

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Paradigmas de Programación

Práctica 1: Programación Lógica

Profesor:

Mtro. Rodríguez Castillo Miguel
Ángel

Alumnos:

Vázquez Blancas Cesar Said
Segundo Cruz Daniel

3CM5



Planteamiento del problema

Planteamiento del Problema 1

Descripción del problema: El objetivo de este problema es utilizar el lenguaje de programación prolog para modelar las relaciones de amor entre un conjunto de personas. Para ello, se utiliza el predicado `ama(X,Y)` que indica que X ama a Y. Se pide escribir un fichero con hechos que representen las relaciones de amor existentes entre las personas, y luego formular una serie de preguntas para consultar el fichero y obtener información sobre dichas relaciones.

Entrada: La entrada consiste en un fichero con hechos que indican quién ama a quién, utilizando el predicado `ama(X,Y)`.

Proceso: El proceso del problema consiste en escribir en prolog las siguientes preguntas, utilizando el operador “,” para indicar la conjunción lógica:

- ¿A quién ama “Juan”?
`ama(juan,X).`
- ¿Quién ama a “Ana”?
`ama(X,ana).`
- ¿Quién ama a alguien?
`ama(X,Y).`
- ¿Quién es amado por alguien?
`ama(Y,X).`
- ¿Quiénes se aman mutuamente?
`ama(X,Y),ama(Y,X).`
- ¿Quién ama sin ser correspondido?
`ama(X,Y),not(ama(Y,X)).`

Además, se pide añadir al fichero "lovers.pl" una regla que permita describir a los "lovers", es decir, aquellas personas que se aman mutuamente. Esta regla se puede escribir como:

```
lovers(X,Y) :- ama(X,Y),ama(Y,X).
```

Salida: La salida del problema consiste en mostrar por pantalla las respuestas a las preguntas formuladas en prolog. Por ejemplo, al consultar la pregunta ¿A quién ama "Juan"? se obtiene:

```
?- ama(juan,X). X = luis ; X = laura ; false.
```

Esto significa que Juan ama a Luis y a Laura, y que no hay más respuestas posibles. Al consultar la regla lovers(X,Y) se obtiene:

```
?- lovers(X,Y). X = ana, Y = miguel ; X = miguel, Y = ana ; X = diana, Y = laura ;  
X = laura, Y = diana ; false.
```

Esto significa que Ana y Miguel, Diana y Laura son parejas que se aman mutuamente, y que no hay más parejas posibles.

Planteamiento del Problema 2

Descripción del problema: El objetivo de este problema es utilizar el lenguaje de programación prolog para modelar las relaciones familiares entre un conjunto de personas. Para ello, se utiliza el predicado hombre(X) que indica que X es un hombre, y el predicado mujer(X) que indica que X es una mujer. Además, se utiliza el predicado matrimonio(X,Y) que indica que X e Y forman un matrimonio. Se pide escribir un fichero con hechos que representen las relaciones familiares existentes entre las personas, y luego definir una serie de reglas para consultar el fichero y obtener información sobre dichas relaciones.

Entrada: La entrada consiste en un fichero con hechos que indican quién es hombre, quién es mujer, quién es hijo o hija de quién, y quién forma matrimonio con quién, utilizando los predicados hombre(X), mujer(X), hijo(X,Y,Z), hija(X,Y,Z) y matrimonio(X,Y).

Proceso: El proceso del problema consiste en definir en prolog las siguientes reglas, utilizando el operador ":-" para indicar la implicación lógica y el operador ";" para indicar la conjunción lógica:

Definimos una regla que indique que "matrimonio" es reflexivo, es decir, si X e Y forman un matrimonio entonces Y y X también lo forman. Esta regla se puede escribir como:

```
matrimonio(X,Y) :- matrimonio(Y,X)
```

Definimos reglas para obtener:

- Los nietos de una persona:

`nieto(X,Z) :- hijo(X,Y,);hija(X,Y,),hijo(Y,Z,);hija(Y,Z,).`

- Los abuelos de una persona:

`abuelo(Z,X) :- nieto(X,Z).`

- Los hermanos de una persona:

`hermano(X,Y) :- hombre(X),hijo(X,Z,W);hija(X,Z,W),hijo(Y,Z,W);hija(Y,Z,W),X = Y.`

- Los tíos de una persona o las tías de una persona:

`tio(X,Y) :- hombre(X),hermano(X,Z),hijo(Y,Z,);hija(Y,Z,).`

`tia(X,Y) :- mujer(X),hermano(Z,X),hijo(Y,Z,);hija(Y,Z,).`

- Los primos de una persona:

`primo(X,Y) :- hijo(X,Z,);hija(X,Z,),tio(Z,Y);tia(Z,Y).`

- Las primas de una persona:

`prima(X,Y) :- hija(X,Z,_),tio(Z,Y);tia(Z,Y).`

- Los suegros de una persona:

`suegro(X,Y) :- hombre(X),matrimonio(Y,Z),hijo(Z,X,);hija(Z,X,).`

Salida: La salida del problema consiste en mostrar por pantalla las respuestas a las consultas formuladas en prolog, utilizando el intérprete de prolog. Por ejemplo, al consultar la regla `abuelo(Z,X)` se obtiene:

`?- abuelo(Z,X). Z = antonio, X = ricardo ; Z = antonio, X = alicia ;`

`Z = ana, X = ricardo ; Z = ana, X = alicia ; false.` Esto significa que Antonio y Ana son los abuelos de Ricardo y Alicia, y que no hay más respuestas posibles.

Planteamiento del Problema 3

Descripción del problema: El objetivo de este problema es utilizar el lenguaje de programación prolog para modelar las relaciones laborales entre un conjunto de trabajadores de una empresa. Para ello, se utiliza el predicado `encargado_de_tarea(trabajador,tarea)` que indica que un trabajador está encargado de una tarea. Se pide declarar los siguientes hechos relativos a los trabajadores y las tareas que realizan, y luego definir una serie de reglas para consultar el fichero y obtener información sobre dichas relaciones.

Entrada: La entrada consiste en un fichero con hechos que indican quién está encargado de qué tarea, utilizando el predicado `encargado_de_tarea(trabajador,tarea)`.

Proceso: El proceso del problema consiste en definir en prolog las siguientes reglas, utilizando el operador `:-` para indicar la implicación lógica y el operador `;` para indicar la disyunción lógica:

Definimos una regla que permita comprobar si una tarea ha sido encargada a alguna persona. Utilizamos el predicado `encargada(Tarea)`. Esta regla se puede escribir como:

```
encargada(Tarea) :- encargado_de_tarea(_,Tarea).
```

Definimos una regla que permita comprobar si dos personas comparten alguna tarea, es decir, `comparten_tarea(Persona1,Persona2)`. Esta regla se puede escribir como:

```
comparten_tarea(Persona1,Persona2) :-  
encargado_de_tarea(Persona1,Tarea),encargado_de_tarea(Persona2,Tarea),Persona1 =  
Persona2.
```

Salida: La salida del problema consiste en mostrar por pantalla las respuestas a las consultas formuladas en prolog, utilizando el intérprete de prolog. Por ejemplo, al consultar la regla `comparten_tarea(Persona1,Persona2)` se obtiene:

```
?- comparten_tarea(Persona1,Persona2). Persona1 = miguel, Persona2 = alicia ; Persona1  
= alicia, Persona2 = miguel ; false.
```

Esto significa que Miguel y Alicia comparten la tarea de control, y que no hay más respuestas posibles,

Planteamiento del Problema 4

Descripción del problema: El objetivo de este problema es utilizar el lenguaje de programación prolog para modelar las relaciones entre personas y deportes que practican. Para ello, se utiliza el predicado `juega(Persona,Deporte)` que indica que una persona juega a un deporte. Se pide declarar los siguientes hechos relativos a los deportistas y los deportes que juegan, y luego definir una regla para consultar el fichero y obtener información sobre dichas relaciones.

Entrada: La entrada del problema consiste en un fichero que declara los deportes que juegan Héctor, Miguel y Alicia, utilizando el predicado `juega(Persona,Deporte)`. Por ejemplo, `juega(miguel,balonmano)` indica que Miguel juega al balonmano.

Proceso: El proceso del problema consiste en definir en prolog una regla que permita comprobar si dos personas juegan al mismo deporte, utilizando el operador “:-” para indicar la implicación lógica y el operador “,” para indicar la conjunción lógica. Esta regla se puede escribir como:

```
juegan_al_mismo_deporte(Persona1,Persona2) :-  
juega(Persona1,Deporte),juega(Persona2,Deporte),Persona1 = Persona2.
```

Salida: La salida del problema consiste en mostrar por pantalla las respuestas a las consultas formuladas en prolog, utilizando el intérprete de prolog. Por ejemplo, al consultar la regla `juegan_al_mismo_deporte(Persona1,Persona2)` se obtiene:

```
?- juegan_al_mismo_deporte(Persona1,Persona2). Persona1 = hector, Persona2 = alicia ;  
Persona1 = miguel, Persona2 = alicia ; Persona1 = alicia, Persona2 = hector ; Persona1  
= alicia, Persona2 = miguel ; false.
```

Esto significa que Héctor y Alicia juegan al baloncesto, Miguel y Alicia juegan al balonmano y al rugby, y que no hay más respuestas posibles.

Implementación del problema

Implementación del problema 1: Lovers

```
%Titulo:lovers

%descripcion:Escribe un fichero denominado "lovers.pl" que contenga los siguientes hechos
dondeelpredicadoama(X,Y)indicaqueXamaaY.Escribeenprologlassiguientespreguntas¿Aqu
iénama"Juan"?o¿Quiénamaa"Ana"?o¿Quiénamaaalguien?o¿Quiénesamadoporalguien?o¿Quiéne
sseamanmutuamente?o¿Quiénamasinsercorrespondido?Añadealfichero lovers.plunareglaque
permitadescribiralos "lovers",esdecir, aquellas personasqueseaman mutuamente.

%fecha:16/09/23

%version:1.5

%autor:Vazquez Blancas Cesar Said, Cruz Segundo Daniel

%Hechos

ama(ana,miguel).
ama(luis,isabel).
ama(diana,laura).
ama(luis,carlos).
ama(miguel,ana).
ama(laura,juan).
ama(juan,luis).
ama(isabel,carlos).
ama(laura,diana).
ama(laura,juan).

%Reglas

%regla para describir los lovers

%esta regla describe a los lovers como X y Y pues se quieren encontrar,
%por lo tanto si ama uno al otro seria como X ama a Y y para que sea mutuo,
%tendria que haber un Y ama a X, por lo que con eso seria suficiente
lovers(X,Y):-ama(X,Y),ama(Y,X).
```

Implementación del problema 2: Familia

%Titulo:familia

%descripcion:Escribe un fichero denominado “familia.pl”, Define hechos en los que se afirmen, Define reglas para nietos, abuelos, hermanos, tíos, tías, primos, primas y suegros

%fecha:16/09/23

%version:2.5

%autor:Vazquez Blancas Cesar Said, Cruz Segundo Daniel

%Hechos

%definimos a los hombres

%hombre(hombre)

hombre(antonio).

hombre(juan).

hombre(luis).

hombre(rodrigo).

hombre(ricardo).

%definimos las mujeres

%mujer(mujer)

mujer(isabel).

mujer(ana).

mujer(marta).

mujer(carmen).

mujer(laura).

mujer(alicia).

%definimos los matrimonios

%matrimonio(hombre,mujer)

matrimonio(antonio,ana).

matrimonio(juan,carmen).

matrimonio(luis,isabel).

matrimonio(rodrigo,laura).


```

hijo(marta,antonio,ana).
hijo(carmen,luis,isabel).
hijo(ricardo,juan,carmen).
hijo(alicia,rodrigo,laura).

%reglas
matrimonio_reflexivo(X,Y):-matrimonio(X,Y);matrimonio(Y,X).%con esta regla hacemos
que si el hombre
%esta casado con la mujer la mujer tambien, asi que la consulta puede ser regular
o inversa

nieto(Nieto, Abuelo) :-
(hombre(Abuelo), %si el abuelo es hombre entonces
hijo(Padre, Abuelo, _),%buscamos al hijo del abuelo
(mujer(Padre), hijo(Nieto, _, Padre); %verificamos si el hijo del abuelo es mujer
y si es buscamos al hijo de la mujer
hombre(Padre), hijo(Nieto,Padre, _)) %si no es mujer entonces buscamos al hijo del
hombre, osea el nieto
);%si es abuela entonces
(mujer(Abuelo),
hijo(Padre, _, Abuelo),%buscamos al hijo de la abuela
(mujer(Padre), hijo(Nieto, _, Padre); %verificamos si el hijo es mujer y si es
mujer buscamos a su hijo osea el nieto
hombre(Padre), hijo(Nieto, Padre, _))%verificamos si el hijo es hombre y si si
entonces buscamos a su hijo osea el nieto
).

abuelo(Abuelo,Nieto):-
nieto(Nieto,Abuelo).%utilizamos nietos para hacer la relacion al reves y ya esta

hermano(Persona,Hermano):-
hijo(Persona,Padre,Madre),%utilizamos a la persona y a los padres de esa persona
hijo(Hermano,Padre,Madre),%al ser los mismos padres solo relacionamos a los padres
de ambos y sacamos al hermano

```

Persona\=Hermano.%al ser hijo tambien de los padres la perosna puede aparecer en resultados pero con esta linea lo eliminamos

tio(Sobrino,Tio):-

hijo(Sobrino,Padre,Madre),%utilizamos al hijo para llegar a los padres

(hermano(Padre,Tio),hombre(Tio);%los hermanos del padre son los tios y verificamos con hombre

hermano(Madre,Tio),hombre(Tio)).%los hermanos de la madre son tios tambien y se verifican con hombre

tia(Sobrino,Tia):-

hijo(Sobrino,Padre,Madre),%utilizamos al hijo para llegar a los padres

(hermano(Padre,Tia),mujer(Tia);%las hermanas del padre son las tias y verificamos con mujer

hermano(Madre,Tia),mujer(Tia);

hermano(Padre,Tio),matrimonio_reflexivo(Tio,Tia)).%las hermanos de la madre son tias tambien y se verifican con mujer

primo(Persona,Primo):-

(tio(Persona,Tio),%buscamos al tio de la persona

hijo(Primo,Tio,_),hombre(Primo);%enocntramos al hijo del tio que es el primo y verificamos que sea hombre

tia(Persona,Tia),%buscamos a la tia de la persona

hijo(Primo,_,Tia),hombre(Primo)).%enocntramos al hijo de la tia que es el primo y verificamos que sea hombre

prima(Persona,Prima):-

(tio(Persona,Tio),%buscamos al tio de la persona

hijo(Prima,Tio,_),mujer(Prima);%encontramos a la hija del tio que es la prima y verificamos que sea mujer

tia(Persona,Tia),%buscamos a la tia de la persona

```
hijo(Prima,_,Tia),mujer(Prima)).%encontramos a la hijoa de la tia que es la prima
y verificamos que sea mujer
```

```
suegro(Persona,Suegros):-
```

```
matrimonio_reflexivo(Persona,Pareja),%utilizamos matrimonio reflexivo para que
cualquiera sea el sexo de la persona se busque como sea
```

```
(hijo(Pareja,Suegros,_);%el padre de la pareja es el primer suegro
```

```
hijo(Pareja,_,Suegros)).%la madre de la pareja es el segundo suegro
```

Implementación del problema 3: Trabajadores

```
%Titulo:trabajadores
```

```
%descripcion:Declarar lossiguientes hechos relativos a trabajadores de una
empresa.Utiliza el predicado encargado_de_tarea(trabajador,tarea) o Miguel está
encargado de las tareas de admisión,
control y vigilancia. o Ricardo está encargado de las tareas de planificación
y asesoramiento. o Alicia está encargada de la dirección y control. Definir reglas que permitan
comprobar lossiguientes hechos: o Si una tarea ha sido encargada a alguna persona. Utiliza
el predicado encargada(Tarea). o Si dos personas comparten alguna tarea,
es decir, comparten_tarea(Persona1,Persona2).
```

```
%fecha:16/09/23
```

```
%version:2
```

```
%autor:Vazquez Blancas Cesar Said, Cruz Segundo Daniel
```

```
%Hechos
```

```
%encargado_de_tarea(trabajador,tarea)
```

```
encargado_de_tarea(miguel,admission).
```

```
encargado_de_tarea(miguel,control).
```

```
encargado_de_tarea(miguel,vigilancia).
```

```
encargado_de_tarea(ricardo,planificacion).
```

```

encargado_de_tarea(ricardo,asesoramiento).
encargado_de_tarea(alicia,direccion).
encargado_de_tarea(alicia,control).

%reglas
encargada(Tarea):-
encargado_de_tarea(_,Tarea).%solo se dice la tarea y si existe mandara un true

comparten_tarea(Persona1,Persona2):-
encargado_de_tarea(Persona1,Tarea),%verificamos que una persona haga una tarea
encargado_de_tarea(Persona2,Tarea),%verificamos que otra perosna haga la misma
tarea
Persona1 \= Persona2.%hace que no se devuelva el mismo nombre que pones en persona

```

Implementación del problema 4

```

%Titulo:Deportistas
%descripcion:
Predicadosobredeportistas:Utilizaelpredicadójuega(Persona,Deporte)paradeclararlo
siguienteshechosrelativosadeportistas:oHéctorjuegaalbaloncesto.oMigueljuegaalbalon
manoyrugbyoAliciajuegaaltenis,baloncestoyajedrez.Defineunpredicadoquepermitacompro
barsidospersonas jueganalmismodeporte.
%fecha:16/09/23
%version:1.5
%autor:Vazquez Blancas Cesar Said, Cruz Segundo Daniel

```

```
%Hechos
%juega(Persona,Deporte).
juega(hector,baloncesto).
juega(miguel,balonmano).
juega(miguel,rugby).
juega(alicia,tenis).
juega(alicia,baloncesto).
juega(alicia,ajedrez).

%reglas
juegan_mismodeporte(Persona1,Persona2):-
juega(Persona1,Deporte),%verificamos que una persona juegue un deporte
juega(Persona2,Deporte),%persona diferente juega el mismo deporte
Persona1\=Persona2.%persona 1 y 2 deben de ser diferentes
```

Funcionamiento

Funcionamiento problema 1: Lovers

```
?- consult('C:/Users/pvl X = juan ;
true. X = isabel ;
X = laura ;
X = laura.

?- ama(juan,X).
X = luis.

?- ama(X,ana).
X = miguel.

?- ama(X,_).
X = ana ;
X = luis ;
X = diana ;
X = luis ;
X = miguel ;
X = laura ;
X = juan ;
X = isabel ;
X = laura ;
X = laura.

?- ama(_,X).
X = miguel ;
X = isabel ;
X = laura ;
X = carlos ;
X = ana ;
X = juan ;
X = luis ;
X = carlos ;
X = diana ;
X = juan.

?- ama(X,Y),ama(Y,X).
X = ana,
Y = miguel ;
X = diana,
Y = laura ;
X = miguel,
Y = ana ;
X = laura,
Y = diana ;
false.

?- ama(X,Y),\+ ama(Y,X).
X = luis,
Y = isabel ;
X = luis,
Y = carlos ;
X = laura,
Y = juan ;
X = juan,
Y = luis ;
X = isabel,
Y = carlos ;
X = laura,
Y = juan.

?- lovers(X,Y).
X = ana,
Y = miguel ;
X = diana,
Y = laura ;
X = miguel,
Y = ana ;
X = laura,
Y = diana ;
false.

?- ama(X,Y),\+ ama(Y,X).
X = luis,
Y = isabel ;
X = luis,
Y = carlos ;
X = laura,
Y = juan ;
X = juan,
Y = luis ;
false.
```

Funcionamiento problema 2: Familia

```
?- matrimonio_reflexivo(X,Y).  
X = antonio,  
Y = ana ;  
X = juan,  
Y = carmen ;  
X = luis,  
Y = isabel ;  
X = rodrigo,  
Y = laura ;  
X = ana,  
Y = antonio ;  
X = carmen,  
Y = juan ;  
X = isabel,  
Y = luis ;  
X = laura,  
Y = rodrigo.
```

```
?- nieto(X,ana).  
X = ricardo ;  
X = alicia ;  
false.
```

```
?- abuelo(X,ricardo).  
X = antonio ;  
X = luis ;  
X = isabel ;  
X = ana ;  
false.
```

```
?- hermano(rodrigo,X).  
X = juan ;  
X = marta.
```

```
?- tio(ricardo,X).  
X = rodrigo ;  
false.
```

```
?- tia(ricardo,X).  
X = marta ;  
X = laura ;  
false.
```

```
?- primo(alicia,X).  
X = ricardo ;  
X = ricardo ;  
false.
```

```
?- prima(ricardo,X).  
X = alicia ;  
X = alicia ;  
false.
```

```
?- suegro(juan,X).  
X = luis ;  
X = isabel ;  
false.
```

Funcionamiento problema 3: Trabajadores

```
?- encargada(admision).  
true.
```

```
?- encargada(contaduria).  
false.
```

```
?- comparten_tarea(miguel,ricardo).  
false.
```

```
?- comparten_tarea(miguel,alicia).  
true ;  
false.
```

```
?- comparten_tarea(miguel,X).  
X = alicia ;  
false.
```

```
?- encargada(X).  
X = admision ;  
X = control ;  
X = vigilancia ;  
X = planificacion ;  
X = asesoramiento ;  
X = direccion ;  
X = control.
```


Funcionamiento problema 4: Deportistas

```
?- juega(alicia,X).  
X = tenis ;  
X = baloncesto ;  
X = ajedrez.
```

```
?- juega(X,baloncesto).  
X = hector ;  
X = alicia.
```

```
?- juegan_mismodeporte(alicia,miguel).  
false.
```

```
?- juegan_mismodeporte(alicia,hector).  
true ;  
false.
```

```
?- juegan_mismodeporte(alicia,X).  
X = hector ;  
false.
```

Conclusiones Individuales

Vázquez Blancas Cesar Said

Esta práctica me ayudo a aplicar lo básico y aprender el lenguaje de PROLOG como tal, así como la declaración de hechos y como con estos se pueden formar reglas necesarias para obtener más comandos, la practica fue complicada por la habilidad de aprender un nuevo lenguaje pues la familiarización con él y su paradigma era nulo para mí, así que el aprender y buscar comando así como la sintaxis de este, el significado de la coma y el punto y coma es bastante complejo, hubo momentos donde por no saber que la sintaxis de las "variables" debían empezar en mayúscula me marcaba error así como también por no respetar el nombre de las mismas, sin embargo al investigar y con ayuda de las presentaciones me ayudo a desarrollar la práctica de manera correcta, una posible mejora sería la de no imprimir elementos repetidos y quitar los false, sin embargo aun así los programas funcionan como deberían y devuelven lo que deberían devolver.

Segundo Cruz Daniel

Con esta práctica tuve mi primera inmersión con el lenguaje PROLOG, que se basa en el paradigma de la programación lógica. Algunos de los conceptos que apliqué para resolver los problemas fueron:

- El uso de predicados, por ejemplo, el predicado ama(X,Y) representa que X ama a Y, y el predicado hombre(X) representa que X es un hombre.
- El uso de variables para representar objetos desconocidos, por ejemplo, la variable X puede representar a cualquier persona.
- El uso de operadores lógicos para combinar o modificar los predicados. Por ejemplo, el operador ";" para representar la disyunción, y el operador "," para representar la conjunción y el operador ":-" para representar la implicación lógica.

Algunas de las dificultades que tuve en el desarrollo fueron:

- Entender la sintaxis y la semántica de PROLOG.
- Adaptar mi forma de pensar al paradigma de la programación lógica.

Los errores en el transcurso fueron mas bien propios dada la carencia de practica y dado que estoy aprendiendo sobre la forma declarativa de trabajar con este tipo de problemas no tengo sugerencias para mejorar el trabajo realizado.

Bibliografía:

Morales, Eduardo. PROLOG. Recuperado el 18 de septiembre del 2023, de <https://ccc.inaoep.mx/~emorales/Cursos/Prolog/curso.pdf>

Toledo Lobo, Francisco, Pacheco Aparicio, Julio, Escrig Moneferrer, Teresa (2001). El Lenguaje de Programación PROLOG. Recuperado el 18 de septiembre de 2023, de <http://mural.uv.es/mijuanlo/PracticasPROLOG.pdf>

Universidad Autonoma Metropolitana, Tutorial básico de programación en Prolog. Recuperado el 18 de septiembre de 2023, de https://academicos.azc.uam.mx/cbr/Cursos/UEA_12p_Log/TutorialdePrologEspa.pdf

Rodríguez Castillo, M. A. *Programación Lógica (1, 2, 3)* [Diapositivas de PowerPoint], Escuela superior de cómputo, IPN.