Ing. de Sistemas e Informática
Matemáticas Discretas
Jorge M. Londoño P.

Taller de Prolog 6

Recursividad y ciclos

En Prolog un procedimiento recursivo permite evaluar un predicado sobre elementos de un dominio definido recursivamente. El procedimiento recursivo se define de forma análoga a la definición recursiva del dominio: Se crea una regla normal (no recursiva) para los casos base y una o más reglas recursivas para los casos recursivos.

Ejemplos

Ciclos `for`

1. Iterando sobre un rango de valores

Una construcción muy común en programación son los ciclos `for`, donde lo que se busca es iterar sobre todos los valores en un rango determinado. El siguiente procedimiento ilustra la construcción recursiva equivalente a un ciclo que imprime todos los valores desde M hasta 1 en orden descendiente:

```
% listaRango/1 : Dado un valor inicial arg1, itera sobre todos los
% enteros hasta 1.
listaRango(M) :- M<1, !.
listaRango(M) :- write(M), nl, MM is M-1, listaRango(MM).
```

(Qué papel cumple el caso base?)

2. Factorial de un rango de enteros

Usualmente se utilizan ciclos para realizar ciertas operaciones para un rango de valores. Por ejemplo, el siguiente programa extiende el caso anterior para imprimir los factoriales de los valores desde M hasta 0.

```
% factorialDesde/1 : Escribe los factoriales desde un entero inicial M
% arg1: M - valor inicial desde el cual se calculan los factoriales
factorialDesde(M) :- M<0, !.
factorialDesde(M) :- factorial(M,F), write(F), nl, MM is M-1, factorialDesde(MM).
```

3. Ciclos ascendientes

El orden en que se itera sobre el contador del ciclo puede ser importante en muchas aplicaciones. Simplemente incrementando la variable contadora y ajustando la condición de terminación, es posible hacer un ciclo ascendente. Por ejemplo, el siguiente programa calcula los factoriales desde 0 hasta un valor M.

```
% factorialHasta/1 : Escribe los factoriales desde 0 hasta un entero final M
% arg1: M – valor final hasta el cual se calculan los factoriales
factorialHasta(M) :- factorialDesdeHasta(0,M).

% factorialDesdeHasta/2 : Escribe los factoriales desde I hasta un entero final M
% arg1 : Valor inicial
% arg2 : Valor final
factorialDesdeHasta(I,M) :- I>M, !.
factorialDesdeHasta(I,M) :- factorial(I,F), write(F), nl,
                            J is I+1, factorialDesdeHasta(J,M).
```

Actividades a realizar:

Desarrollar los siguientes ejercicios en una base de datos llamada bd7-<Nombre>-<id>.pl y anexar las consultas ejemplos solicitadas en consultas7-<Nombre>-<id>.txt.

Ejercicios:

1. Para obtener los divisores de un número N se prueba divisibilidad por todos los naturales entre 1 y N (inclusive). Hacer un procedimiento divisores/2 que imprima en pantalla todos los divisores de N. Nota: Para el chequeo de divisibilidad se comprueba si el residuo de la división es cero. El residuo se obtiene con el operador mod/2.

- Probar el procedimiento con un compuesto
- Probar el procedimiento con un primo
- Probar el procedimiento con un número factorial
- Probar el procedimiento con un negativo

2. Implementar un procedimiento sucesiónAritmética/4 que acepte como entradas las constantes A, R, y el número de términos N. El procedimiento debe imprimir los primeros N términos de la sucesión.

- Ilustrar el correcto funcionamiento del procedimiento para dos combinaciones de valores A, R, N.

3. Implementar el procedimiento sucesiónGeométrica/4 que acepte como entradas los parámetros A,R de la sucesión y el número de términos N. El procedimiento debe imprimir en pantalla los N primeros términos de la sucesión.

- Ilustrar el correcto funcionamiento del procedimiento para dos combinaciones de valores A, R, N.

Nota: Documentar apropiadamente sus procedimientos.