

# Ejercicios Prácticos de Programación Declarativa

## Sesión de laboratorio 7

Curso 2019/20

- Sube un fichero .pl al Campus Virtual antes de que acabe la clase con las soluciones de los ejercicios de esta sesión (en el caso del ejercicio 1 debes incluir tus respuestas como comentarios). Es suficiente con que lo suba uno si lo hacéis entre dos.
- No olvidéis incluir comentarios y poner vuestros nombres en las primeras líneas del fichero.

1. Considera la siguiente especificación:

$\text{elimina}(L, X, NL) \longleftrightarrow NL$  es la lista resultante de eliminar de la lista  $L$  todas las apariciones de  $X$ .

Estudia las diferencias entre las siguientes definiciones de este predicado escritas en Prolog.

- (a) Usando igualdad sintáctica:

```
elimina1([ ],X,[ ]).
elimina1([X|R],Y,NR) :- Y == X, elimina1(R,Y, NR).
elimina1([X|R],Y,[X|NR]) :- Y \== X, elimina1(R,Y,NR).
```

- (b) Usando unificación:

```
elimina2([ ],X,[ ]).
elimina2([X|R],Y,NR) :- Y = X, elimina2(R,Y, NR).
elimina2([X|R],Y,[X|NR]) :- Y \= X, elimina2(R,Y,NR).
```

- (c) Combinando las dos anteriores:

```
elimina3([ ],X,[ ]).
elimina3([X|R],X,NR) :- elimina3(R,X,NR).
elimina3([X|R],Y,[X|NR]) :- Y \== X, elimina3(R,Y,NR).
```

Ejecuta los siguientes objetivos en cada una de las tres versiones.

```
?- elimina1([a,b,a,c],a,L). (i= 1,2,3)
```

```
?- elimina1([a,b,a,c],X,L). (i= 1,2,3)
```

Compara los resultados. ¿Qué puedes concluir?

2. Utilizando la estructura Prolog de árbol binario definida en clase, programa los siguientes predicados:

- (a)  $\text{simetrico}(A_1, A_2) \longleftrightarrow A_1$  y  $A_2$  son árboles binarios simétricos.

- (b)  $\text{sumatree}(A, N) \longleftrightarrow A$  es un árbol binario con números enteros en sus nodos y  $N$  es la suma de sus elementos.

- (c)  $\text{maxveces}(A, X) \longleftrightarrow A$  es un árbol binario y  $X$  es el elemento que aparece repetido más veces en sus nodos.

3. Programa los predicados *subtermino* y *subst* especificados a continuación. Haz uso de los predicados metalógicos de Prolog cuando te sea útil.

$\text{subtermino}(S, T) \longleftrightarrow S$  es un subtérmino de  $T$ .

Por ejemplo, el objetivo `subtermino(S,f(a,g(b)))` tiene que dar las siguientes soluciones:

`S=f(a,g(b)); S=a; S=g(b); S=b.`

$\text{subst}(T, S, E, NE) \longleftrightarrow NE$  es el resultado de reemplazar en la estructura  $E$  cada aparición del término  $T$  por  $S$ .

Por ejemplo, el objetivo `subst(a, f(b), h(a,f(a)), E)` tiene que dar la siguiente solución:

`E=h(f(b),f(f(b))).`

Analiza en qué modos de uso funcionan bien tus implementaciones.

4. Programa en Prolog una versión recursiva para resolver el problema de las torres de Hanoi. Para ello, define un predicado cuyos argumentos sean el número de fichas, los nombres de las torres inicial, final y auxiliar y la secuencia de movimientos de una torre a otra hasta conseguir la traslación de todas las fichas.