

PROGRAMACIÓN DECLARATIVA

Área personal / Cursos / (23666) PROGRAMACIÓN DECLARATIVA / Bloque 3
/ Enunciado Laboratorio 7

LABORATORIO 7 (Manipulación de términos y orden superior)

Para medir lo aprendido durante la Práctica 7, los alumnos deberán enviar los programas y soluciones a las preguntas que a continuación se indican. El envío se realizará en un archivo ZIP de nombre:

gN_Apellidos_Lab7.ZIP

donde N es el número del equipo de trabajo. El archivo ZIP contendrá los ficheros:

Apellidos_Lab7.pl con el código Prolog de todos los programas solicitados.

Apellidos_Lab7.txt con los datos personales de la persona que hace el envío y el enunciado y solución de aquellas preguntas que (no pudiéndose responder vía programa) se formulen en cada uno de los ejercicios del laboratorio.

En todos los casos, “**Apellidos**” son los apellidos del alumno que hace el envío.

Ejercicio 27.

Definir un predicado, **ficha(H)**, que actuando sobre un hecho almacenado en una base de datos lo visualice en forma de ficha. Por ejemplo, si se lanzase el objetivo

```
?- ficha(libro(tol185,  
autor('Tolkien', 'J.R.R.' ),  
titulo('El Señor de los Anillos'),  
editorial(minotauro),  
prestado(jul101, fecha(28, noviembre, 2004)))).
```

se mostraría:

```

libro:

    tol85,

    * autor:

        Tolkien,

        J.R.R.,

    * titulo:

        El Señor de los Anillos,

    * editorial:

        minotauro,

    * prestado:

        jul101,

        ** fecha:

            28,

            noviembre,

            2004

```

[Observación: Escriba 8 caracteres para el sangrado de cada elemento.]

Ejercicio 28.

a) Defina un predicado **unific_a(X,Y)** que tenga éxito cuando las expresiones

X e Y puedan ser unificadas. Esto es, queremos que se comporte como el predicado predefinido en Prolog para la unificación de expresiones. b) Amplíe el programa anterior para que el predicado **unific_a/2** compruebe la ocurrencia de variables. Denomine al nuevo predicado **unific_b(X,Y)**.

Ejercicio 29.

a) Defina un predicado **reduce(List,Func,Base,Result)** que toma como argumentos una lista, **List**, el nombre de una función de aridad 2, **Func**, (representada por un predicado de aridad 3) y un valor base, **Base**, para dar un resultado **Result**, que reduce la lista a un valor. Supongamos que el predicado **reduce** toma como argumentos de entrada, por ejemplo, una lista [e1, e2, e3], una función f (de aridad 2) y un valor inicial (base) b, el objeto es producir como

resultado el valor $f(e1, f(e2, f(e3, b)))$.

(**AYUDA:** utilice el predicado predefinido **apply/2** que aplica “funciones” sobre sus argumentos de forma eficiente. Por ejemplo, si hemos definido la función **sum(X,Y,S)**, el objetivo **apply(sum, [2,3,S])** construye el átomo **sum(2,3,S)** y lo lanza evaluándolo, con lo que se obtiene **S=5**.)

b) Haciendo uso del predicado **reduce/4** y de la “función” **sum(X,Y,S)**, que toma los números **X** e **Y**, dando su suma **S=X+Y** como resultado, definir el predicado **sumList(List, Suma)** que suma los elementos de una lista de números.

c) Haciendo uso del predicado **reduce/4** y de la “función” **mult(X,Y,P)**, que multiplica los números **X** e **Y** dando su producto **P**, definir el predicado **multList(List, Prod)** que multiplica los elementos de una lista de números.

Última modificación: jueves, 24 de abril de 2014, 14:57:57