

Taller Prolog 7

Recursividad en programación

Introducción

De forma análoga a como en la inducción matemática se define un caso base y se realiza el paso inductivo (para $n+1$) partiendo de un caso previamente conocido (la hipótesis para n), en programación se pueden definir procedimientos recursivos: Se definen reglas para establecer uno o varios casos base (e.g. para $n=0$) y reglas recursivas que calculen los demás casos haciendo uso de casos anteriores.

Ejemplo 1: Función factorial

La función factorial se define recursivamente así:

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

Esta función se implementa 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 y no hará retrocesos a metas anteriores al corte.

Ejemplo 2: Conteo descendente

Para realizar un conteo descendente partiendo de un valor inicial M, se define la regla recursiva que decrementa M en una unidad y se repite recursivamente,

excepto cuando llega a cero, en cuyo caso termina el conteo.

El procedimiento en Prolog queda así:

```
% conteoDescendente/1 : Realiza un conteo descendente desde un valor
% inicial M
% arg1 : Valor inicial
conteoDescendente(0) :- !.
conteoDescendente(M) :- write(M), nl, MM is M-1,
                        conteoDescendente(MM) .
```

Ejemplo 3: Llamando otros procedimientos dentro de un ciclo

Digamos que queremos obtener todos los factoriales desde M. Entonces podemos modificar el `conteoDescendente` para hacer un ciclo que calcule el factorial de M para cada uno de los valores de M.

```
% cicloFactoriales/1 : Calcula los factoriales para los valores
% desde M hasta 0.
% arg1 : Valor inicial
cicloFactoriales(0) :- factorial(0,X), write(X), nl, !.
cicloFactoriales(M) :- factorial(M,X), write(X), nl,
                        MM is M-1, cicloFactoriales(MM) .
```

Ejercicio 1

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. El primer argumento es `N` y el segundo es el resultado.

- b) Ilustrar el correcto funcionamiento calculando $\text{fib}(n)$ para $n=1,3,8,15,30$. Comprobar que la respuesta obtenida sea correcta.
- c) Se puede calcular $\text{fib}(40)$? $\text{fib}(50)$? $\text{fib}(100)$? Qué ocurre en estos casos?
- d) La secuencia no está definida para $n < 0$. Qué hace el programa si se le pide calcular $\text{fib}(-1)$?

Ejercicio 2

Hacer un procedimiento para obtener todos los números de Fibonacci desde un valor inicial M .

Ilustrar el correcto funcionamiento para algunos valores positivos de M .

Ilustrar el correcto funcionamiento para cuando $M=0, -1$.

Informe:

Enviar la base de datos (`<NombreApellido>-<ID>-bd7.pl`) y el informe incluyendo todas las consultas solicitadas (`<NombreApellido>-<ID>-consultas7.txt`).

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