

Taller Prolog 5

Aritmética, I/O, Recursividad

Operaciones Aritméticas

El operador '=' se utiliza en Prolog para unificar expresiones, pero no evalúa expresiones aritméticas.

Para *evaluar* el valor de una expresión aritmética se utiliza el operador 'is'. Así por ejemplo:



```
jmlon@MS-7850: ~  
jmlon@MS-7850:~$ swipl  
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 6.6.6)  
Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam  
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,  
and you are welcome to redistribute it under certain conditions.  
Please visit http://www.swi-prolog.org for details.  
  
For help, use ?- help(Topic). or ?- apropos(Word).  
  
?- X is 5+2^3, Y is 7-4/2, Z is 3*5.  
X = 13,  
Y = 5,  
Z = 15.  
  
?- X is abs(-1), Y is pi/2, Z is sin(Y).  
X = 1,  
Y = 1.5707963267948966,  
Z = 1.0.  
  
?- 
```

Se pueden incorporar expresiones dentro de reglas, por ejemplo para calcular la distancia X de un móvil que se desplaza a velocidad constante V por un tiempo T , se utiliza la expresión

$$X = V * T$$

En Prolog la podemos definir como la regla:

```
% distancia/2 : Calcula la distancia recorrida por un móvil a
velocidad V durante un tiempo T
distancia(V,T,X) :- X is V*T.
```

Entradas y salidas (Input/Output)

Prolog define los predicados `write/1` y `read/1` que leen y escriben respectivamente un átomo desde la consola de consultas. Adicionalmente, existen algunos predicados que facilitan dar formato a la salida, específicamente:

`tab/1` : Hace una tabulación del número de espacios indicados.

`nl/0` : Hace que el cursor pase a la siguiente línea (new line).

Por ejemplo el siguiente procedimiento permite calcular de forma interactiva la distancia recorrida por un móvil:

```
% distancia/0: Calcular la distancia recorrida por un móvil de forma
interactiva
distancia :- write('Ingrese velocidad '), read(V), nl,
              write('Ingrese tiempo '), read(T), nl,
              distancia(V,T,X),
              write('La distancia recorrida es '), write(X), nl.
```

Ejercicio 1

1. En trigonometría la [ley de cosenos](#) relaciona un lado de un triángulo con los lados adyacentes y el ángulo opuesto.

$$c^2 = a^2 + b^2 - 2 \cdot a \cdot b \cdot \cos(\gamma)$$

Definir un procedimiento `ladoOpuesto/4` que tome como entradas las longitudes de los lados `a`, `b`, el ángulo opuesto `γ`, y que devuelva la longitud del lado `c`.

Hacer un par de consultas ilustrando el uso del procedimiento `ladoOpuesto`.

2. Hacer una versión interactiva de `ladoOpuesto/0` que pregunte al usuario los valores `a`, `b`, `γ` e imprima en pantalla el lado `c`.

Ilustrar la correcta operación del procedimiento con los mismos casos considerados en el punto 3.

Procedimientos recursivos

Se pueden definir procedimientos recursivos por medio de reglas que definan uno o varios casos base (e.g. para $n=0$) y reglas recursivas que manejen los demás casos.

Por ejemplo, la función factorial se define recursivamente así:

$$factorial(n) = \begin{cases} 1 & n=0 \\ n * factorial(n-1) & n>0 \end{cases}$$

Se puede implementar mediante el siguiente procedimiento recursivo:

```
% factorial/2 : Calcula factorial de n recursivamente
% arg1 = valor n
% arg2 = Resultado n!
factorial(0,1) :- !.
factorial(N,X) :- M is N-1, factorial(M,Y), X is N*Y.
```

El símbolo ! se utiliza en Prolog para indicar un corte. Cuando la evaluación de una regla llega a un corte, el Prolog no evaluará otras reglas o hará retrocesos a metas anteriores al corte.

Ejercicio 2

Leonardo de Pisa (más conocido como Fibonacci) descubrió [la secuencia de Fibonacci](#):

0, 1, 1, 2, 3, 5, 8, 13, ...

la cual tiene innumerables aplicaciones para describir todo tipo de fenómenos naturales.

La secuencia se puede definir recursivamente observando que siempre empieza en 0,1 y que de ahí en adelante todo término es la suma de los dos que le preceden. En forma de recurrencia, la sucesión se define así:

$$fib(n) = \begin{cases} n & n=0,1 \\ fib(n-1) + fib(n-2) & n \geq 2 \end{cases}$$

- a) Definir un procedimiento recursivo fib/2 que dado un valor n calcule el correspondiente número de fibonacci.
- b) Ilustrar el correcto funcionamiento calculando fib(N,F) para n=1,3,8,15,50. Comprobar que la respuesta obtenida sea correcta.
- c) Hacer un procedimiento interactivo fib/0 que le pregunte al usuario el término N que desea obtener y le muestre en pantalla el número de fibonacci correspondiente.
- d) Ilustrar el correcto funcionamiento de fib/0 con N=15, 0, -1, 3.5, hola.
- e) Si se presentan errores en (d), proponer una solución e implementarla en el procedimiento. Repetir las consultas y validar que el resultado es correcto.

Informe:

Enviar la base de datos (<NombreApellido>-<ID>-bd5.pl) y el informe incluyendo todas las consultas solicitadas (<NombreApellido>-<ID>-consultas5.txt).

No olvidar documentar apropiadamente los hechos y reglas en la base de datos.