

ACTIVIDAD04

1.1

Ejercicio 1.1

Dada la base de datos familiar del ejemplo 1.1, se pide la respuesta de PROLOG y el enunciado verbal de las siguientes preguntas:

- a) ?-progenitor(jaime,X).
- b) ?-progenitor(X,jaime).
- c) ?-progenitor(clara,X), progenitor(X,patricia).
- d) ?-progenitor(tomas,X), progenitor(X,Y), progenitor(Y,Z).

a). ¿Tiene Jaime hijo o hija?

```
?- progenitor(jaime,X).  
false.
```

b). ¿Quien es progenitor de Jaime?

```
?- progenitor(X,jaime).  
X = patricia.
```

c). ¿Quien es el hijo/a de clara que tambien es progenitor de patricia?

```
?- progenitor(clara,X) , progenitor(X,patricia).  
X = jose.
```

d). ¿Quien es el hijo, nieto y bisnieto de thomas?

```
?- progenitor(tomas,X) , progenitor(X,Y) , progenitor(Y,Z).  
X = jose,  
Y = patricia,  
Z = jaime .
```

1.2

Ejercicio 1.2

Dada la base de datos familiar del ejemplo 1.1, formula en PROLOG las siguientes preguntas:

- a) ¿Quién es el progenitor de Patricia?
- b) ¿Tiene Isabel un hijo o una hija?
- c) ¿Quién es el abuelo de Isabel?
- d) ¿Cuáles son los tíos de Patricia? (no excluir al padre)

a). progenitor(X,praticia).

```
?- progenitor(X,patricia).  
X = jose.
```

b). progenitor(isabel,X).

```
?- progenitor(isabel,X).  
false.
```

c). abuelo_de(X,isabel).

```
?- abuelo_de(X,isabel).  
false.
```

d). progenitor(X,patricia) , hermanos(X,Y).

```
?- progenitor(X,patricia) , hermanos(X,Y).  
X = jose,  
Y = isabel.
```

1.3Codigo:

```
%Actividad04
progenitor(clara,jose).
progenitor(tomas,jose).
progenitor(tomas,isabel).
progenitor(jose,ana).
progenitor(jose,patricia).
progenitor(patricia,jaime).

hombre(tomas).
hombre(jose).
hombre(jaime).

mujer(clara).
mujer(isabel).
mujer(ana).
mujer(patricia).

dif(X,Y) :- X\=Y.

%es_madre(X)
es_madre(X):- mujer(X), progenitor(X,_).

%es_padre(X)
es_padre(X) :- progenitor(X,_), hombre(X), !.

%es_hijo(X)
es_hijo(X) :- progenitor(_,X), !.

%hermana_de(X,Y)
hermana_de(X,Y):- mujer(X), dif(X,Y), progenitor(Z,X), progenitor(Z,Y), !.

%abuelo_de(X,Y)
abuelo_de(X,Y):- ( progenitor(Z,Y), progenitor(X,Z)), hombre(X), !.

%abuela_de(X,Y)
abuela_de(X,Y):- ( progenitor(Z,Y), progenitor(X,Z)), mujer(X), !.
.
%hermanos(X,Y)
hermanos(X,Y):-
    progenitor(Z,X), progenitor(Z,Y),
    dif(X,Y).
```

```
%tia(X,Y)
tia(X,Y) :- progenitor(Z,Y), hermana_de(X,Z), !.
```

dif(X,Y)

```
?- dif(jose,jose).  
false.  
  
?- dif(jose,patricia).  
true.
```

es_madre(X)

```
?- es_madre(clara).  
true.  
  
?- es_madre(jaime).  
false.
```

es_padre(X)

```
?- es_padre(patricia).  
false.  
  
?- es_padre(jose).  
true.
```

es_hijo(X)

```
?- es_hijo(tomas).  
false.  
  
?- es_hijo(ana).  
true.
```

hermana_de(X,Y)

```
?- hermana_de(ana, patricia).  
true.  
  
?- hermana_de(isabel, jaime).  
false.
```

abuelo_de(X,Y)

```
?- abuelo_de(jose, patricia).  
false.  
  
?- abuelo_de(tomas, ana).  
true.
```

abuela_de(X,Y)

```
?- abuela_de(clara, ana).  
true.  
  
?- abuela_de(isabel, jaime).  
false.
```

hermanos(X,Y)

```
?- hermanos(ana, patricia).  
true.  
  
?- hermanos(ana, jose).  
false.
```

tia(X,Y)

```
?- tia(ana, jaime).  
true.  
  
?- tia(patricia, jaime).  
false.
```

Conclusiones

Durante el desarrollo de la actividad, se identificaron retos como la estructuración correcta de reglas y hechos, lo que permitió afianzar la habilidad de modelar problemas utilizando lógica de primer orden.

Arboles SDL