

# Laboratorio de Fundamentos de Inteligencia Artificial

## TEMA 4

### Ejercicio Nº 1.-

Vamos a realizar un primer ejercicio que no conlleva recursividad, pero que prepara para el siguiente.

Describir la disposición de los objetos de la figura usando hechos como **derecha\_de**(llave, tortuga) y **encima\_de**(coche, tortuga) (*derecha\_de* significa *inmediatamente a la derecha de* y *encima\_de* significa *inmediatamente encima de*).

Definir los predicados **izquierda\_de** y **debajo\_de** en términos de los anteriores (*derecha\_de* y *encima\_de*). No hay que volver a repetir todos los hechos anteriores al revés, sino generalizar mediante reglas.



### Ejercicio Nº 2.-

Completamos el ejercicio 1 con las sentencias que hagan falta para programar recursividad.

**Generalizar** los predicados **derecha\_de** y **encima\_de**, presentados en el ejercicio anterior, con los predicados **derecha** y **encima**, de manera que una pregunta como **derecha(X,tortuga)** proporcione sucesivamente los distintos objetos que están a la derecha de la tortuga (llave, semáforo, martillo). El orden en el que salgan no es importante.

### Ejercicio Nº 3.-

**Escribir un programa** en PROLOG, que ante la siguiente pregunta: **?- factorial(N, X)**, responda con el valor de X igual al factorial del número N concreto. (Recordar que factorial de 0 es 1).

#### Ejercicio N° 4.-

En un sistema Prolog se almacena información sobre diferentes enlaces entre ciudades con el formato:

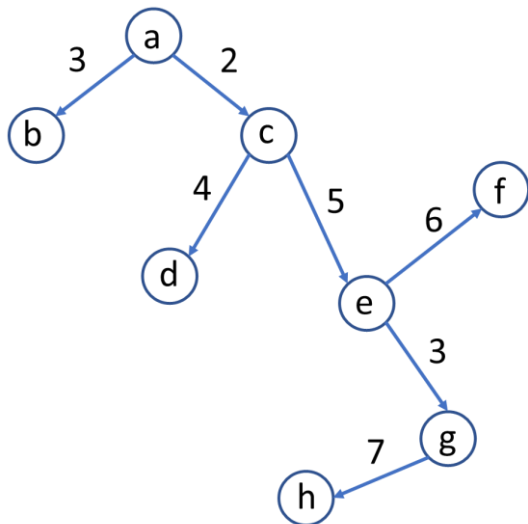
`enlace(Ciudad1, Ciudad2, Distancia)`

**Se pide:** Escribir la definición de un predicado

**`ruta(Ciudad1, Ciudad2, DistanciaTotal)`**

que se cumple si existe una ruta entre Ciudad1 y la Ciudad2 de distancia DistanciaTotal

**Ejemplo:** La situación de la figura se podría representar con los hechos:



`enlace(a,b,3).`

`enlace(a,c,2).`

`enlace(c,d,4).`

`enlace(c,e,5).`

... (habría que terminar de completar la representación de la figura)

Ante la pregunta `?- ruta(a,f,D)` el sistema debería responder `D=13`

#### Ejercicio N° 5.-

Calcular la potencia de un número dado sin utilizar aritmética como  $5^3$  ni predicados predefinidos, sino aplicando recursividad. Recordar que un número elevado a 0 es 1. Por ejemplo:

`?- potencia(5,3,X).`

`X=125;`

`false.`

#### Ejercicio N° 6.-

Obtener el producto de dos números naturales dados mayores que cero sin aplicar la aritmética siguiente:  $X*Y$ , ni predicados predefinidos, sino recursividad. Recordar que  $X*Y = X + X + \dots + X$ , Y veces. Por ejemplo:  $5*3 = 5 + 5 + 5$ .

`producto2(X,Y,P)` (X e Y son los dos naturales que hay que dar, y P es el producto).

Por ejemplo:

`?- producto2(5,3,P).`

`P = 15 ;`

`false.`

### Ejercicio Nº 7.-

Escribir un programa en PROLOG, que ante la siguiente pregunta: ?- fibonacci(N, X). responda con el valor de X igual al N-ésimo término de la serie de Fibonacci. Para ello, téngase en cuenta que la numeración (N) empieza por cero, de tal manera que,  $f(0)=1$ ,  $f(1)=1$ ,  $f(2)=2$ ,  $f(3)=3$ ,  $f(4)=5$ , y, en general,  $f(n>1)=f(n-1)+f(n-2)$ .

### Ejercicio Nº 8.-

La sucesión de Padovan es la secuencia de números enteros  $P(n)$  definida por los siguientes valores iniciales  $P(0) = P(1) = P(2) = 1$ , y la siguiente relación de recurrencia

$$P(n) = P(n-2) + P(n-3).$$

Los primeros valores de  $P(n)$  son: 1,1,1,2,2,3,4,5,7,9,12,16,21,28,37, ...

Obtener el elemento N de la sucesión.

$p(N,R)$  (N es la posición del elemento de la sucesión y R su valor). Las posiciones comienzan en 0. Por ejemplo:

?-  $p(7,R)$ .

R = 5;

false.

### Ejercicio Nº 9.-

Los números de Perrin están definidos por la relación de recurrencia:

$$P(0) = 3, P(1) = 0, P(2) = 2,$$

y

$$P(n) = P(n-2) + P(n-3) \text{ si } n > 2.$$

La serie comienza: 3, 0, 2, 3, 2, 5, 5, 7, 10, 12, 17, 22, 29, 39...

Obtener el elemento N de la sucesión.

$p(N,R)$  (N es la posición del elemento de la sucesión y R su valor). Las posiciones comienzan en 0. Por ejemplo:

?-  $p(8,R)$ .

R = 10;

false.

### Ejercicio Nº 10.-

Construir un programa en PROLOG que, ante la siguiente pregunta: ?- sucesion(N,X). responda con el valor de X igual al N-ésimo término de la sucesión que se calcula de la siguiente forma:

$$s(0)=1.$$

Para las posiciones pares,  $s(N)=s(N-1)*2$ .

Para las posiciones impares,  $s(N)=(s(N-1)*2)+1$ .

Por ejemplo:

?-sucesion(5,X).

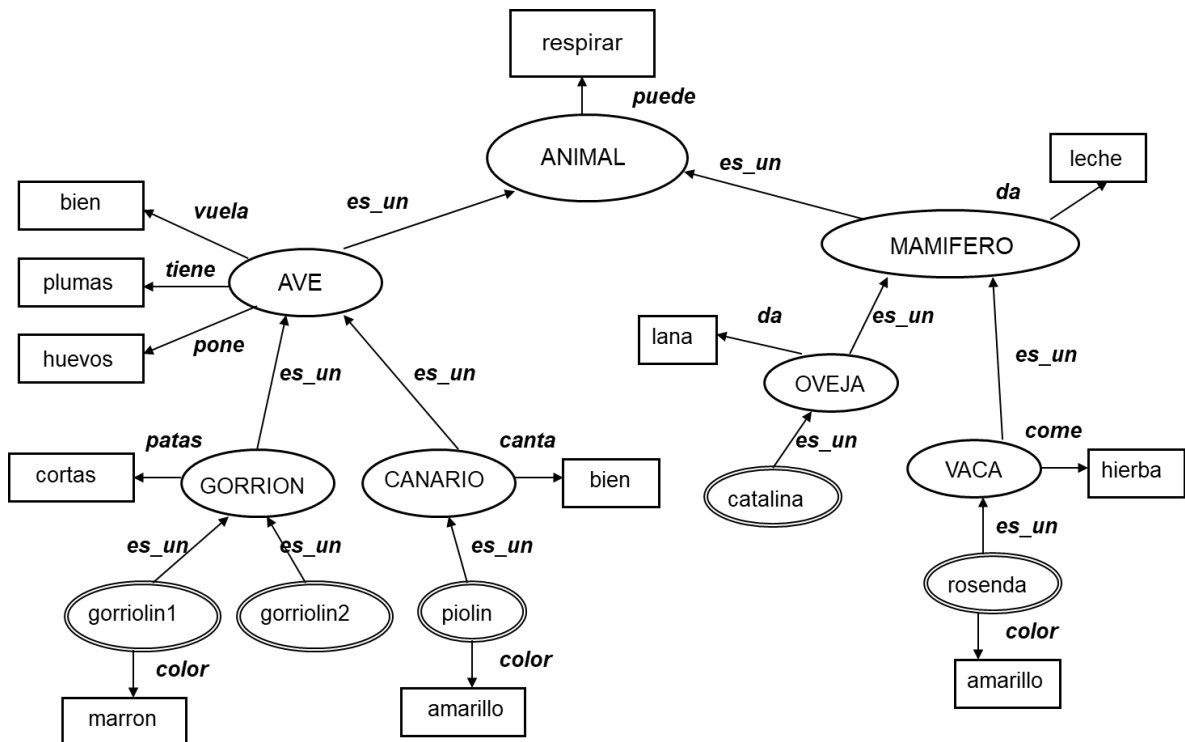
X=53 ;

false.

## Ejercicio Nº 11.-

Representar en Prolog la red semántica de la figura. Se sugiere emplear un predicado binario por cada relación (las de **es\_un**). Por ejemplo: **es\_un(ave,animal)**. Para las propiedades se sugiere que se emplee el predicado **atributo(Objeto, Atributo, Valor)**. Por ejemplo: **atributo(ave,vuela,bien)**.

Incluir las reglas necesarias para que todo objeto herede los atributos de todas las clases a las que pertenece. Se deben poder hacer consultas sobre la información almacenada en la red semántica. Por ejemplo, ¿puede respirar una vaca?, ¿quiénes vuelan bien?, etc.



## Ejercicio Nº 12.-

Considera el siguiente programa Prolog:

```

f(1,one).
f(s(1),two).
f(s(s(1)),three).
f(s(s(s(X))),N):-f(X,N).
  
```

¿Qué respuesta dará Prolog a las siguientes preguntas? Cuando sean posibles varias respuestas, dar al menos tres.

Pensarlo y razonarlo sin ejecutarlo. Comprobar el resultado ejecutándolo. Realizar el diagrama de obtención de las respuestas en todos los casos.

- (a) ?- f(s(1),A).
- (b) ?- f(s(s(1)),two).
- (c) ?- f(s(s(s(s(s(s(1))))))),C).
- (d) ?- f(D,three).

**f(s(s(s(X),N):-f(X,N)**