

RESUMEN PARADIGMA LÓGICO
FACULTAD REGIONAL BUENOS AIRES
UNIVERSIDAD TECNOLÓGICA NACIONAL

PARADIGMAS DE PROGRAMACIÓN – *K2032*

Trabajos Prácticos

PAZ PORTILLA, José Miguel 2028244
jpazportilla@frba.utn.edu.ar

29 de noviembre de 2023

Índice

1. Introducción a Lógico ->Video 10 Youtube	1
1.1. <i>maestro_alumno.pl</i>	1
1.2. <i>aritmetica.pl</i>	2
1.3. <i>familia.pl</i>	2
2. Orden Superior ->Video 11 Youtube	4
2.1. forall vs not ->caso 1	4
2.2. forall vs not ->caso 2	4
2.3. forall vs not ->caso 3	4
2.4. <i>animales.pl</i>	4

1. Introducción a Lógico -> Video 10 Youtube

En prolog se deben escribir predicados en un archivo con extensión .pl, luego para ejecutarlo desde terminal se debe escribir `swipl nombre_archivo.pl`, para salir `halt.`, para actualizar `make.`, para correr test `run_test`.

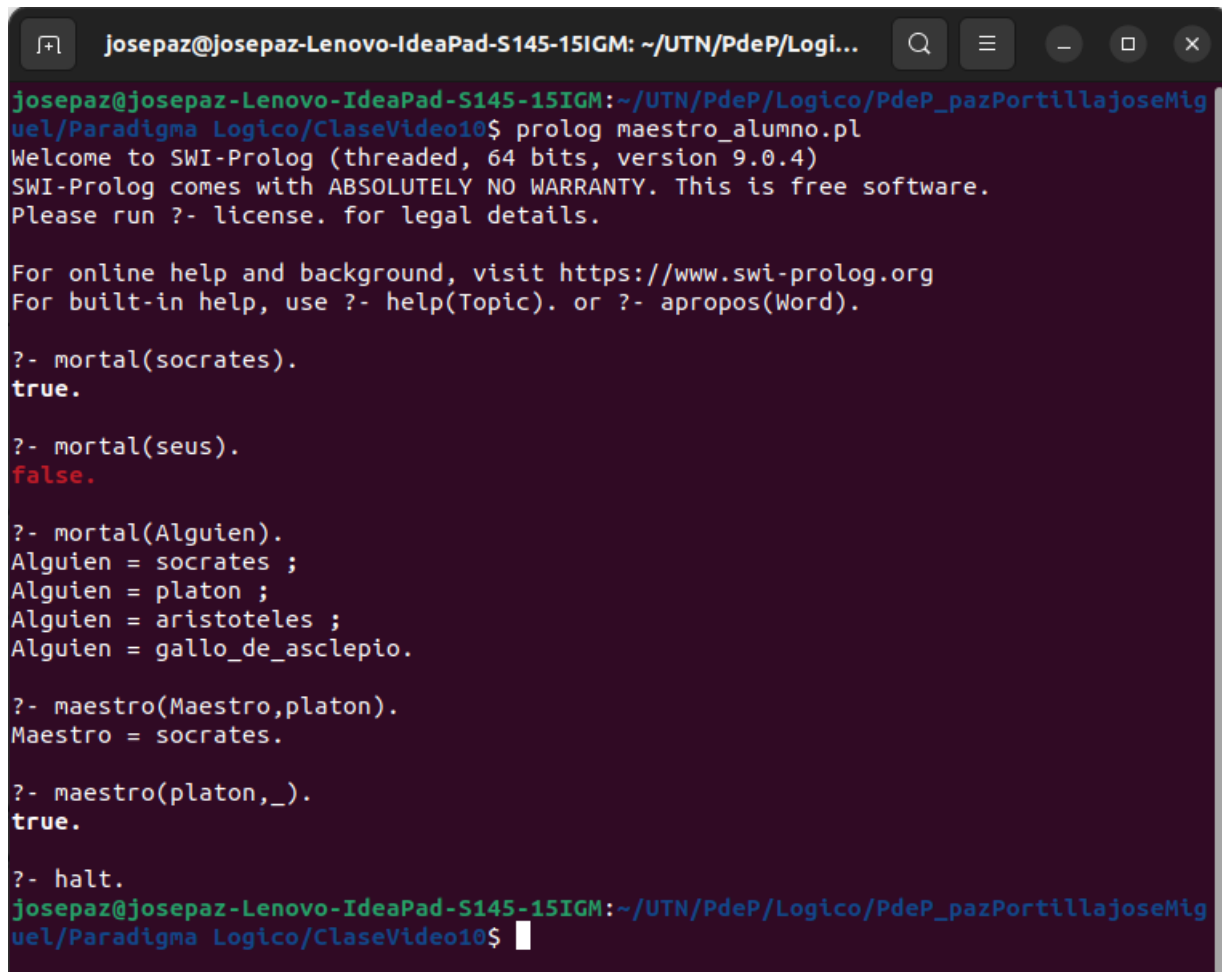
La cantidad de parametros que recibe un predicado se llama aridad. Un predicado puede estar constituido por un hecho o bien por una regla. Donde los hechos no tienen pre requerimiento sino que es algo definido como verdadero, las reglas separan el nombre del predicado con un `:-` y a su derecha los pre requerimientos que debe cumplirse.

Los parametros pueden ser individuos/átomos que empieza con minúscula o bien pueden ser una Variable que empieza con mayúscula.

1.1. *maestro_alumno.pl*

```
1 % humano(Humano)
2 humano(socrates).
3 humano(platon).
4 humano(aristoteles).
5
6 % mortal(Alguien)
7 % Alguien es mortal si es humano o bien si es el gallo de asclepio
8 mortal(Alguien):-
9     humano(Alguien).
10 mortal(gallo_de_asclepio).
11
12 % maestro(Maestro,Alumno)
13 maestro(socrates,platon).
14 maestro(platon,aristoteles).
15 maestro(socrates,jose).
16
17 % groso(Alguien)
18 % Alguien es groso si es maestro de al menos 2 alumnos
19 groso(Alguien):-
20     maestro(Alguien,UnAlumno),
21     maestro(Alguien,OtroAlumno),
22     UnAlumno \= OtroAlumno.
```

Se observa que el predicado humano esta contituido por 3 hechos de aridad 1, el predicado mortal por 1 regla y 1 hecho de aridad 1, el predicado maestro por 2 hechos de aridad 2, el predicado groso por 1 regla de aridad 1.



```

josepaz@josepaz-Lenovo-IdeaPad-S145-15IGM: ~/UTN/PdeP/Logi...
josepaz@josepaz-Lenovo-IdeaPad-S145-15IGM:~/UTN/PdeP/Logico/PdeP_pazPortillajoseMig
uel/Paradigma Logico/ClaseVideo10$ prolog maestro_alumno.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- mortal(socrates).
true.

?- mortal(seus).
false.

?- mortal(Alguien).
Alguien = socrates ;
Alguien = platon ;
Alguien = aristoteles ;
Alguien = gallo_de_asclepio.

?- maestro(Maestro,platon).
Maestro = socrates.

?- maestro(platon,_).
true.

?- halt.
josepaz@josepaz-Lenovo-IdeaPad-S145-15IGM:~/UTN/PdeP/Logico/PdeP_pazPortillajoseMig
uel/Paradigma Logico/ClaseVideo10$

```

Figura 1. Consultas a prolog con base de conocimiento maestro_alumno.pl

1.2. *aritmetica.pl*

```

1 % siguiente(Anterior,Siguiente)
2 % El is es inversible a izquierda, es decir siguiente,
3 % pero is no es inversible a derecha.
4 % El + no es inversible, por ende Anterior debe venir ligado,
5 % El 1 ya es un numero, por ende el + lo reconoce.
6 siguiente(Anterior,Siguiente):-
7     % Como el predicado numero es inversible, liga Anterior para hacer al
8     % predicado siguiente inversible
9     numero(Anterior),
10    Siguiente is Anterior + 1.
11
12 % El predicado numero es inversible ya que between lo es.
13 numero(Numero):-
14     between(0,100,Numero).

```

1.3. *familia.pl*

```
1  % padre(Padre,Hijo)
2  padre(homero,bart).
3  padre(homero,lisa).
4  padre(homero,maggi).
5  padre(abraham,homero).
6
7  % abuelo(Abuelo,Nieto)
8  abuelo(Abuelo,Nieto):-
9      padre(Padre,Nieto),
10     padre(Abuelo,Padre).
11 % Fry es su propio abuelo.
12 abuelo(fry,fry).
13
14 % hermano(UnHermano,OtroHermano)
15 hermano(UnHermano,OtroHermano):-
16     padre(Padre,UnHermano),
17     padre(Padre,OtroHermano),
18     UnHermano \= OtroHermano.
19
20 % Alguien es ancestro de un descendiente si es su padre, abuelo, bisabuelo
21   , ...
22 % ancestro(Ancastro,Descendiente)
23 ancestro(Ancastro,Descendiente):-
24     padre(Ancastro,Descendiente).
25 ancestro(Ancastro,Descendiente):-
26     padre(Ancastro,Alguien),
27     ancestro(Alguien,Descendiente).
```

2. Orden Superior -> Video 11 Youtube

Son predicados que como parametro tienen otro predicado.

2.1. forall vs not -> caso 1

$$\forall x \in X : P(x) \iff \neg \exists x \in X : \neg P(x) \quad (1)$$

Por ende decir todos los animales son friolentos, equivale a decir que no existe un animal que no sea friolento. En prolog serian:

forall(animal(Animal), friolento(Animal)).

y

not((animal(Animal), not(friolento(Animal)))).

2.2. forall vs not -> caso 2

$$\neg \forall x \in X : P(x) \iff \exists x \in X : \neg P(x) \quad (2)$$

Por ende decir no todos los animales son friolentos, equivale a decir que existen animales que no son friolentos. En prolog serian:

not(forall(animal(Animal), friolento(Animal))).

y

animal(Animal), not(friolento(Animal)).

2.3. forall vs not -> caso 3

$$\forall x \in X : \neg P(x) \iff \neg \exists x \in X : P(x) \quad (3)$$

Por ende decir de todos los animales ninguno es friolento, equivale a decir que no existen animales que sean friolentos. En prolog serian:

forall(animal(Animal), not(friolento(Animal))).

y

not((animal(Animal), friolento(Animal))).

2.4. animales.pl

```
1 | % habitat(Animal,Bioma)
2 | % El predicado habitat es inversible porque permite hacer consultas
   |   existenciales.
3 | habitat(jirafa,sabana).
4 | habitat(tigre,sabana).
5 | habitat(tigre,bosque).
```

```

6  habitat(tiburon,mar).
7
8  acuatico(Animal):-
9      habitat(Animal,mar).
10
11 terrestre(Animal):-
12     % Para volverlo inversible
13     habitat(Animal,_),
14     not(habitat(Animal,mar)).
15
16
17 templado(Bioma):-
18     habitat(_,Bioma),
19     Bioma == sabana.
20 templado(Bioma):-
21     habitat(_,Bioma),
22     Bioma == bosque.
23
24 friolento(Animal):-
25     habitat(Animal,_),
26     forall(habitat(Animal,Bioma),templado(Bioma)).
27
28 % come (Depredador,Presa)
29 come(tigre,jirafa).
30 come(tigre,tiburon).
31
32 % hostil/2 relaciona un animal con un bioma, si todos los animales que
33     viven ahi se lo comen.
34 hostile(Animal,Bioma):-
35     habitat(Animal,_),
36     habitat(_,Bioma),
37     forall(habitat(OtroAnimal,Bioma),come(OtroAnimal,Animal)).
38
39 % terrible/2 relaciona un animal con un bioma si todos los animales que se
40     lo comen, habitan en el bioma.
41 terrible(Animal,Bioma):-
42     habitat(Animal,_),
43     habitat(_,Bioma),
44     forall(come(OtroAnimal,Animal),habitat(OtroAnimal,Bioma)).
45
46 % compatibles/2 relaciona dos animales si ninguno de los dos come al otro
47 compatibles(UnAnimal,OtroAnimal):-
48     habitat(UnAnimal,_),
49     habitat(OtroAnimal,_),
50     not(come(UnAnimal,OtroAnimal)),
51     not(come(OtroAnimal,UnAnimal)).
52
53 % adaptable/1 se cumple para los animales que habitan todos los biomas
54 adaptable(Animal):-
55     habitat(Animal,_),
56     forall(habitat(_,Bioma), habitat(Animal,Bioma)).
57
58 % raro/1 se cumple para todos los animales que habitan un unico bioma
59 raro(Animal):-
60     habitat(Animal,Bioma),
61     not((habitat(Animal,OtroBioma), Bioma \= OtroBioma)).

```

```
60 | % dominante/1 se cumple para los animales que se comen a todos los otros
    | animales del bioma en el que viven
61 | dominante(Animal):-
62 |     habitat(Animal,Bioma),
63 |     forall((habitat(OtroAnimal,Bioma), Animal \= OtroAnimal),come(Animal,
    | OtroAnimal)).
```