

Informe: Actividad 1 "Enunciados". IS453-G1

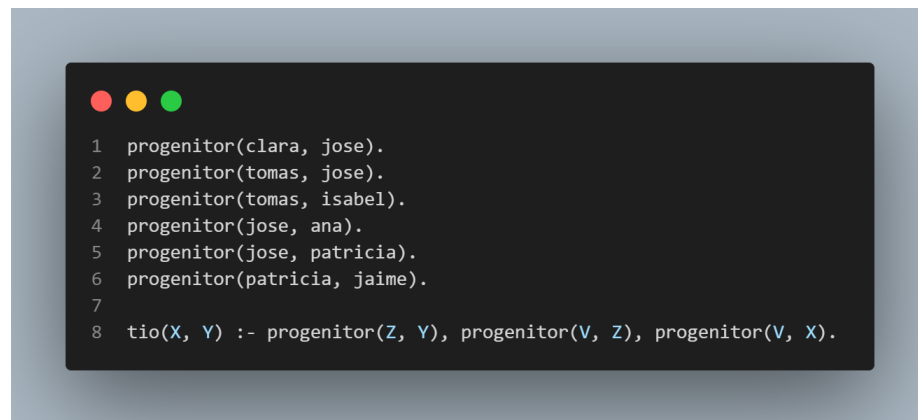
Kevin Esguerra Cardona - 1004 699 587

February 17, 2023

En el presente documento se consigna el informe de la actividad número 1 realizada en el curso de Programación 3 (Enunciados) en la cual se le pide al estudiante realizar una serie de consultas simples al intérprete del lenguaje prolog, previa base de datos suministrada, e interpretar tanto las peticiones enviadas cómo las respuestas recibidas.

Base de datos.

Esta, es una base datos sencilla que contiene el arbol genealógica de una familia.

A screenshot of a Prolog interpreter window. The window has a dark background with three colored window control buttons (red, yellow, green) in the top-left corner. It contains a list of Prolog facts and a rule, numbered 1 through 8. The facts define the 'progenitor' relationship for various family members. The rule defines the 'tio' (uncle) relationship based on the 'progenitor' facts.

```
1 progenitor(clara, jose).
2 progenitor(tomas, jose).
3 progenitor(tomas, isabel).
4 progenitor(jose, ana).
5 progenitor(jose, patricia).
6 progenitor(patricia, jaime).
7
8 tio(X, Y) :- progenitor(Z, Y), progenitor(V, Z), progenitor(V, X).
```

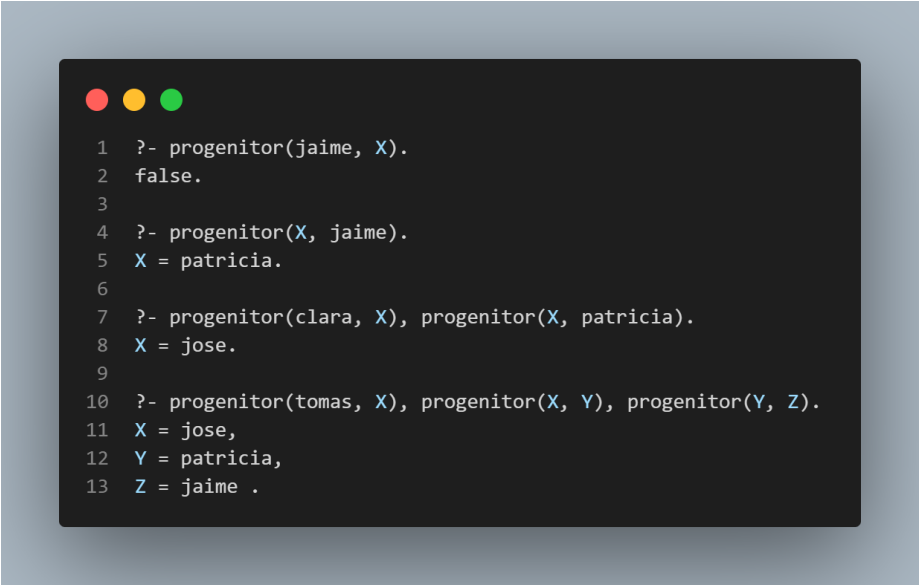
Figure 1: Base de datos utilizada en todos los ejercicios de la actividad.

Ejercicio primero, resultados.

En este ejercicio se pide la respuesta de prolog y el enunciado verbal a las siguientes consultas:

1. ?-progenitor(jaime, X).
2. ?-progenitor(X, jaime).
3. ?-progenitor(clara, X), progenitor(X, patricia).
4. ?-progenitor(tomas, X), progenitor(X, Y), progenitor(Y, Z).

Peticiones, cuyas respuestas, en orden de llamada, fueron:



```
1  ?- progenitor(jaime, X).
2  false.
3
4  ?- progenitor(X, jaime).
5  X = patricia.
6
7  ?- progenitor(clara, X), progenitor(X, patricia).
8  X = jose.
9
10 ?- progenitor(tomas, X), progenitor(X, Y), progenitor(Y, Z).
11 X = jose,
12 Y = patricia,
13 Z = jaime .
```

Figure 2: Resultados del primer ejercicio.

Y, en el mismo orden, sus posibles enunciados verbales serían:


1. ¿Jaime es progenitor de quién? R: false. Es decir, Jaime no posee progeñie.
2. ¿Jaime es progeñie de quién? R: Patricia. Es decir, Jaime es progeñie de Patricia.
3. ¿Quién progenitor de Patricia, que a su vez, es progeñie de Clara? R: Jose. Es decir, Jose es progeñie de Clara y progenitor de Patricia.
4. ¿Quién progeñie de la progeñie de la progeñie de Tomas? R: Jaime. Es decir, La progeñie de Tomas, es progenitor del progenitor de Jaime.

Ejercicio segundo, resultados

En este ejercicio se le pide al estudiante formular al interprete de prolog las siguientes preguntas, dadas en lenguaje natural:

1. ¿Quién es el progenitor de Patricia?
2. ¿Tiene Isabel un hijo o una hija?
3. ¿Quién es el abuelo de Isabel?
4. ¿Cuáles son los tíos de Patricia? (no excluir al padre)

Preguntas, cuyas interpretaciones y resultados se ven a continuación:



```
1  ?- progenitor(X, patricia).
2  X = jose.
3
4  ?- progenitor(isabel, X).
5  false.
6
7  ?- progenitor(X, isabel), progenitor(Abuelo, X).
8  false.
9
10 ?- tio(X, patricia).
11 X = jose ;
12 X = jose ;
13 X = isabel.
```

Figure 3: Resultados del segundo ejercicio.

Cabe a aclarar, que en el último caso jose aparece dos veces debido a la naturaleza no excluyente de la formulación realizada a prolog, pues esta busca los hijos de los abuelos de patricia, patricia tiene dos abuelos que son padres de jose, por esta razón jose aparece dos veces en los resultados.

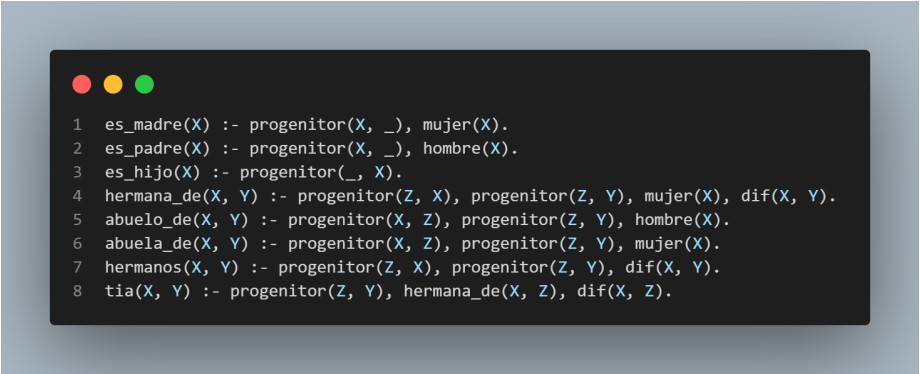
Ejercicio tercero, resultados

En este ejercicio se le pide al estudiante que, dadas 5 cláusulas nuevas previamente establecidas, escriba las reglas de Prolog que expresen las siguientes

relaciones:

1. es_madre(X).
2. es_padre(X).
3. es_hijo(X).
4. hermana_de(X, Y).
5. abuelo_de(X, Y) y abuela_de(X, Y).
6. hermanos(X, Y).
7. tia(X, Y). Excluid a los padres.

Cuya interpretación resulta en:



```
1 es_madre(X) :- progenitor(X, _), mujer(X).
2 es_padre(X) :- progenitor(X, _), hombre(X).
3 es_hijo(X) :- progenitor(_, X).
4 hermana_de(X, Y) :- progenitor(Z, X), progenitor(Z, Y), mujer(X), dif(X, Y).
5 abuelo_de(X, Y) :- progenitor(X, Z), progenitor(Z, Y), hombre(X).
6 abuela_de(X, Y) :- progenitor(X, Z), progenitor(Z, Y), mujer(X).
7 hermanos(X, Y) :- progenitor(Z, X), progenitor(Z, Y), dif(X, Y).
8 tia(X, Y) :- progenitor(Z, Y), hermana_de(X, Z), dif(X, Z).
```

Figure 4: Interpretaciones del tercer ejercicio.

Y cuyas pruebas son, por ejemplo:

- es_madre(clara): false.
- es_madre(jose): false.
- es_madre(patricia): true.
- es_padre(tomas): true.
- es_padre(isabel): false.
- es_hijo(jose): true.
- es_hijo(clara): false.
- hermana_de(isabel, patricia): true.
- hermana_de(patricia, ana): false.

- `abuelo_de(tomas, ana): true.`
- `abuela_de(clara, jaime): true.`
- `hermanos(jose, isabel): true.`
- `hermanos(patricia, jaime): false.`
- `tia(isabel, jaime): true.`
- `tia(tomas, ana): false.`

Conclusión

Al realizar esta actividad, he podido comprender, aunque en poca profundidad, la naturaleza de este lenguaje de programación y, más importante aún, de este paradigma. He utilizado conectores lógicos como el operador Y (`,`) y la implicación (`:-`). Entre otros. Estos me sirvieron para profundizar un poco más en la naturaleza de la lógica y de lo que debo esperar de esta primera parte del curso de programación 3.