

LÓGICA

1º AÑO

Clase N.º 8: Lógica de Predicados.

Contenido: Operadores aritméticos. Operadores relacionales con evaluación. Operadores relacionales sin evaluación.

Hola, ¿Cómo están? Bienvenidas y bienvenidos a la clase N°8 de Lógica. La clase anterior trabajamos sobre el concepto de reglas de recursividad, variable anónima y predicado predefinido, específicamente abordamos el “not”. En la clase de hoy seguiremos avanzando por aquellos predicados predefinidos, pero en relación a operaciones. ¡Comencemos!



Prolog, tiene predefinidos aquellos predicados que nos permiten trabajar con operaciones. A su vez, estas operaciones nos permiten trabajar/programar diferentes tipos de objetos:

- **Operadores aritméticos:** Permite llevar a cabo operaciones aritméticas entre números de tipo entero o real. Sin embargo, seleccionamos las operaciones básicas, pero existen para las funciones trigonométricas, valor absoluto, piso, techo, entre otros muchas más.

Expresión	Operación
$X+Y$	suma de X e Y
$X-Y$	X menos Y
$X*Y$	producto de X por Y
X/Y	cociente real de la división de X por Y
$X//Y$	cociente entero de la división de X por Y
X^Y	potencia entera de X a la Y
$X^{**}Y$	potencia real de X a la Y
$X \text{ mod } Y$	resto de la división entera de X por Y

La notación de Positivo y Negativo es prefijo, el resto es infijo.

- Operadores relaciones: Las operaciones relacionales nos permiten establecer relaciones de orden.

- Operadores relacionales con evaluación: Recibe valores numéricos y/o expresiones antes de realizar unificación o comparaciones.

Evalúa el valor de la expresión:

Operador	Significado	Ejemplo
is	Unificación	X is 10 + 2
=:=	Igualdad	10 + 2 =:= 5 + 7
=\=	Desigualdad	10 + 2 =\= 5 + 8
>	Mayor que	11 * 3 > 3 ^ 2
<	Menor que	2 ** 10 < 5 * 2
>=	Mayor o igual que	99.0 >= 0
=<	Igual o menor que	-15 =< 15

Es importante tener presente que, en Prolog los predicados relacionan individuos, esto repercute en la manera que expresamos la resolución de cuentas en el programa. Veamos:

Si queremos preguntarle a prolog cuánto es 13+5, guiándonos por la relación entre predicados e individuos, lo correcto sería escribir en el programa:

?- 13+5

Pero, Prolog nos responde:

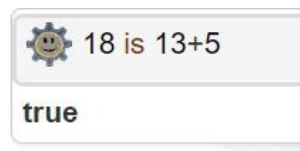
```

≡ ?- 13+5=18.
Singleton variables: [X]
false

```



Eso no es porque no sea posible sumar números, sino que el “+” no es un predicado, pero entonces ¿cómo podemos saber cuál es el resultado de una cuenta? ¿O si una cuenta da un determinado resultado? Bueno, para poder evaluar operadores aritméticos en Prolog hay que utilizar los operadores relacionales con evaluación. Enfocándonos en nuestro ejemplo, utilizaremos el predicado “is”. Dicho predicado realiza la operación de “Si Y es una expresión aritmética, ésta se evalúa y el resultado se intenta unificar con X.” Al ser un predicado, sabemos que podemos hacer consultas individuales como la siguiente, para saber si es cierto que se verifica que una cuenta da un determinado resultado:



```
18 is 13+5
true
```

Pero **ojo**, la cuenta sólo puede ir a la derecha:



```
13+5 is 18
false
```

A la derecha del “is” se escribe una operación aritmética, y a su izquierda se escribe el resultado de esa operación aritmética.

- Operadores relacionales sin evaluación: Recibe cualquier tipo de datos:

Operador	Significado	Ejemplo
=	Unificación	$X = 10 + 2$
==	Igualdad	$10 + 2 == 10 + 2$
\==	Desigualdad	$10 + 2 \backslash == 5 + 7$
@>	Mayor que	banana @> bananin
@<	Menor que	parse @< tree
@>=	Mayor o igual que	ser @>= humano
@<=	Igual o menor que	raton @<= teclado

Veamos algunos ejemplos:

- Sí tenemos dos variables X y Y y las comparamos con el predicado $X == Y$, el resultado será verdadero si X y Y tienen el mismo nombre, y falso, en caso contrario. Sin embargo, si comparamos dos variables diferentes, es decir, $X \backslash == Y$, el resultado será indeterminado, ya que Prolog no puede determinar si dos variables diferentes son iguales o no.
- Sí tenemos los números 5 y 10, podemos compararlos con los predicados $5 < 10$ o $10 >= 5$ para verificar si uno es menor, mayor o igual que el otro.



Copia y pega en Prolog, luego hazle al menos dos preguntas por regla dándole valores numéricos o alfabéticos a las variables.

```
% Predicado para verificar igualdad
igual(X,X).
% Predicado para verificar desigualdad
diferente(X,Y) :- not(igual(X,Y)).
% Predicado para verificar menor que
menor(X,Y) :- X @< Y.
% Predicado para verificar menor o igual que
menor_o_igual(X,Y) :- X @=< Y.
% Predicado para verificar mayor que
mayor(X,Y) :- X @> Y.
% Predicado para verificar mayor o igual que
mayor_o_igual(X,Y) :- X @>= Y.
% Predicado para evaluar expresiones aritméticas
evaluar(X,Y) :- X is Y.
```



Con la siguiente base de conocimiento:

```
% Hechos: edad de las personas
edad(juan,20).
edad(maria,17).
edad(pedro,25).
edad(laura,16).
edad(carlos,19).
% Regla: determinar si una persona es mayor de edad
mayor_de_edad(X) :- edad(X,Edad), Edad @>= 18.
```

2a) Copia la base de conocimiento y pega en prolog.



2b) Escribe en Prolog las siguientes preguntas, con el formato correspondiente al programa, ejecútalas y reporta los resultados:

2.b.1) ¿Es mayor de edad maria?

Consulta en Prolog: _____

Resultados: _____

2.b.2) ¿Es mayor de edad pedro?

Consulta en Prolog: _____

Resultados: _____

2.b.3) ¿Quién es mayor de edad?

Consulta en Prolog: _____

Resultados: _____



Copia y pega esta base de conocimiento en Prolog:

% Definición de hechos

ciudad(paris,2200000,europa).

ciudad(londres,8900000,europa).

ciudad(tokio,13929286,asia).

ciudad(nueva_york,8398748,america).

% Reglas para filtrar ciudades con una población mayor a un valor dado

poblacion_mayor(Ciudad, Valor) :- ciudad(Ciudad,Poblacion,_),
Poblacion @> Valor.

% Reglas para filtrar ciudades por continente

ciudades_en_continente(Ciudad,Continente) :-
ciudad(Ciudad,_,Continente).

% Reglas para ordenar ciudades por población

es_mayor_poblacion(Ciudad1,Ciudad2) :-

ciudad(Ciudad1,Poblacion1,_),ciudad(Ciudad2,Poblacion2,_), Poblacion1
@> Poblacion2.



3b) Escribe en Prolog las siguientes preguntas, con el formato correspondiente al programa, ejecútalas y reporta los resultados:

3.b.1) ¿Ciudades con población mayor a 4000000?

Consulta en Prolog: _____

Resultados: _____

3.b.2) ¿Londres tiene mayor población que Tokio?

Consulta en Prolog: _____

Resultados: _____

3.b.3) ¿Londres es ciudad de que continente?

Consulta en Prolog: _____

Resultados: _____



Llegamos al fin de la clase 8, hemos visto los operadores aritméticos y sus distintos usos en operadores relacionales con o sin evaluación. A partir de esta clase, no habrá contenido teórico, sólo prácticas a fin de aprender a programar diversas bases de conocimientos. ¡Las y los esperamos en la siguiente clase!



