

INSTITUTO POLITÉCNICO ESCOM

NACIONAL

ESCUELA SUPERIOR DE COMPUTO

PARADIGMAS DE PROGRAMACIÓN

PRÁCTICA DE LABORATORIO N° 5

PROF: GARCÍA FLORIANO ANDRÉS

INTEGRANTES:

SOLIS LUGO MAYRA

SOLARES VELAZCO ARTURO MISAEL

REYES RUIZ YOSELYN ESTEFANY

ELIZALDE HERNÁNDEZ ALAN

TORAL HERNÁNDEZ LEONARDO JAVIER

3CV1

DESCRIPCIÓN DE NUESTRO CÓDIGO

Este código en Prolog establece un conjunto de hechos y reglas para modelar relaciones familiares entre personas. Está organizado en dos secciones principales: "Facts" (hechos) y "Rules" (reglas).

Hechos

En la sección de hechos, se establecen las relaciones básicas entre personas. Cada línea de código que comienza con un hecho describe una relación específica. Por ejemplo:

`male(jack).`

De manera similar, `female(helen)` establece que "helen" es una mujer. Los hechos también pueden incluir relaciones, como `married(santiago, elena)`, que establece que "santiago" está casado con "elena".

```
/* Facts */
male(jack).
male(oliver).
male(ali).
male(james).
male(simon).
male(harry).
male(santiago).
male(erik).
male(ernesto).
male(raul).
female(helen).
female(sophie).
female(jess).
female(lily).
female(paola).
female(elena).
female(silvia).
female(ana).
married(santiago,elena).
married(raul,paola).
married(raul,silvia).
married(ernesto,silvia).
parent_of(silvia,ana).
parent_of(raul,ana).
parent_of(ernesto,helen).
parent_of(silvia,helen).
parent_of(jack,jess).
parent_of(jack,lily).
parent_of(helen, jess).
parent_of(helen, lily).
parent_of(oliver,james).
parent_of(sophie, james).
parent_of(jess, simon).
parent_of(ali, simon).
```

```
parent_of(lily, harry).  
parent_of(james, harry).
```

Reglas

En la sección de reglas, se definen relaciones más complejas que se derivan de los hechos establecidos. Por ejemplo:

`father_of(X,Y):- male(X), parent_of(X,Y).`

Esta regla establece que "X" es el padre de "Y" si "X" es un hombre (**male(X)**) y "X" es el padre de "Y" (**parent_of(X,Y)**). De manera similar, otras reglas definen relaciones como la maternidad, la abuelidad, hermanas, etc.

- **grandfather_of(X, Y)** y **grandmother_of(X, Y)**: Determinan si "X" es el abuelo o la abuela de "Y".
- **sister_of(X, Y)**: Determina si "X" es la hermana de "Y".
- **aunt_of(X, Y)**: Determina si "X" es la tía de "Y".
- **brother_of(X, Y)**: Determina si "X" es el hermano de "Y".
- **uncle_of(X, Y)**: Determina si "X" es el tío de "Y".
- **ancestor_of(X, Y)**: Determina si "X" es un ancestro de "Y".
- **descendant_of(X, Y)**: Determina si "X" es un descendiente de "Y".
- **wife_of(X, Y)** y **husband_of(X, Y)**: Determinan si "X" es la esposa o el esposo de "Y".
- **step_sister_of(X, Y)**: Determina si "X" es la hermanastra de "Y".

Funcionalidad

El código permite realizar consultas sobre las relaciones familiares establecidas. Por ejemplo, al ejecutar la consulta:

?- father_of(X, simon).

El sistema responderá con todas las personas que son padres de "simon", cumpliendo así con la regla **father_of**.

Ahora con nuestras reglas propuestas

1. ENCONTRAR TODOS LOS DESCENDIENTES DE "JAMES":

?- descendant_of(X, james).

swish.swi-prolog.org

```
21 married(raul,paola).
22 married(raul,silvia).
23 married(ernesto,silvia).
24 parent_of(silvia,ana).
25 parent_of(raul,ana).
26 parent_of(ernesto,helen).
27 parent_of(silvia,helen).
28 parent_of(jack,jess).
29 parent_of(jack,illy).
30 parent_of(helen, jess).
31 parent_of(helen, illy).
32 parent_of(oliver,james).
33 parent_of(sophie, james).
34 parent_of(jess, simon).
35 parent_of(all, slmon).
36 parent_of(illy, harry).
37 parent_of(james, harry).
```

?- descendant_of(X, james).

X = harry

1 % Student exercise profile

Examples History Solutions

ESP 12:18 a. m.
LAA 22/04/2024

2. ENCONTRAR TODAS LAS ESPOSAS

?- wife_of(X, Y).

swish.swi-prolog.org

```
24 parent_of(silvia,ana).
25 parent_of(raul,ana).
26 parent_of(ernesto,helen).
27 parent_of(silvia,helen).
28 parent_of(jack,jess).
29 parent_of(jack,illy).
30 parent_of(helen, jess).
31 parent_of(helen, illy).
32 parent_of(oliver,james).
33 parent_of(sophie, james).
34 parent_of(jess, simon).
35 parent_of(all, slmon).
36 parent_of(illy, harry).
37 parent_of(james, harry).
```

?- wife_of(X, Y).

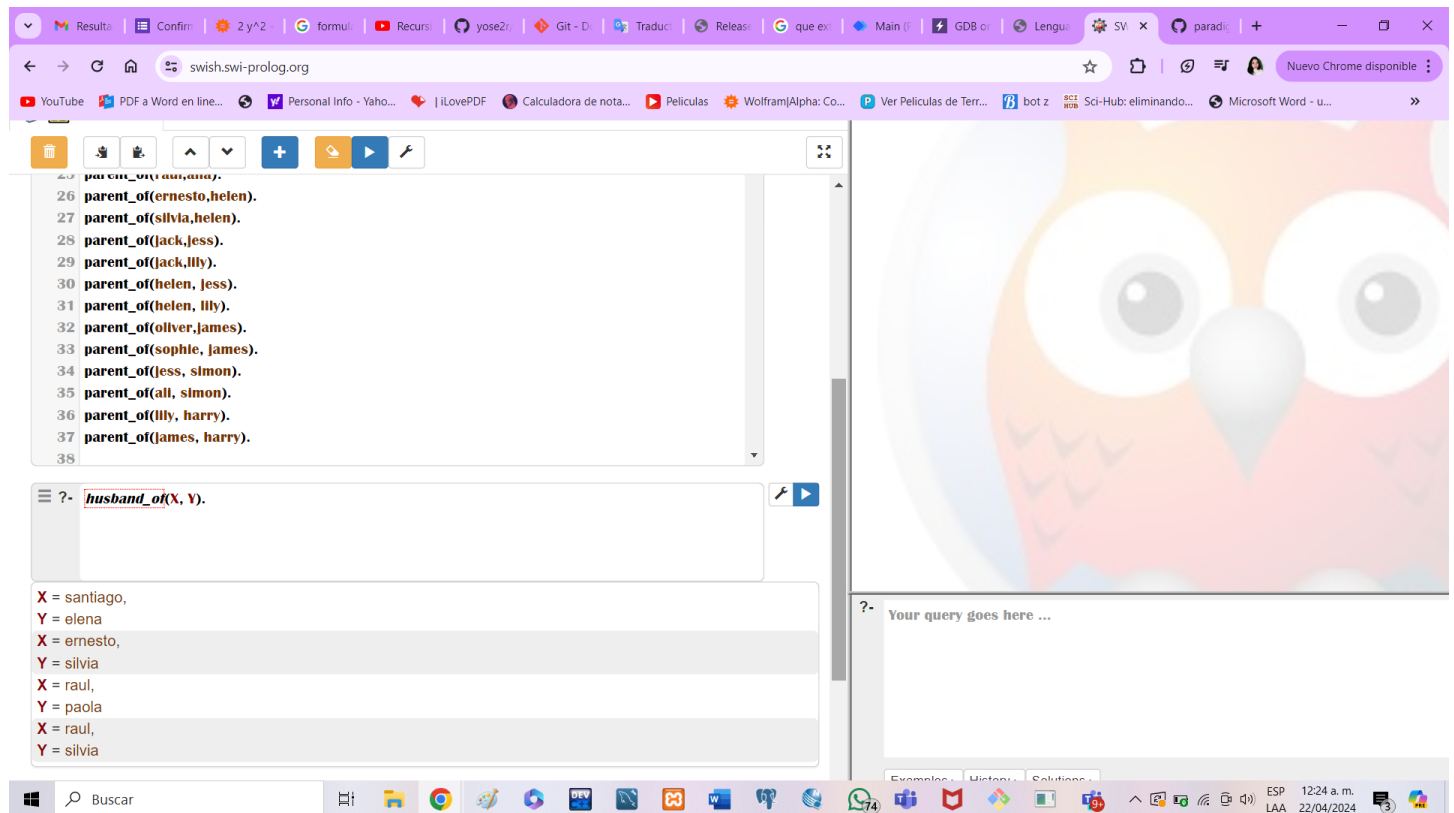
X = raul,
Y = paola
X = santiago,
Y = elena
X = raul,
Y = silvia
X = ernesto,
Y = silvia

Examples History Solutions

ESP 12:22 a. m.
LAA 22/04/2024

3. ENCONTRAR TODOS LOS ESPOSOS

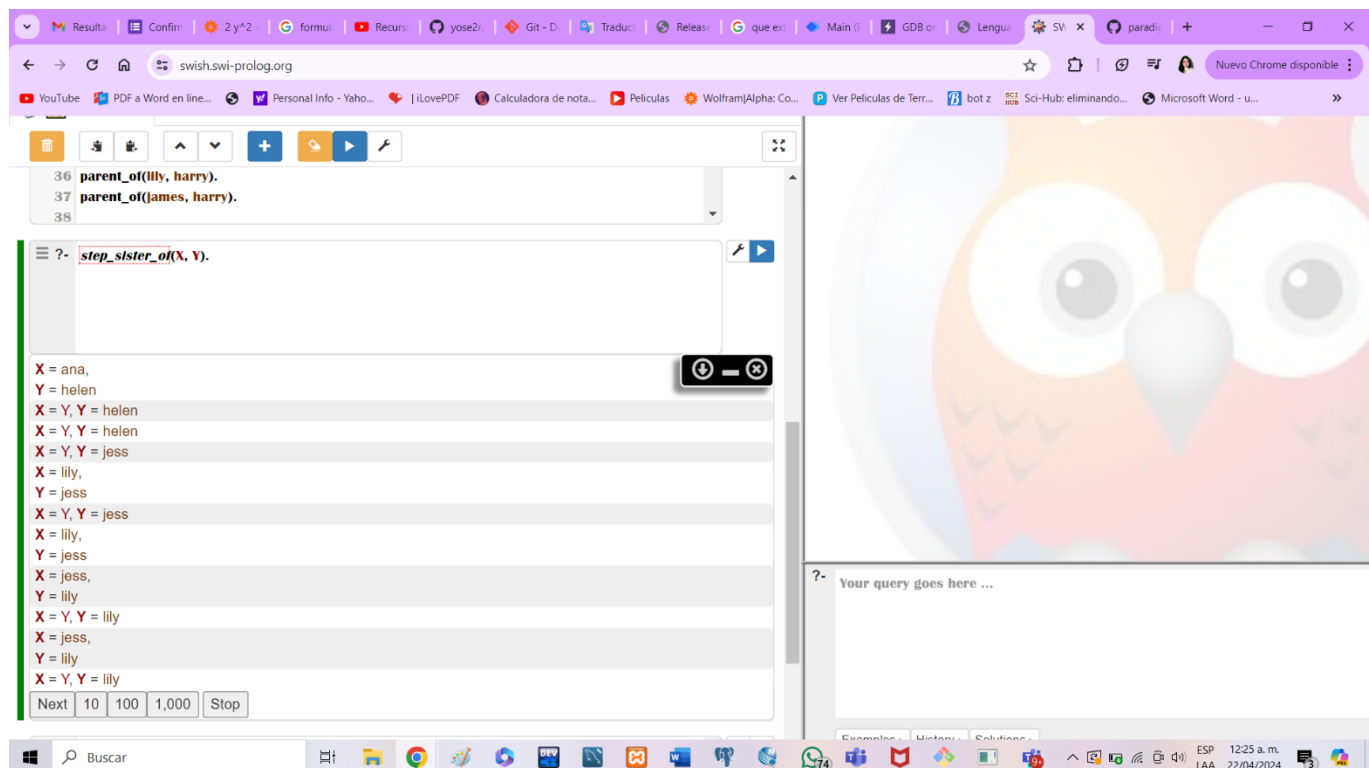
?- husband_of(X, Y).



The screenshot shows a web browser window with the URL `swish.swi-prolog.org`. The left pane displays a list of Prolog facts: `parent_of(ana, ernesto).`, `parent_of(ana, helen).`, `parent_of(silvia, helen).`, `parent_of(jack, jess).`, `parent_of(jack, lily).`, `parent_of(helen, jess).`, `parent_of(helen, lily).`, `parent_of(oliver, james).`, `parent_of(sophie, james).`, `parent_of(jess, simon).`, `parent_of(ali, simon).`, `parent_of(lily, harry).`, and `parent_of(james, harry).`. The right pane shows the query `?- husband_of(X, Y).` and its results: `X = santiago,`, `Y = elena,`, `X = ernesto,`, `Y = silvia,`, `X = raul,`, `Y = paola,`, `X = raul,`, and `Y = silvia`. The bottom status bar shows the system language as ESP and the date as 22/04/2024.

4. ENCONTRAR TODAS LAS HERMANASTRAS

?- step_sister_of(X, Y).



The screenshot shows the same Prolog IDE as before, but with the query `?- step_sister_of(X, Y).` and its results. The results are: `X = ana,`, `Y = helen,`, `X = Y, Y = helen,`, `X = Y, Y = helen,`, `X = Y, Y = jess,`, `X = lily,`, `Y = jess,`, `X = Y, Y = jess,`, `X = lily,`, `Y = jess,`, `X = jess,`, `Y = lily,`, `X = Y, Y = lily,`, `X = jess,`, `Y = lily,`, and `X = Y, Y = lily`. The bottom status bar shows the system language as ESP and the date as 22/04/2024.

Código

```
/* Facts */
```

```
male(jack).
```

```
male(oliver).
```

```
male(ali).
```

```
male(james).
```

```
male(simon).
```

```
male(harry).
```

```
male(santiago).
```

```
male(erik).
```

```
male(ernesto).
```

```
male(raul).
```

```
female(helen).
```

```
female(sophie).
```

```
female(jess).
```

```
female(lily).
```

```
female(paola).
```

```
female(elena).
```

```
female(silvia).
```

```
female(ana).
```

```
married(santiago,elena).
```

```
married(raul,paola).
```

```
married(raul,silvia).
```

```
married(ernesto,silvia).
```

```
parent_of(silvia,ana).
```

```
parent_of(raul,ana).
```

```
parent_of(ernesto,helen).
```

```
parent_of(silvia,helen).
```

```
parent_of(jack,jess).
parent_of(jack,lily).
parent_of(helen, jess).
parent_of(helen, lily).
parent_of(oliver,james).
parent_of(sophie, james).
parent_of(jess, simon).
parent_of(ali, simon).
parent_of(lily, harry).
parent_of(james, harry).
```

```
/* Rules */
```

```
father_of(X,Y):- male(X),
    parent_of(X,Y).
```

```
mother_of(X,Y):- female(X),
    parent_of(X,Y).
```

```
grandfather_of(X,Y):- male(X),
    parent_of(X,Z),
    parent_of(Z,Y).
```

```
grandmother_of(X,Y):- female(X),
    parent_of(X,Z),
    parent_of(Z,Y).
```

```
sister_of(X,Y):- %(X,Y or Y,X)%
    female(X),
    father_of(F, Y), father_of(F,X),X \= Y.
```

```
sister_of(X,Y):- female(X),
```

```

mother_of(M, Y), mother_of(M,X),X \= Y.

aunt_of(X,Y):- female(X),
    parent_of(Z,Y), sister_of(Z,X),!.

brother_of(X,Y):- %(X,Y or Y,X)%
    male(X),
    father_of(F, Y), father_of(F,X),X \= Y.

brother_of(X,Y):- male(X),
    mother_of(M, Y), mother_of(M,X),X \= Y.

uncle_of(X,Y):-
    parent_of(Z,Y), brother_of(Z,X).

ancestor_of(X,Y):- parent_of(X,Y).
ancestor_of(X,Y):- parent_of(X,Z),
    ancestor_of(Z,Y).

descendant_of(X,Y):-parent_of(Y,X).
wife_of(X,Y):-female(Y),married(X,Y).
husband_of(X,Y):-male(X),married(X,Y).
step_sister_of(X,Y):-female(Y),parent_of(Q,Y),parent_of(P,X), parent_of(P,Y),P\=Q.

```

Conclusiones

Este código en Prolog proporciona un marco básico para modelar relaciones familiares y permite realizar consultas para obtener información sobre estas relaciones. Es útil para comprender cómo se pueden utilizar hechos y reglas en Prolog para representar y trabajar con datos relacionales.