Tema: Paradigma Lógico - SWI-Prolog

Objetivos: Introducción. Lenguaje Prolog. Conceptos básicos, características. Axiomas, Predicados y Clausulas de Horn o reglas de inferencias, functor.

1. Problemas a resolver en clase

1.- Clausulas de Horn

Dada la siguiente base de conocimiento:

```
amigos(pedro, antonio).
amigos(pedro, flora).
amigos(pedro, juan).
amigos(pedro, vicente).
amigos(luis, felipe).
amigos(luis, maria).
amigos(luis, vicente).
amigos(carlos, paloma).
amigos(carlos, lucia).
amigos(carlos, juan).
amigos(carlos, vicente).
amigos(fernando, eva).
amigos(fernando, pedro).
```

```
millonario(pedro).
millonario(antonio).
millonario(flora).

soltero(pedro).
soltero(flora).
soltero(eva).
soltero(luis).

padre_de(carlos, fernando).
padre_de(antonio, maria).
padre_de(antonio, carlos).
```

Se pide a través de una consulta extraer la siguiente información:

- ☑ Definir una regla que determine quienes son hermanos.

```
padre_de(carlos, fernando).
padre_de(antonio, maria).
padre_de(antonio, carlos).

hermano_de(A,B) :- padre_de(C,A), padre_de(C,B) .
```

SWI-Prolog -- d:/FACULTAD/FACULTAD/NIVEL 2/Paradigmas de Programación/PROLOG/claus...

File Edit Settings Run Debug Help

```
?- hermano_de('maria','carlos').
true.
```

- ☑ Mi amigo, Vicente, busca amigos/as de mis amigos que sean millonarios/as y estén solteros/as.

```
amigos_de_vicente(B,vicente) :-
      amigos(A,B),
      amigos(A,vicente),
      soltero(B),
      millonario(B).
```

```
?- amigos_de_vicente('flora','vicente').
true .

?- amigos_de_vicente('antonio','vicente').
false.
```

☑ Determinar a través de una regla si una persona es pobre, considerando pobre a todo aquel que no es millonario.

```
pobre(A) :- not(millonario(A)).
```

```
n compiled 0.00 sec, 1 clauses
?- pobre('pedro').
false.

?- pobre('juan').
true.
```

☑ Se considera una persona interesante si no tiene hijos y es millonario, crear la regla necesaria.

```
padre(A) :- padre_de(A,B).
interesante(A) :- not(padre(A)), millonario(A).
```

```
?- interesante('carlos').
false.

?- interesante('antonio').
false.

?- interesante('pedro').
true.
```

2.- Realiza un programa que dado un signo del horóscopo nos muestre el día y mes de inicio y fin de ese signo.

Aries	21/3 al 21/4
Tauro	21/4 al 21/5
Géminis	21/5 al 21/6
Cáncer	21/6 al 21/7
Leo	21/7 al 21/8
Virgo	21/8 al 21/9
Libra	21/9 al 21/10
Escorpio	21/10 al 21/11
Sagitario	21/11 al 21/12
Capricornio	21/12 al 21/1
Acuario	21/1 al 21/2

☑ Escribir una regla que permita calcular el signo del Zodiaco para un día y un mes concreto, por ejemplo: `signo(Dia,Mes,Signo)`

```
aguario(A,B) :- (A >=21,A <30, B =:= 1);(A >= 1,A < 21, B =:=2).
pisis(A,B) :- (A>=21,A<30, B=:=2);(A>=1,A<21, B=:=3).
aries(A,B) :- (A>=21,A<30, B=:=3);(A>=1,A<21, B=:=4).
tauro(A,B) :- (A>=21,A<30, B=:=4);(A>=1,A<21, B=:=5).
geminis(A,B) :- (A>=21,A<30, B=:=5);(A>=1,A<21, B=:=6).
cancer(A,B) :- (A>=21,A<30, B=:=6);(A>=1,A<21, B=:=7).
leo(A,B) :- (A>=21,A<30, B=:=7);(A>=1,A<21, B=:=8).
virgo(A,B) :- (A>=21,A<30, B=:=8);(A>=1,A<21, B=:=9).
libra(A,B) :- (A>=21,A<30, B=:=9);(A>=1,A<21, B=:=10).
escorpio(A,B) :- (A>=21,A<30, B=:=10);(A>=1,A<21, B=:=11).
sagitario(A,B) :- (A>=21,A<30, B=:=11);(A>=1,A<21, B=:=12).
capricornio(A,B) :- (A>=21,A<30, B=:=12);(A>=1,A<21, B=:=1).
```

3.- Dada la siguiente base de conocimiento con los Jugadores de Futbol:

```
jugador(maradona).
jugador(chamot).
jugador(balbo).
jugador(caniggia).
jugador(passarella).
jugador(pedemonti).
jugador(basualdo).

% relaciona lo que toma cada jugador
tomo(maradona, sustancia(efedrina)).
tomo(maradona, compuesto(cafeVeloz)).
tomo(caniggia, producto(cocacola, 2)).
tomo(chamot, compuesto(cafeVeloz)).
tomo(balbo, producto(gatoreit, 2)).

% relaciona la máxima cantidad de un producto que 1 jugador puede ingerir
maximo(cocacola, 3).
maximo(gatoreit, 1).
maximo(naranju, 5).

% relaciona las sustancias que tiene un compuesto
composicion(cafeVeloz, [efedrina, ajipupa, extasis, whisky, cafe]).

% sustancias prohibidas por la asociación
sustanciaProhibida(efedrina).
sustanciaProhibida(cocaina).
```

1) Hacer lo que sea necesario para incorporar los siguientes conocimientos:
   a) passarella no toma nada que tome Maradona

```
pasarela(A) :- not(tomo(maradona,sustancia(A))),
               not(tomo(maradona,compuesto(A))).
```

   b) pedemonti toma todo lo que toma chamot y lo que toma Maradona

```
tomoPedemonti(A) :- ((tomo(maradona,compuesto(A))),(tomo(chamot,compuesto(A)))).
```

   c) basualdo no toma coca cola.

```
producto(cocacola).
toma_basualdo(A) :- not(producto(A)).
```

2) Definir el predicado puedeSerSuspendido/1 que relaciona si un jugador puede ser suspendido en base a lo que tomó. El predicado debe ser inversible.

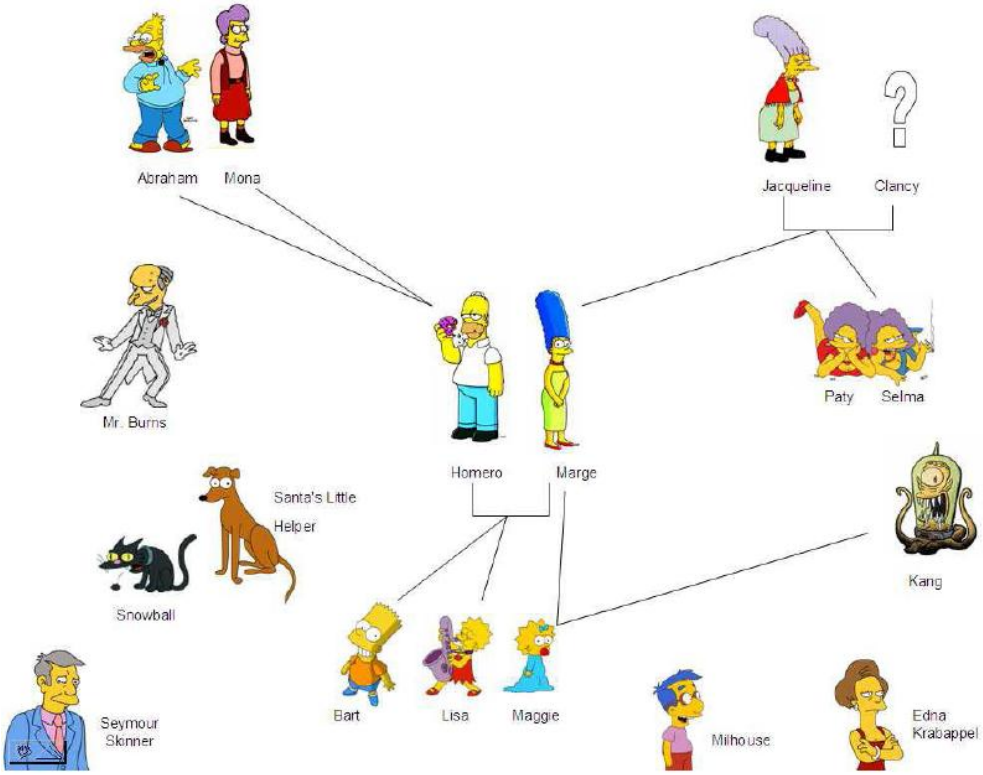
- un jugador puede ser suspendido si tomó una sustancia que está prohibida
- un jugador puede ser suspendido si tomó un compuesto que tiene una sustancia prohibida
- o un jugador puede ser suspendido si tomó una cantidad excesiva de un producto (más que el máximo permitido)

```
suspendido1(A,B) :- jugador(A),sustanciaProhibida(B).
suspendido2(A,B) :- jugador(A),composicion(B,_).
suspendido3(A,B,C) :- suspendido1(A,B); maximo(B,C).
suspendido(A,B,C) :- suspendido2(A,B); suspendido3(A,B,C).
```

2. Problemas Propuestos

Clausulas de Horn

1.- Tomando el grafico de la serie animada "Los Simpsons" represente las características de los objetos y las relaciones entre ellos.



Asimismo, basadas en las relaciones "progenitor" y el género de las personas (hombre o mujer), establecer reglas para:

abuelo(X,Y).
abuela(X,Y).
tio(X,Y).
tia(X,Y).

hermano(X,Y).

hermana(X,Y).

2.- Escribir un programa procedimental que, dados tres valores: [a, b y c], responda si es posible construir un triángulo cuyos lados tengan longitud a, b y c.

De ser así, deberá indicar también el tipo de triángulo: escaleno, isósceles o equilátero.

La caberecera que se definiría en C++ sería:

FUNCION TrianguloQ (X, Y, Z : REAL) : BOOLEAN ; ...

Luego escribir un programa equivalente en Prolog.

trianguloq(A,B,C) …

```
escaleno(A,B,C) :- (A =\= B),(B =\= C).
isoceles(A,B,C) :- (A =:= B), (B =\= C).
equilatero(A,B,C) :- (A =:= B),(B =:= C).

triangulo(A,B,C) :- (escaleno(A,B,C);isoceles(A,B,C);equilatero(A,B,C)).
```

3.- Una empresa de servicios varios ofrece la posibilidad de realizar compras varias desde un sitio

en Internet. Los usuarios solo deben registrarse e ir eligiendo cada artículo, como también su cantidad.
Este sistema corre un proceso al finalizar el día que genera la siguiente base de conocimientos:

```
precioUnitario(producto(tomate), 12.50).
precioUnitario(producto(leche, sancor), 2.45).
precioUnitario(producto(papa), 4.50).
precioUnitario(producto(yogur, laSerenisima), 1.75).
precioUnitario(producto(yogur, sancor), 1.65).
precioUnitario(producto(yogur, manfrey), 1.15).
precioUnitario(producto(fósforos, los3Patitos), 1).

El producto se representa como un functor con:
• El nombre de un producto genérico, o bien
• El nombre de un producto y la marca que lo comercializa

compro(leo, producto(tomate), 2).
compro(leo, producto(yogur, manfrey), 10).
compro(leo, producto(papa), 1).
compro(nico, producto(tomate), 3).
compro(flor, producto(yogur, laSerenisima), 2).
compro(flor, producto(leche, sancor), 4).

marcaImportante(sancor).
marcaImportante(laSerenisima).
```

Se pide:
1) Realice las consultas que permitan determinar a. quiénes compraron productos de Sancor (debe devolver los individuos leo y flor)

```
?- compro(X,producto(_, sancor), _).
X = flor.
```

a. qué compró Leo

```
?- compro(leo,producto(X),_).
X = tomate ;
X = papa ;
```

b. si Leo compró 2 cosas de algún producto (debe decirme que sí).

```
?- compro(leo,producto(_), 2).
true
```

2) Resuelva el predicado cuantoGasto/2 que relaciona una persona con el total que gastó.
? cuantoGasto(flor, Total)
Total = 13.5 (3.5 de los dos yogures y 10 de las cuatro leches)

```
?- cuantoGasto(flor,X).
X = 3.5 ;
X = 9.8 ;
```

```
cuantoGasto(flor,Total) :- compro(flor,A,L),precioUnitario(A,H), Total is L*H.
```

Problemas adicionales.

1.- Supongamos que tenemos el siguiente conocimiento sobre divisibilidad:
"2 divide a 6"
"2 divide a 12"
"3 divide a 6"
"3 divide a 12"
"Si un número es divisible por 2 y por 3 entonces es divisible por 6"

```
divide(2,6).
divide(2,12).
divide(3,6).
divide(3,12).
divide(6,B) :- divide(2,B),divide(3,B).
```

Escribir un programa que represente este conocimiento y usarlo para responder a las siguientes preguntas:
(1) ¿Existe algún múltiplo de 2?

```
?- divide(2,X).
X = 6 ;
X = 12.
```

(2) ¿Cuáles son los divisores de 6?

```
?- divide(X,6).
X = 2 ;
X = 3 ;
X = 6 ;
```

(3) ¿Conocemos algún múltiplo de 6?

```
?- divide(6,X).
X = 6 ;
X = 12.
```

2.- Representar una base de conocimientos que relaciona platos con la lista de sus ingredientes. Por ejemplo, un bizcocho contiene como ingredientes: leche, azúcar, harina y huevo. En la base de datos también se recoge qué ingredientes hay disponibles en la cocina.

Definir "puedo_cocinar(X)" que debe ser cierto si se dispone de todos los ingredientes necesarios para el plato X.

```
alimentos(bizcocho([leche, azucar, harina, huevo])).

puedo_cocinar(A):- alimentos(A).
```

3.- Modificar la base de conocimientos anterior indicando la cantidad disponible de cada ingrediente en la cocina y, para cada plato, la cantidad necesaria de cada ingrediente.

Modificar "puedo_cocinar(X)" de forma que un plato pueda ser cocinado si cada uno de sus ingredientes esté disponible en cantidad suficiente.

```
alimento1(bizcocho([(leche,1),(azucar,2),(harina,1),(huevo,3)])).
alimento2(ensalada([(lechuga,1),(tomate,2),(huevo,1)])).

puedo_cocinar(A,B):- (alimento1(A), B =:= 4);(alimento2(A), B =:= 3).
```

4.- Abrir el ejercicio 3 de los realizados en clases y agregamos los siguientes hechos

```
amigo(maradona, caniggia).
amigo(caniggia, balbo).
amigo(balbo, chamot).
amigo(balbo, pedemonti).
```

a) defina el predicado malaInfluencia/2 que relaciona dos jugadores, si ambos pueden ser suspendidos y se conocen, ya sea:
   - porque son amigos
   - porque son amigos de otra persona que lo conoce

En el ejemplo, maradona se conoce con caniggia, pero también con balbo, chamot y pedemonti.

Ejemplos:

```
? malaInfluencia(maradona, Quien).
devuelve entre otras soluciones
Quien = chamot ;
Quien = balbo ;
```

El predicado debe funcionar a n niveles de profundidad posibles.
El predicado debe ser inversible para ambos argumentos.

```
?- malaInfluencia(maradona, balbo).
true .

?- malaInfluencia(maradona, caniggia).
false.

?- malaInfluencia(maradona, X).
X = balbo
```

```
amigo(maradona, caniggia).
amigo(caniggia, balbo).
amigo(balbo, chamot).
amigo(balbo, pedemonti).


suspendidos(balbo).
malaInfluencia(A,B):- (amigo(A,B);amigo(_,B);amigo(B,_)),suspendidos(B).
```