3.a) solo1(Web, ListaVisitas)

solo1(Web, [Web|Resto]) :- ninguna(Web, Resto).

solo1(Web, [Web2|Resto]) :- Web \== Web2, solo1(Web, Resto).

ninguna(Web, []).

ninguna(Web, [Web2|Resto]) :- Web \== Web2, ninguna(Web, Resto).

3.b) muy\_interesantes(ListaWebs, ListaInteresantes)

muy\_interesantes([], []).

muy\_interesantes([w(Dir, Tiempo)|Cola], [Dir|Z]) :-

Tiempo>900, !, muy\_interesantes(Cola, Z).

muy\_interesantes([\_|Cola], Z) :- muy\_interesantes(Cola, Z).