

PRÁCTICA 7: INTRODUCCIÓN A PROLOG. EJERCICIOS.

Ejercicio 1. Escribir una base de datos en PROLOG que contenga lo siguiente:

```
es_mujer(belen).
es_mujer(maria).
es_mujer(julia).
juega_a_poker(julia).
juega_a_poker(maria).
juega_a_poker(pepe).
asiste_a(belen,lmd).
asiste_a(pepe,lmd).
asiste_a(julia,alem).
```

Plantear distintos objetivos a PROLOG que devuelvan true o false.

Ejercicio 2. Usando la base de datos del ejercicio anterior, tratar de predecir la salida de PROLOG en modo `trace` para el objetivo

```
(es_mujer(X);juega_a_poker(X)),asiste_a(X,lmd).
```

Comparar la predicción con la salida real de PROLOG en modo `trace` para dicho objetivo.

Ejercicio 3. Cargar la base de datos `pizza.pl` que acompaña a la práctica 7 e interpretar su contenido. Utilizar PROLOG para encontrar:

- (1) todas las pizzerías frecuentadas por al menos un hombre;
- (2) todas las mujeres que comen pizzas funghi y pepperoni;
- (3) todas las mujeres que comen pizzas funghi o pepperoni.

Ejercicio 4. Usando la base de datos `pizza.pl`, encontrar lo siguiente:

- (1) todas las pizzerías que sirven una pizza, por menos de 8 euros, comida por Diana.
- (2) todas las pizzerías frecuentadas por al menos una persona menor de edad.

Ejercicio 5. Crear una base de datos `familia.pl` que contenga lo siguiente:

```
es_madre_de(ana, belen).
es_madre_de(ana, maria).
es_madre_de(julia, rocio).
es_madre_de(rocio, nuria).
es_madre_de(rocio, jorge).
es_madre_de(belen, juan).
es_madre_de(belen, eva).

es_padre_de(paco, belen).
es_padre_de(paco, maria).
es_padre_de(antonio, rocio).
es_padre_de(jose, nuria).
es_padre_de(jose, jorge).
es_padre_de(antonio, juan).
```

```
son_hermanos(H1, H2):-
  es_madre_de(M,H1),
  es_madre_de(M,H2),
  es_padre_de(P,H1),
  es_padre_de(P,H2),
  H1 \== H2.
```

```
es_progenitor_de(P, H):- es_madre_de(P,H).
es_progenitor_de(P, H):- es_padre_de(P,H).
```

```
es_abuela_de(A, N):-
  es_madre_de(A,X),
  es_progenitor_de(X,N).
```

- (1) Interpretar las reglas `son_hermanos`, `es_progenitor` y `es_abuela`.
- (2) ¿Qué objetivo hay que plantear a PROLOG para obtener como respuesta los nietos de `ana`?
- (3) Crear una regla `es_tio_de(T,S)` que devuelva `true` si T es el/la tío/tía de S.
- (4) Crear una regla `son_primos(P1,P2)` que devuelva `true` si P1 e P2 son primos.

Ejercicio 6. Ampliar la base de datos `pizza.pl` incluyendo:

- (1) Una regla de una variable `Pizzeria` que devuelva `true` si `Pizzeria` sirve pizza pepperoni.
- (2) Una regla de una variable `Nombre` que devuelva `true` si `Nombre` es una mujer que come pizzas funghi y pepperoni.
- (3) Una regla de una variable `Nombre` que devuelva `true` si `Nombre` es una mujer que come pizzas funghi o pepperoni.
- (4) Una regla de una variable `Nombre` que devuelva `true` si `Nombre` es una persona mayor de edad.
- (5) Una regla de una variable `Pizza` que devuelva `true` si `Pizza` es una pizza con coste inferior a 8 euros.
- (6) Una regla de dos variables `Persona1` y `Persona2` que devuelva `true` si `Persona1` y `Persona2` comen la misma pizza.
- (7) Una regla de dos variables `Persona1` y `Persona2` que devuelva `true` si `Persona1` y `Persona2` pueden encontrarse en la misma pizzería.
- (8) Una regla de tres variables `Pizza`, `Pizzeria1` y `Pizzeria2` que devuelva `true` si `Pizza` es más barata en `Pizzeria1` que en `Pizzeria2`.

Ejercicio 7. Definir, de forma recursiva, una función `factorial(N,F)` que devuelva `true` si F es el factorial de N. Usar la función para obtener el factorial de 17.

Ejercicio 8. Anticipar cuáles de las siguientes expresiones unifica PROLOG:

- (1) `X = es_madre_de(b,X).`
- (2) `a(c,r(X)) = a(d,Y).`
- (3) `a(X,Y) = a(b(c,d),r(X)).`
- (4) `arbol(izquierda,raiz, Derecha) = arbol(izquierda,raiz, arbol(a,b,c)).`

Ejercicio 9. Definir la regla `estaciones(Estacion,Dia,Mes)` que devuelva `true` si el día `Dia` del mes `Mes` forma parte de la estación `Estacion`.

Ejercicio 10. Definir la regla `es_elemento(X,Lista)` que devuelva `true` si `X` es un elemento de la lista `Lista`.

Ejercicio 11. Definir la regla `suma(Lista,S)` que devuelva `true` si `S` es la suma de los elementos de la lista `Lista`.

Ejercicio 12. Definir la regla `ultimo(X,Lista)` que devuelva `true` si `X` es el último elemento de la lista `Lista`.

Ejercicio 13. Definir la regla `elimina(X,M,R)` que devuelva `true` si `R` es la lista que resulta de eliminar en la lista `M` el elemento `X`.

Ejercicio 14. Definir la regla `creciente(L)` que devuelva `true` si los elementos (supuestos números todos ellos) de la lista `L` siguen un orden natural creciente.