

PARÁCTICA 3. Seminario de Sistemas Basados en conocimiento.

Operaciones aritméticas

Término	Significado	Ejemplo
+/2	Suma	X is A + B.
*/2	Producto	X is 2 * 7.
-/2	Resta	X is 5 - 2.
'/'/2	División	X is 7 / 5.
-/1	Cambio de signo	X is -Z.
'//'/2	División entera	X is 7 // 2.
mod/2	Resto de la división entera	X is 7 mod 2.
'^'/2	Potencia	X is 2^3.
abs/1	Valor absoluto	X is abs(-3).
pi/0	La constante PI	X is 2*pi.
sin/1	seno en radianes	X is sin(0).
cos/1	coseno en radianes	X is cos(pi).
tan/1	tangente en radianes	X is tan(pi/2).
asin/1	arcoseno en radianes	X is asin(7.2).
acos/1	arcocoseno en radianes	X is acos(Z).
atan/1	arcotangente en radianes	X is atan(0).
floor/1	redondeo por defecto	X is floor(3.2).
ceil/1	redondeo por exceso	X is ceil(3.2).

Práctica 3. Realiza las preguntas que se te va pidiendo y muestra los resultados que vas obteniendo.

1.

Prolog, como otros lenguajes de programación, es capaz de realizar operaciones aritméticas. Aquí nos interesan solo las operaciones más básicas que podemos realizar con números enteros. Veamos cómo se expresan con el operador `is` :

- Suma

Ejemplo:

```
?- 8 is 6+2.
```

```
    true.
```

```
?- 5 is 3+3.
```

¿Por qué dieron esos resultados?

- Resta

```
?- 12 is 15-3.
```

```
?- -3 is 5-8.
```

- Multiplicación

```
?- 63 is 9*7.
```

```
?- 736 is 8*96.
```

- División

```
?- 4 is 36/12.
```

```
?- 25 is 125/5.
```

```
?- X is 8*96.
```

¿Por qué unas preguntas tienes como respuesta true/false y otras una solución?

- Resto de la división

?- 1 is mod(7,2) .

?- 0 is mod(1002,2)

?- Resto is mod(83,7) .

% también podemos tener números reales

?- Resultado is ((23*2)+5)/25.

% el tiempo pasa

?- Años=23,Lustros is 23/5,Dias is Años*365,Horas is Dias*24,Minutos is Horas*60,Segundos is Minutos*60.

Como habrás notado, la expresión aritmética que se evalúa aparece siempre a la derecha del operador `is`. Si probamos al revés, no funciona.

?- X is 3*5.

?- 3*5 is X.

Este error es debido al particular modo de funcionar que tienen las operaciones aritméticas en Prolog.

Los operadores tienen asociada una prioridad.

Por ejemplo, la expresión $a+b*c$ es en realidad el término $+(a, *(b, c))$.

Esto es así porque el operador producto (*) tiene más prioridad que el operador suma (+).

Si no fuese así, se trataría del término $*(+(a, b), c)$.

Recuerda que, para Prolog, una expresión como $3 + 2$ es un tipo de término, un edulcorante sintáctico del término complejo $+(3, 2)$. Es más, una consulta como:

```
?- X is 3*5.
```

En realidad es equivalente a:

```
?- is(X, *(3,5)).
```

¿Cuál es la aridad de las relaciones `is` y `*`?

1. Comparando números

Los operadores que podemos utilizar en Prolog para comparar números enteros son los siguientes:

- Mayor estricto y menor estricto que:

```
?- 2 < 4.
```

```
?- 2 > 4.
```

- Mayor o igual y menor o igual que:

```
?- 4 =< 4.
```

```
?- 2 >= 4.
```

% ojo, ese operador no existe

```
?- 2 => 4.
```

- Igual o distinto que:

?- 4 == 4.

?- 4 == 6.

?- 3 \= 4.

?- 4 \= 4.

Revisando los 4 resultados obtenidos anteriormente, ¿Qué operaciones son == y \=?

Fíjate en que estos operadores fuerzan el cálculo de cualquier operación aritmética que necesiten evaluar.

?- 3 =< 2+1.

?- 5+5 == 9+1.

?- 5+5=9+1.

?- 5+5 == 10.

Ojo, el operador = no es lo mismo que == . El primero unifica argumentos, el segundo fuerza la evaluación de la operación aritmética.

?- 5+5 == 9+1.

?- 5+5=9+1.

?- 5+5 =:= 10.

?- 5+5=10.

Como pasaba antes con las operaciones aritméticas, si usamos estos operadores con variables, tenemos que asegurarnos antes de que estas variables están previamente instanciadas. De lo contrario, darán error:

?- X < 2.

?- 0 > Negativo.

?- Negativo = -2, 0 > Negativo.

?- Negativo = -2, 0 < Negativo.