1. Hello World

No se puede aprender un nuevo lenguaje de programación sin hacer el Hello World. Por tanto tendremos que hacerlo, entra al intérprete y escribe:

```
write('Hello World').
```

2. Sócrates

Haremos ahora el famoso ejemplo de «Todos los hombres son mortales, Sócrates es hombre» para ver si de esto se deduce "Sócrates es mortal". Abre un archivo nuevo y escribe:

```
mortal(X) :- hombre(X).
hombre(socrates).
```

Consultar el archivo y pregunta:

```
mortal(socrates).
```

3. Parentesco Familiar

3.1.

Ahora trabajaremos con un ejemplo clásico: definir relaciones de parentesco familiar. Ingresemos las siguientes cláusulas (o el árbol genealógico familiar de cada uno):

```
padre(juan, luis).
padre(juan, lia).
padre(luis, jorge).
padre(luis, ines).
padre(jorge, diego).
```

Luego consultar:

```
padre(juan, X).
```

Hacer además, las siguientes consultas:

1. ¿Quién es el padre de Inés?
2. ¿Es Luis padre de Jorge?
3. ¿Es Jorge padre de María?
4. ¿Quién es el padre de María?
5. Indicar una consulta para que retorne la relación padre completa.
6. Consultar si Juan es padre.

Soluciones

1. `padre(X, ines).`
2. `padre(luis, jorge).`
3. `padre(jorge, maria).`
4. `padre(X, maria).`
5. `padre(X, Y).`
6. `padre(juan, X).`

3.2.

Definir la relación abuelopaterno de la siguiente manera:

```
abuelopaterno(X, Z) :- padre(X, Y), padre(Y, Z).
```

Realizar las siguientes consultas:

1. ¿Quién es el abuelo de Diego?
2. Obtener todos los pares de la relación abuelopaterno.

Soluciones

1. `abuelopaterno(X, diego).`

2. `abuelopaterno(X, Y).`

Incorporar los siguientes hechos:

```
madre(ana, luis).
madre(lia, maria).
madre(lia, rosa).
```

3.3.

Definir la relación abuelomaterno.

Solución

```
abuelomaterno(X, Z) :- padre(X, Y), madre(Y, Z).
```

3.4.

Definir la relación abuelo, de la siguiente manera:

```
abuelo(X, Y) :- abuelopaterno(X, Y).
abuelo(X, Y) :- abuelomaterno(X, Y).
```

Consultar por todos los pares de la relación abuelo.

Solución

```
abuelo(X, Y).
```

3.5.

Definir la relación hermano y obtener todos los pares de la misma.

Solución

```
hermano(X,Y) :- padre(Z,X), padre(Z,Y), X\=Y.
hermano(X,Y) :- madre(Z,X), madre(Z,Y), X\=Y.

%% hermano(X,Y).
```

3.6.

Definir la relación tío. Nota: hacemos distinción entre tíos directos y tíos políticos.

Solución

```
COMPLETAR.
```

3.7.

Defina la relación ancestro como clausura transitiva de las relaciones madre y padre según la siguiente definición:

«Una persona X es el ancestro de una persona Y; si X es el padre o madre de Y, o si existe una persona Z tal que Z es padre o madre de Y y además X es ancestro de ese mismo Z»

Solución

```
ancestro(X,Y) :- padre(X,Y); madre(X,Y).
ancestro(X,Y) :- (padre(Z,Y); madre(Z,Y)), ancestro(X,Z).
```

3.8.

La familia Adams está compuesta por Homero, Morticia, Pericles, Merlina, Tío Cosa, Tío Lucas, la Abuela y Largo. Homero es hermano de Lucas y Cosa, todos ellos son hijos de la Abuela. Morticia es esposa de Homero y tiene dos hijos: Pericles y Merlina.

Generar las reglas y hechos en Prolog utilizando los siguientes predicados:

- varon(nombre)
- mujer(nombre)
- progenitor(nombre, nombre)
- padre(nombre, nombre)
- madre(nombre, nombre)
- hijo(nombre, nombre)
- hija(nombre, nombre)
- hermana(nombre, nombre)
- tio(nombre, nombre)
- abuela(nombre, nombre)

Se deben responder las siguientes consultas:

1. ¿Es Pericles hijo de Homero?
2. ¿Quién es hija de Homero?
3. ¿Quién es el padre de Pericles?
4. ¿Quiénes son tíos de la familia?
5. ¿Quién es la abuela de Pericles?
6. ¿Quién tiene hermana en la familia?

Solución

```
%% HECHOS
varon(homero).
varon(pericles).
varon(cosa).
varon(lucas).
varon(largo).

mujer(morticia).
mujer(merlina).
mujer(abuela).
```

```
hijo(homero,abuela).
hijo(lucas,abuela).
hijo(cosa,abuela).
hijo(pericles,morticia).
hijo(pericles,homero).

hija(merlina,homero).
hija(merlina,morticia).

hermano(homero,lucas).
hermano(homero,cosa).

%% REGLAS
hermano(X,Y) :- progenitor(Z,X), progenitor(Z,Y), varon(X), X \= Y.

hermana(X,Y) :- progenitor(Z,X), progenitor(Z,Y), mujer(X), X \= Y.

progenitor(X,Y) :- hijo(Y,X).
progenitor(X,Y) :- hija(Y,X).

abuela(X,Z) :- progenitor(X,Y), progenitor(Y,Z), mujer(X).

padre(X,Y) :- progenitor(X,Y), varon(X).

tio(X,Y) :- hermano(X,Z), progenitor(Z,Y), varon(X).
```

1. `hijo(pericles,homero).`
2. `hija(X,homero).`
3. `padre(X,pericles).`
4. `tio(X,Y).`
5. `abuela(X,pericles).`
6. `hermana(X,Y).`

4. Club de Ping-pong

4.1.

Realice un programa en Prolog que a partir del predicado `jugador(nombre,edad)` y los siguientes hechos:

```
jugador(pedro,9).
jugador(pablo,10).
jugador(cristian,9).
jugador(susana,9).
```

organice un torneo de ping-pong entre niños de 9 años.

A partir de este ejemplo discutiremos los pasos que sigue PROLOG en la resolución del problema.

Solución COMPLETAR.

4.2. Control de Backtracking

En el ejercicio anterior, notamos que PROLOG devuelve algunos resultados redundantes. Para controlar el backtracking hay dos mecanismos: el predicado *fail* y el predicado *cut* (representado por !).

Es mucho lo que hay por decir sobre estos mecanismos. Por ahora se pretende solamente intentar la modificiación del programa del ejercicio anterior para evitar los resultados redundantes. y analizar el siguiente ejemplo que realiza un proceso repetitivo con el uso del fail:

```
country("England").
country("France").
country("Germany").
country("Denmark").

print_countries:-
    country(X),
    write(X), /* write the value of X */
    nl,       /* start a new line */
    fail.

print_countries.
```

Solución COMPLETAR.

5. El predicado not

Es importante notar que el predicado *not* tiene éxito si la meta no se puede probar verdadera.

Realizar un programa que decida a quién le gusta realizar compras, sabiendo que si una persona tiene tarjeta de crédito y no está vencida disfruta de hacer compras y que: Chris tiene las tarjetas Visa y Diners, Joe y Sam tienen Mastercard y Sam también tiene Citibank. Además se sabe que Chirs tiene vencidas ambas tarjetas y que Sam tiene vencida la Mastercard.

Solución

```
tiene(chris,visa).
tiene(chris,diners).
tiene(joe,mastercard).
tiene(sam,mastercard).
tiene(sam,citibank).

vencida(chris,visa).
vencida(chris,diners).
vencida(sam,mastercard).

disfrutacompras(X) :- tiene(X,Y), not(vencida(X,Y)).
```

6. Muerte de un gato

- Juan tiene un perro y Pedro tiene un gato.
- Todos los que tienen una mascota aman a los animales.
- Nadie que ame a los animales los mata.
- Juan, Pedro o María mataron a la gata de Luis que se llama Iris.

Probar que María mató a Iris, usando resolución. Resolver utilizando PROLOG.

Solución Practica 1.

7. Señor de los anillos

Practica 1.

8. Fauna

Dados los siguientes hechos y relaciones,

1. Un ungulado es un animal.
2. Un pez es un animal.
3. Una cebra es un ungulado.
4. Un arenque es un pez.
5. Un tiburón es un pez.
6. La cebra vive en la tierra.
7. La rana vive en la tierra y en el agua.
8. Los peces viven en el agua.
9. Un animal que vive en el agua puede nadar.

realizar un programa PROLOG que decida que animal puede nadar (seguir el programa con el Trace).

Solución

```
animal(X) :- unglado(X). animal(X) :- pez(X).

unglado(X) :- cebra(X).

pez(arenque). pez(tiburon).

vive(cebra,tierra).
vive(rana,tierra).
vive(rana,agua).
vive(X,agua) :- pez(X).

nada(X) :- vive(X,agua).
```

9. Comidas

Considérense las siguientes sentencias:

- A John le gusta toda clase de comida.
- Las manzanas son comida.
- El pollo es comida.
- Cualquier cosa que uno coma y no le mate es comida.
- Bill come cacahuetes y aún está vivo.
- Sue come todo lo que come Bill.

Hacer un programa PROLOG y mostrar que a John le gustan los cacahuetes.

Solución

```
comida(manzana).
comida(pollo).
comida(Y) :- come(X,Y), nomata(Y,X).

come(bill,cacahuetes).
come(sue,X) :- come(bill,X).

nomata(cacahuetes,bill).

gusta(john,X) :- comida(X).
```

10. Club de bridge

- Los miembros del club de bridge de la calle Elm son Joe, Sally, Bill y Ellen.
- Joe está casado con Sally.
- Bill es hermano de Ellen.
- El cónyuge de cada persona casada del club también está en el club.
- La última reunión fue en la casa de Joe.

A partir de estos hechos (y algunos hechos de conocimiento del dominio) realizar un programa que determine si «La última reunión fue en casa de Sally».

Solución COMPLETAR.